

การพัฒนาซอฟต์แวร์เชิงคอมโพเนนท์  
Component Software Development



นายสุรเชษฐ นิธิเสถียร 40010904

นายสุรินทร์ อัสวชัยสิทธิกุล 40010916



2/11  
5848/1  
2543

เลขหมู่.....  
เลขทะเบียน..... 42763  
วัน, เดือน, ปี 10 ส.ย. 2545

b.....  
f.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DM 2 2547

การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์  
Component Software Development



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาระบบงานเชิงวัตถุโดยใช้คอมโพเนนต์สำหรับสถาปัตยกรรมเอ็นทีเยร์ส  
(N-Tiers System Development Using Component)

ผู้จัดทำ

1. นายร่มธรรม สินธุประสิทธิ์ รหัสประจำตัว 40010644
2. นายวัลลภ วรกิจวิบูลย์ รหัสประจำตัว 40010718



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์

สุรเชษฐ นิธิเสถียร	40010904
สุรินทร์ อิศวชัยสิทธิกุล	40010916
อาจารย์รวรรณ ลิม โทคา	อาจารย์ที่ปรึกษา
ปีการศึกษา 2543	

### บทคัดย่อ

เทคโนโลยี COM (Component Object Model) เป็นหนึ่งในพื้นฐานของการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ มีการขยายออกไปใช้มากมาย เป็นประโยชน์ในการพัฒนา โปรแกรมเชิงวัตถุเป็นอย่างดี ด้วยลักษณะที่เป็นออบเจกต์ นอกจากนั้นการติดต่อกับแต่ละคอมโพเนนต์ก็ใช้ลักษณะที่เป็นมาตรฐานช่วยให้การสร้างโปรแกรมขึ้นมาน่ากลับมาใช้ใหม่โดยไม่ขึ้นกับภาษาที่ใช้พัฒนาโปรแกรม

ในวิทยานิพนธ์ฉบับนี้ศึกษาถึงการพัฒนาระบบงานขนาดใหญ่ โดยใช้การพัฒนาแบบคอมโพเนนต์ซึ่งเป็นเทคโนโลยีที่สนับสนุนการพัฒนาระบบเชิงออบเจกต์อเรียนเตดที่มีขนาดใหญ่ รองรับการทำงานที่มีจำนวนมากได้อย่างมีประสิทธิภาพ ซึ่งเทคโนโลยีที่ศึกษาเป็นกรอบการทำงานที่ทางไมโครซอฟต์เสนอออกมาคือ DNA (Distribute interNet Architecture) โดยมีเทคโนโลยีที่สนับสนุนการทำงาน คือ COM+, Microsoft SQL Sever 2000, IIS (Internet Information Service Server), Internet Explorer, ASP (Active Serve Page)

ตัวอย่างการพัฒนาระบบขนาดใหญ่ที่วิทยานิพนธ์ฉบับนี้เสนอเป็นระบบการท่องเที่ยวที่รวมการจองเครื่องบิน, รถบัส, โรงแรม, ร้านอาหาร, การจองแพ็คเกจการท่องเที่ยว ได้แบบครบวงจร ซึ่งการพัฒนาจะเป็นการพัฒนาเชิงคอมโพเนนต์และใช้เทคโนโลยีตามกรอบของ DNA โดยภาษาที่ใช้พัฒนาจะใช้ Borland Delphi 5 และ Visual Basic 6 ร่วมกันในการพัฒนา

## Component software development

Surachad Nithisathind

Surin Asawachaisittigul

Dr. Worawat limpoka Advisor

### Abstract

COM (Component Object Model) Technology is one of base technology used to develop software component. They have a lot implementation and have advantage to object oriented programming because of Object characteristic. Additionally, Every connection between any component are standard form that make it easy to develop COM object by any tools.

This thesis is study in developing large scale system by Component Base Development (CBD), which is technology that support development of large scale object oriented system, with Microsoft's Architecture call DNA (Distributed interNet Architecture), that contain of COM+, Microsoft SQL Server 2000, Internet Information Service (IIS), Internet Explorer and Active Server Page (ASP)

Example of large scale system development in this thesis is travelling agency with reserve aircraft seat, bus, bus seat, hotel room, restaurant and package tour feature using DNA by Borland Delphi 5.0 and Microsoft Visual Basic 6.0

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ ท่านอาจารย์ ดร. วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอ นอกจากนั้นยังได้ให้ความรู้และแนวคิดในการทำงานรวมถึงการใช้ชีวิต อันมีค่าอย่างยิ่งแก่ผู้เขียน ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณที่ ๆ เพื่อนๆ ห้อง ปฏิบัติการฮาร์ดแวร์ ห้อง Olala ห้อง ICL และเพื่อนๆ ห้องเน็ตเวิร์ก ที่ช่วยเป็นกำลังใจและให้ประสบการณ์ในการใช้ชีวิตในมหาวิทยาลัยที่มีค่าอย่างยิ่งให้กับผู้เขียน

สุดท้ายนี้ ขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

สุรเชษฐ นิธิเสถียร  
สุรินทร์ อัสวชัยสิทธิกุล

## สารบัญ

หน้าที่

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	X
สารบัญภาพ	XI
บทที่ 1 บทนำ	1
วัตถุประสงค์	
บทที่ 2 การวิเคราะห์ ออกแบบและการพัฒนาระบบเชิงวัตถุ	3
2.1 ความหมายของการ โปรแกรมเชิงออบเจกต์ (Object-Oriented Programming)	3
2.2 คุณสมบัติของการ โปรแกรมเชิงออบเจกต์	3
2.2.1 เอกลักษณ์ (Abstraction)	3
2.2.2 การซ่อนรายละเอียด (Encapsulation)	3
2.2.3 การสืบทอด (Inheritance)	3
2.2.4 โพลิมอร์ฟิซึม (Polymorphism)	4
2.3 ออบเจกต์ (Object)	4
2.4 ข่าวสาร (Message)	5
2.5 คลาสและอินสแตนซ์ (Class and Instance)	5
2.6 ตัวอย่างการออกแบบออบเจกต์	5
2.7 การสืบทอด	7
บทที่ 3 Component Object Model (COM)	8
3.1 ซอฟต์แวร์คอมโพเนนท์	8
3.2 COM คอมโพเนนท์	9
3.2.1 COM เซิร์ฟเวอร์	9
3.2.2 COM อินเตอร์เฟส	11
3.2.1.1 ลักษณะของ IUnknown	12
3.2.3 ดูอัลอินเตอร์เฟส (Dual Interface)	13
3.2.4 โคคลาส (Coclasses) และ คลาสแฟกทอรี (Class factory)	14
3.2.5 มาร์แชลลิ่งแมคานิซึม ( Marshaling mechanism )	14
3.3 COM ไคลเอนต์	15
3.4 ส่วยขยาย COM	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 Automation sever and controller	16
3.4.2 ActiveX controls	16
3.4.3 Type libraries	16
3.4.4 Active Server Page (ASP)	17
3.4.5 Active Document	17
3.4.6 Visual cross-process objects	17
<b>บทที่ 4 การพัฒนาคอมโพเนนต์โดยใช้วิซวลเบสิก</b>	<b>18</b>
4.1 ชนิดของคอมโพเนนต์ในวิซวลเบสิก	18
4.2 อินเตอร์เฟสกับวิซวลเบสิก	18
4.3 การสร้าง COM ออบเจกต์ในวิซวลเบสิก	19
4.4 ตัวอย่างการพัฒนาคอมโพเนนต์ซอร์ฟแวร์	19
ตัวอย่างที่ 4-1 พัฒนาคอมโพเนนต์ซอฟต์แวร์ประเภทอินโพรเซส (In-process)	19
ตัวอย่างที่ 4-2 การพัฒนาคอมโพเนนต์ซอร์ฟแวร์ประเภทเอาท์ออฟโพรเซส (Out-of-process)	25
ตัวอย่างที่ 4-3 การพัฒนาคอมโพเนนต์ประเภทรีโมทเซิร์ฟเวอร์ (Remote Server)	31
<b>บทที่ 5 การพัฒนา COM โดยใช้ Borland Delphi 5.0</b>	<b>36</b>
5.1 กรณีศึกษาการพัฒนา คอมโพเนนต์ ออบเจกต์ ประเภท อิน – โพรเซส (In-Process COM Object)	36
5.1.1 ขั้นตอนการพัฒนา	36
5.1.2 ขั้นตอนการเรียกใช้งาน ด้วย Borland Delphi 5.0	37
5.1.3 ขั้นตอนการเรียกใช้งาน ด้วย Microsoft Visual Basic 6.0	38
5.2 กรณีศึกษาการพัฒนา คอมโพเนนต์ ออบเจกต์ ประเภท เอาท์ – โพรเซส (Out-Process COM Object)	39
5.2.1 ขั้นตอนการพัฒนา	39
5.2.2 ขั้นตอนการเรียกใช้งาน ด้วย Borland Delphi 5.0	41
5.2.3 ขั้นตอนการเรียกใช้งาน ด้วย Microsoft Visual Basic 6.0	41
5.3 กรณีศึกษาการพัฒนา คอมโพเนนต์ ออบเจกต์ ประเภท รีโมท เซิร์ฟเวอร์ (Remote Server)	42
5.3.1 ขั้นตอนการพัฒนา	43
5.3.2 ขั้นตอนการเรียกใช้งาน ด้วย Borland Delphi 5.0	44
<b>บทที่ 6 ActiveX Data Object (ADO)</b>	<b>47</b>
6.1 ADO และ OLE DB	47
6.2 ออบเจกต์ ADO	48
6.3 คอมโพเนนต์ของ ADO ใน Borland Delphi 5.0	48

6.4 การใช้คอมโพเนนต์ ADO ใน Borland Delphi 5.0	49
6.5 โครงสร้างแบบ Master/Detail	51
6.6 การใช้โครงสร้างแบบ Master/Detail ใน Borland Delphi 5.0	51
6.7 การจัดการทรานแซกชันด้วย ADO	52
บทที่ 7 อีเวนต์ และ คอลแบ็กของ COM	53
7.1 การสร้างอโตเมชันเซิร์ฟเวอร์ในการใช้งานอีเวนต์	53
7.2 การสร้างไคลเอนต์ในการใช้งานอีเวนต์	54
7.3 การสร้างเซิร์ฟเวอร์แบบหลายไคลเอนต์	56
บทที่ 8 ไมโครซอฟต์แวร์ทรานแซกชันเซิร์ฟเวอร์	58
8.1 การพัฒนาระบบงานเอ็นทีเอชโดยไมโครซอฟต์แวร์ทรานแซกชันเซิร์ฟเวอร์	58
8.2 ข้อดีในการใช้เอ็มทีเอส	58
8.3 เอ็มทีเอสคอมโพเนนต์	59
8.4 บริการต่าง ๆ ของ MTS	60
8.5 ตัวอย่างของเอ็มทีเอสแอปพลิเคชัน	61
บทที่ 9 วินโดวส์ดีเอ็นเอ	63
9.1 ความหมาย	63
9.2 องค์ประกอบของวินโดวส์ดีเอ็นเอ	63
บทที่ 10 สภาพแวดล้อม COM+ รันไทม์และบริการของ COM+	65
10.1 สถาปัตยกรรมของ COM+	65
10.1.1 Attributed-Based Programming	65
10.1.2 คอนเทก (Context)	65
10.1.3 COM+ ไคลเอนต์ กับ Context	66
10.1.4 คอนเทก และ อพาร์ทเมนต์	67
10.1.5 Interception	68
10.2 การปรับแต่ง ออบเจกต์เพื่อใช้กับ สภาพแวดล้อม COM+ ไรม์	69
10.3 COM+ แอปพลิเคชัน	70
10.3.1 เซิร์ฟเวอร์แอปพลิเคชัน ( Server Application )	70
10.3.2 ไลบรารีแอปพลิเคชัน ( Library Application )	70
10.4 การสร้าง COM+ แอปพลิเคชัน	72
บทที่ 11 เทรดโมเดลของ COM+	79
11.1 โพรเซส ( Process )	79
11.2 เทรด ( Thread )	79
11.3 มัลติเทรด ( Multi-Thread )	79
11.4 Thread-Safe	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.5 Multi-Threading และ COM	79
11.6 อพาร์ทเมนต์	80
11.7 วิธีที่ COM+ runtime จัดการกับ User Apartment	82
11.8 การเรียกใช้เมธอดแบบ Inter-Apartment	83
11.9 Marshalling Interface Pointer ระหว่าง อพาร์ทเมนต์	83
11.10 Threading Model แบบ Both และ Neutral	83
11.11 วิธีพิจารณาว่าแอปพลิเคชันเป็นมัลติเทรด	86
11.12 วิธีการที่ออบเจกต์จะบอกว่าเป็น Thread-Safe	86
บทที่ 12 ทรานแซคชัน	87
12.1 คุณสมบัติของทรานแซคชัน	87
12.2 ทรานแซคชันแบบกระจาย (Distribute Transaction)	88
12.3 โพรโตคอลแบบทู-เฟสคอมมิต (2-Phase Commit Protocol)	88
12.4 ทรานแซคชันกับ COM+	90
12.5 ทรานแซคชันที่เกี่ยวข้องกับออบเจกต์มากกว่า 1 ตัว	92
บทที่ 13 Scalability	94
13.1 JIT Activation	94
13.2 Object Pooling	96
13.3 Synchronization	98
13.4 Activity	98
บทที่ 14 ความปลอดภัยบนอินเทอร์เน็ต (Internet security)	102
14.1 การแก้ปัญหาการลอบฟัง	102
14.1.1 การเข้ารหัสข้อมูลและถอดรหัสข้อมูล (Encryption and Decryption)	102
14.1.2 การเข้ารหัสแบบสมมาตร (Symmetric-Key Encryption)	102
14.1.3 การเข้ารหัสโดยพับลิคคีย์ (Public-Key Encryption)	103
14.2 การแก้ปัญหาการลักลอบเปลี่ยนข้อมูล	103
14.2.1 ลายเซ็นดิจิทัล (Digital signature)	103
14.3 การแก้ปัญหาการปลอมตัวเข้ามา	104
14.3.1 Certificate	104
14.3.2 Authentication	105
14.4 SSL Protocol	105
14.5 Certificate hierarchy	106
14.6 Certificate chains	106
14.7 การตรวจสอบ Certificate chains	108
14.8 SSL Server authentication	109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14.9 Client Authentication	110
14.10 SSL Handshake	111
14.11 โมเดลความปลอดภัยของ COM+	112
14.11.1 Declarative Security	112
14.11.2 Role	113
14.11.3 การสร้าง Role	113
14.11.4 การเพิ่มผู้ใช้เข้าไปในโรล	113
14.11.5 การกำหนดความปลอดภัยให้แต่ละส่วนของแอปพลิเคชัน	114
บทที่ 15 แอพลิเคชันเซิร์ฟเวอร์เพจและอินเทอร์เน็ตอินฟอร์เมชันเซอร์วิส	115
15.1 แนวคิดการทำงานของแอพลิเคชันเซิร์ฟเวอร์เพจ	115
15.2 ออบเจกต์พื้นฐานของเอเอสพี	115
15.2.1 ออบเจกต์รีเควส	116
15.2.2 ออบเจกต์เรสพอนซ์	117
15.2.3 ออบเจกต์เสตชัน	119
15.2.4 ออบเจกต์แอปพลิเคชัน	119
15.2.5 ออบเจกต์เซิร์ฟเวอร์	120
15.2.6 ออบเจกต์เอเอสพีเอเรอร์	121
15.3 อินเทอร์เน็ตอินฟอร์เมชันเซอร์วิส (ไอไอเอส)	121
บทที่ 16 การออกแบบระบบ	123
16.1 ความต้องการของระบบ	123
16.1.1 ส่วนกลาง	123
16.1.2 ธนาคาร	127
16.2 การออกแบบระบบงาน OlalaTour	128
16.3 การสร้าง Use Case Diagram และ แผนภาพคลาส	132
16.4 การสร้างแผนภาพคอมโพเนนต์ (Component Diagram)	134
บทที่ 17 ลักษณะแอปพลิเคชัน	137
17.1 ระบบงาน Olala Tour บนอินเทอร์เน็ต	137
17.1.1 หน้าจอหลัก	137
17.1.2 การสมัครเป็นสมาชิก	138
17.1.3 การค้นหาสินค้าและการเลือกสินค้าใ้รถเข็น	139
17.1.4 การดูสินค้าในรถเข็น	148
17.1.5 การซื้อหรือจองสินค้าจากรถเข็น	149
17.1.6 การชำระเงิน	152
17.2 Admin application	154

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17.2.1	เข้าสู่การใช้งาน Admin application	154
17.2.2	การจัดการเกี่ยวกับสถานที่ท่องเที่ยว	154
17.2.3	การจัดการแพคเกจการท่องเที่ยว	157
17.2.4	การจัดเกี่ยวกับสมาชิกของระบบ	162
17.2.5	การดูแลประวัติการทำงานจากระบบ	163
บทที่ 18	เปรียบเทียบประสิทธิภาพระหว่าง COM+ และ EJB	164
บทที่ 19	บทวิจารณ์ และสรุป	166
ภาคผนวก ก	ยูนิฟิเคชัน โมเดลลิ่งแลงแกจ (Unification Modeling Language :UML )	168

### บรรณานุกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้าที่
4-1 รายละเอียดของคอน โทล	21
4-2 รายละเอียดของคอน โทล	28
4-3 รายละเอียดของคอน โทล	34
12-1 คำทรานแซคชันแอตทริบิวต์	91
13-1 ค่าที่เป็นไปได้ทั้งหมดของการซิงโครไนซ์	100
13-2 สถานะของ Activity ของออบเจกต์ซึ่งผู้สร้างอยู่ใน Activity	100
15-1 ออบเจกต์พื้นฐานของเอเอสพี	115
15-2 ตัวอย่างของ ServerVariables	116
18-1 เปรียบเทียบคุณสมบัติระหว่าง COM+ และ EJB	164
ก-1 ความหมายของพารามิเตอร์ต่างๆ ของทรานซิชัน	177



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

รูปที่	หน้าที่
2-1 การจำลองส่วนประกอบของออบเจกต์	4
2-2 การส่งข่าวสารระหว่างออบเจกต์	5
2-3 รายละเอียดของคลาส	5
2-4 ไคอะแกรมของตัวอย่างการออกแบบ	6
2-5 การส่งเมสเสจที่มีพารามิเตอร์	6
2-6 คลาสไคอะแกรมของรถยนต์	7
2-7 การสืบทอดคลาสรถยนต์	7
3-1 แสดงลักษณะของซอฟต์แวร์คอมพิวเตอร์	9
3-2 แสดง In-process server	10
3-3 แสดง Out-of-process server และ Remote server	11
3-4 แสดง ลักษณะของ COM	11
3-5 แสดง ลักษณะของ vtable	13
3-6 แสดงลักษณะของอินเตอร์เฟสที่เป็น คูอัลอินเตอร์เฟส	13
3-7 แสดง แสดงส่วนขยาย COM	15
4-1 การใช้โปรแกรม OLE View แสดง ไทป์ไลบรารีของไฟล์ InprocServer.dll	18
4-2 หน้าจอเริ่มต้นในการสร้างคอมพิวเตอร์ประเภทอินโพรเซสโดยใช้วิซวลเบสิก	20
4-3 โค้ดของคลาส โมดูล InprocServer	21
4-4 หน้าจอติดต่อกับผู้ใช้	22
4-5 การเรียกใช้คอมพิวเตอร์ InprocServer.dll	22
4-6 โค้ดของโปรแกรม	23
4-7 ผลการทำงานของโปรแกรม	23
4-8 หน้าต่าง Import Type Library	24
4-9 การเรียกใช้คอมพิวเตอร์ InprocServer.dll	24
4-10 หน้าจอส่วนติดต่อกับผู้ใช้และโค้ด	25
4-11 ผลการทำงานของไคลเอนต์แอปพลิเคชันที่สร้างจากเดลไฟ	25
4-12 โค้ดในโมดูล mdlOutProcServer	26
4-13 โค้ดในคลาสโมดูล clsOutProcServer	27
4-14 โค้ดของฟอร์ม frmOutProcServer	27
4-15 ผลการทดสอบแอปพลิเคชัน	27
4-16 หน้าจอส่วนติดต่อกับผู้ใช้	28
4-17 การเรียกใช้คอมพิวเตอร์ prjOutProcServer.exe	29
4-18 โค้ดภายในฟอร์มที่ใช้ติดต่อกับผู้ใช้	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4-19 ผลการทดลองเมื่อพิมพ์ข้อความในแท็บ็อกซ์ของไคลเอนท์แล้วคลิก Send Text ข้อความจะไปปรากฏที่แท็บ็อกซ์ของเซิร์ฟเวอร์	30
4-20 ผลการทดลองเมื่อพิมพ์ข้อความลงในแท็บ็อกซ์ของเซิร์ฟเวอร์เมื่อไคลเอนท์คลิก Show Text ข้อความจะไปปรากฏที่แท็บ็อกซ์ในเฟรม Result	30
รูปที่ 4-21 ผลการทดลองเมื่อพิมพ์ข้อความในแท็บ็อกซ์ของเคลไฟไคลเอนท์แล้วคลิก Send Text ข้อความจะไปปรากฏที่แท็บ็อกซ์ของเซิร์ฟเวอร์	31
รูปที่ 4-22 ผลการทดลองเมื่อพิมพ์ข้อความลงในแท็บ็อกซ์ของเซิร์ฟเวอร์เมื่อเคลไฟไคลเอนท์คลิก Show Text ข้อความจะไปปรากฏที่แท็บ็อกซ์ในเฟรม Result	31
4-23 โค้ดภายในคลาสโมดูล clsRemoteServerWC	32
4-24 เซ็ทบ็อกซ์ Remote Server Files	33
4-25 การรีจิสเตอร์รีโมทคอมโพเนนต์บนเครื่องไคลเอนท์โดยใช้โปรแกรม Clireg32.exe	33
4-26 โปรแกรม Clireg32.exe	33
4-27 หน้าจอส่วนติดต่อกับผู้ใช้	34
4-28 โค้ดภายในฟอร์มที่ใช้ติดต่อกับผู้ใช้	35
4-29 ผลการทำงานของไคลเอนท์แอปพลิเคชัน	35
5-1 แสดงหน้าจอของ COM Object Wizard ในการสร้างอิน-โพรเซส คอมโพเนนต์	36
5-2 แสดงซอร์สโค้ดของตัวโปรแกรมของ อิน-โพรเซส คอมโพเนนต์	37
5-3 แสดงการเรียกใช้ อิน-โพรเซส คอมโพเนนต์โดยใช้ Microsoft Visual Basic 6.0	39
5-4 แสดงรายละเอียดของตัวเอาท์-โพรเซส เซิร์ฟเวอร์	40
5-5 แสดงหน้าจอของ Automation Object Wizard ในการสร้าง เอาท์-โพรเซส คอมโพเนนต์	40
5-6 แสดงรายละเอียดของเมธอดในตัวเอาท์-โพรเซสคอมโพเนนต์	40
5-7 แสดงตัวอย่างการเรียกใช้งานเอาท์-โพรเซสคอมโพเนนต์	42
5-8 แสดงลักษณะของรีโมท เซิร์ฟเวอร์	43
5-9 แสดงหน้าจอของ Remote Data Module ในการสร้างรีโมท เซิร์ฟเวอร์	43
5-10 แสดงหน้าจอของ type library editor ในการเพิ่ม เมธอดและคุณสมบัติต่างๆ ให้กับอินเตอร์เฟส	44
5-11 แสดงการเพิ่มเติมรายละเอียดในส่วน Remote Data Module	44
5-12 แสดงการเลือก DCOMConnect คอมโพเนนต์	45
5-13 แสดงการเลือกตัวรีโมท เซิร์ฟเวอร์ในคุณสมบัติ ServerName	45
5-14 แสดงการรันไคลเอนต์จะให้ตัวเซิร์ฟเวอร์รันด้วย	46
6-1 แสดงการทำงานของ ADO	48
6-2 แสดงหน้าจอในการเลือกสร้าง Connection String	49
6-3 แสดงหน้าจอในการสร้าง Connection String	49
6-4 แสดงหน้าจอในการสร้างคำสั่งภาษา SQL	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6-5 แสดงผลลัพธ์ จาก โปรแกรมทดสอบที่ติดต่อกับฐานข้อมูลสำเร็จ	50
6-6 แสดง หน้าจอของ โปรแกรมตัวอย่าง	51
6-7 แสดงผลลัพธ์ของการทดสอบลักษณะ Master/Detail	52
7-1 แสดงหน้าจอในเพิ่มคุณสมบัติอีเวนต์เข้ากับคอมโพเนนต์ใหม่ใน Borland Delphi 5.0	53
7-2 แสดงการกำหนดเมธอดของ อินเตอร์เฟส IEventIntf2 และ IEventIntf2Events	54
7-3 แสดงโปรแกรมทดสอบการเป็นไคลเอนต์ ของ คอมโพเนนต์TEventIntf2	55
8-1 ความสัมพันธ์ระหว่างไคลเอนต์แอปพลิเคชันและออบเจกต์ที่รันอยู่ในรันไทม์เอนไวรอนเมนต์	59
8-2 เอ็มทีเอสเอ็กซ์พลอเรอร์ซึ่งเป็นส่วนหนึ่งของไมโครซอฟท์เมเนจเมนต์คอนโซล	60
8-3 กระบวนการโอนเงินโดยใช้เอ็มทีเอสและไม่ใช้เอ็มทีเอส	61
9-1 สถาปัตยกรรมของวินโดวส์ดีเอ็นเอ	64
10-1 แสดงโพรเซส อพาร์ทเมนต์ และ คอนเทค	66
10-2 แสดงCOM+ ไคลเอนต์ทั้งสองประเภท	66
10-3 แสดงอพาร์ทเมนต์ และ คอนเทค	67
10-4 แสดงInterception ที่ใช้กับ COM+ ออบเจกต์ชนิด Out-of-Process	68
10-5 แสดงInterception ที่ใช้กับ COM+ ออบเจกต์ชนิด In-Process	69
10-6 แสดง Surrogate Process	70
10-7 แสดงCOM+ Server Application 2 ตัว ใช้ Server Application ตัวที่สามร่วมกัน	71
10-8 แสดงCOM+ Server Application 2 ตัวใช้ Library Application ร่วมกัน	72
10-9 แสดงComponent Service Explorer	72
10-10 แสดงขั้นตอนแรกในการสร้าง COM+ Application	73
10-11 แสดงหน้าจอแรกของ COM+ Application Install Wizard	74
10-12 แสดงหน้า Install Pre-Built COM+ Application หรือ สร้างใหม่	74
10-13 แสดงการกำหนดชื่อของ Application และ ชนิดการ Activate	74
10-14 แสดงการเลือก Identity ของ COM+ Application	75
10-15 แสดงComponent Service Explorer ที่แสดง Application ใหม่	75
10-16 แสดงGeneral Tab ของ COM+ Application Properties Window	76
10-17 แสดงคอมโพเนนต์โฟลเดอร์สำหรับ COM+ Application	76
10-18 แสดงหน้าจอแรกของ Component Install Wizard	77
10-19 แสดงการเลือก Install ระหว่างคอมโพเนนต์ใหม่ กับคอมโพเนนต์ที่ลงทะเบียนเรียบร้อยแล้ว	77
10-20 แสดงการเลือก คอมโพเนนต์ที่ต้องการจะติดตั้ง	78
10-21 แสดงไดอะล็อก COM+ Component Properties	78
11-1 แสดงProcess, Apartment, Thread และ COM ออบเจกต์	82
11-2 แสดงIntra-Process Marshaling	83

11-3 แสดงเรขาคณิตที่เป็น MTA และ STA และออบเจกต์ที่มี เรขาคณิตโมเดลชนิด Both	84
11-4 แสดงอพาร์ทเมนต์ชนิด STA และ MTA เรียกใช้ออบเจกต์ที่รันอยู่ใน TNA	85
12-1 การคอมมิททรานแซกชัน โดยใช้โพรโตคอลทวู-เฟสคอมมิท	89
12-2 การโรลแบ็กทรานแซกชันโดยใช้โพรโตคอลทวู-เฟสคอมมิท	90
12-3 ออบเจกต์หลาย ๆ ตัวทำงานอยู่ในทรานแซกชันเดียวกัน	92
13-1 แสดงออบเจกต์ที่สนับสนุน JIT Activation ในสถานะ Activated	95
13-2 แสดงออบเจกต์ที่สนับสนุน JIT Activation ในสถานะ Deactivated	95
13-3 ไลอะลอก Component Properties	96
13-4 แสดง Object Pooling	97
14-1 แสดงการเข้ารหัสแบบสมมาตร	102
14-2 การเข้ารหัสแบบอสมมาตร	103
14-3 แสดงการลงลายเซ็นดิจิทัล	104
14-4 แสดงลำดับขั้นของ CA	106
14-5 แสดง Certificate chains	107
14-6 แสดงการ Trust ของ Certificate chains	108
14-7 แสดงการไม่ Trust ของ Certificate chains	108
14-8 แสดงการ Authentication ของเซิร์ฟเวอร์	109
14-9 แสดงการ Authentication ของไคลเอนต์	110
16-1 แสดงตารางฐานข้อมูล และความสัมพันธ์ระหว่างตารางต่าง ๆ ในระบบ OlalaTour	128
16-2 แสดงความสัมพันธ์ และแอททริบิวต์ ของตารางฐานข้อมูล ส่วน สถานที่ ของระบบ OlalaTour	129
16-3 แสดงความสัมพันธ์ และ แอททริบิวต์ของระบบ การจองและลูกค้า ของระบบ OlalaTour	130
16-4 แสดงความสัมพันธ์ และแอททริบิวต์ ของส่วน ชุดการเดินทางของระบบ OlalaTour	131
16-5 แสดง Use Case Diagram ของระบบ OlalaTour	132
16-6 แสดงแผนภาพคลาส ของระบบ OlalaTour ในส่วน ของสถานที่, ชุดการท่องเที่ยว และ Log ของระบบ	132
16-7 แสดงเมธอด และแอททริบิวต์ของคลาสในส่วนที่เกี่ยวข้องกับสถานที่โดยละเอียด	133
16-8 แสดงรายละเอียดของเมธอด และแอททริบิวต์ของคลาสในส่วนที่เกี่ยวข้องกับชุดการท่องเที่ยว และ Log	134
16-9 แสดงรายละเอียดคอมโพเนนต์ SystemLog	135
16-10 แสดงรายละเอียดคอมโพเนนต์ PackageSystem	135
16-11 แสดงรายละเอียดคอมโพเนนต์ PlaceSystem	136
17-1 หน้าจอหลัก	137
17-2 การสมัครสมาชิก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17-3 การค้นหาสายการบิน	138
17-4 การค้นหาสนามบิน	139
17-5 การแสดงผลการค้นหาสายการบิน	140
17-6 การดูรายละเอียดของตั๋วการเดินทาง	141
17-7 การเลือกสายการบินขาไปของตั๋วการเดินทางแบบไป-กลับ	141
17-8 การเลือกสายการบินขากลับของตั๋วการเดินทางแบบไป-กลับ	142
17-9 การค้นหาเที่ยวการเดินทาง	143
17-10 การแสดงผลการค้นหาเที่ยวการเดินทาง	144
17-11 การค้นหาโรงแรม	145
17-12 การแสดงผลการค้นหาโรงแรมพร้อมใส่จำนวนห้อง	145
17-13 การค้นหาสถานที่จากแผนที่ 1	146
17-14 การค้นหาสถานที่จากแผนที่ 2	146
17-15 การค้นหาสถานที่จากตัวเลือก	147
17-16 การแสดงข้อมูลในรถเข็น	148
17-17 การคำนวณราคา	149
17-18 การใส่ข้อมูลผู้เดินทาง 1	149
17-19 การใส่ข้อมูลผู้เดินทาง 2	150
17-20 การเลือกผู้เดินทางจากข้อมูลของสมาชิก	150
17-21 การแก้ไขข้อมูลของผู้เดินทาง	151
17-22 แสดงผลการจอง	151
17-23 การดูข้อมูลที่ได้จองไว้	152
17-24 การชำระเงิน	152
17-25 การโอนเงินผ่านทางธนาคาร	153
17-26 การยืนยันการชำระเงิน	153
17-27 แสดงการเริ่มทำงาน	154
17-28 แสดงการ Login	154
17-29 แสดงหน้าจอการจัดการสถานที่	155
17-30 แสดงการสร้างสถานที่	155
17-31 แสดงการแก้ไขสถานที่	156
17-32 แสดงการดูข้อมูลสถานที่	156
17-33 แสดงการจัดการกับคุณสมบัติของสถานที่	157
17-34 แสดงการจัดการแพคเกจ	157
17-35 แสดงการดูข้อมูลการจองแพคเกจ	158
17-36 แสดงก่อนการสร้างแพคเกจ	158

17-37 แสดงการใส่สถานที่เข้าสู่แพคเกจ	159
17-38 แสดงการจองโรงแรมของแพคเกจ	159
17-39 แสดงการจองเครื่องบินของแพคเกจ	160
17-40 แสดงการจองร้านอาหารของแพคเกจ	160
17-41 แสดงการจองรถบัสของแพคเกจ	161
17-42 แสดงการจองแพคเกจเสร็จ	161
17-43 แสดงการตกลงหรือปฏิเสธการจอง	162
17-44 แสดงการจัดการสมาชิกของระบบ	162
17-45 แสดงการดูประวัติการทำงานจากระบบ	163



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

การพัฒนาโปรแกรมคอมพิวเตอร์นั้นได้มีการพัฒนาจาก การใช้การพัฒนาโดยใช้รูปแบบโครงสร้าง (Structural Development) ไปเป็นการพัฒนาเชิงวัตถุ (Object Oriented Development) ซึ่งก็มีข้อดีที่สามารถจัดการกับระบบที่มีความซับซ้อนสูงได้อย่างมีประสิทธิภาพมากขึ้น แต่ในขณะเดียวกันตัวการพัฒนาแบบวัตถุ ก็มีจุดบกพร่องอยู่เป็นจำนวนมากเช่น จุดบกพร่องของหลักการสืบทอดคุณสมบัติ (Inheritance) การนำกลับมาใช้ใหม่ซึ่งยังไม่มีประสิทธิภาพเพียงพอเป็นต้น ซึ่งทำให้เกิดแนวความคิดใหม่ของการ นำกลับมาใช้ใหม่ซึ่งมีประสิทธิภาพสูงกว่า และเชื่อถือได้มากกว่ารูปแบบเดิม

การพัฒนาโดยสถาปัตยกรรม COM เป็นพื้นฐาน (COM base Architecture) เป็นอีกแนวทางหนึ่งซึ่งได้ถูกนำมาใช้ โดยมีจุดเด่นดังนี้

- มีความเป็นอิสระในการพัฒนาไม่ขึ้นกับเครื่องพัฒนาใด ๆ เช่น COM ออบเจ็กต์ที่ถูกสร้างจาก Borland Delphi 5.0 ก็สามารถเรียกใช้โดย โปรแกรมที่พัฒนาโดย Microsoft Visual Basic ได้
- มีความสมบูรณ์ภายในตัวเอง ไม่ต้องคอมไพล์ใหม่เมื่อมีการเรียกใช้จากไคลเอนต์
- มีความสะดวกในการเรียกใช้ การเรียกใช้ไม่ว่าตัว COM ออบเจ็กต์นั้นจะอยู่ที่ไหนก็ไม่มี ความแตกต่างมากนัก
- เป็นหลักการพื้นฐานของ OLE และ ActiveX
- เนื่องจากมีการใช้อินเทอร์เน็ตเป็นตัวติดต่อกับ ไคลเอนต์ซึ่งอินเทอร์เน็ตนั้นจะไม่มีการเปลี่ยนแปลง การ
- เปลี่ยนแปลงภายในตัวเซิร์ฟเวอร์จะไม่ผลต่อตัวไคลเอนต์โดยตรง

เมื่อระบบงานนั้นต้องการจะมีเพิ่มความสามารถในการเชื่อมต่อกับ อินเทอร์เน็ต ทางบริษัท ไมโครซอฟท์ก็ได้เสนอสถาปัตยกรรมที่เป็นสภาพแวดล้อมที่เหมาะสม กับ แพลตฟอร์มของไมโครซอฟท์ ซึ่งเรียกว่า DNA องค์ประกอบหนึ่งที่สำคัญของ DNA คือ สภาพแวดล้อม COM+ รันไทม์ หรือ MTS เวอร์ชัน 3.0 นั้นเอง ซึ่งมีบริการใหม่ ๆ เพิ่มเติมจาก MTS เวอร์ชันก่อนหน้ารวมทั้งยังใช้งานง่ายยิ่งกว่า

ปรัชญาพันธบัตรฉบับนี้ได้นำเสนอทั้งทฤษฎี และตัวอย่างการสร้าง ทั้ง COM คอมโพเนนท์ และ การออกแบบระบบงานขนาดใหญ่ โดยใช้หลักการคอมโพเนนท์ ซึ่งมีความละเอียดพอสมควร

### วัตถุประสงค์

ภายในโครงการนี้ได้ทำการศึกษาถึงทฤษฎี ,การสร้าง และการนำ COM ออบเจ็กต์ไปใช้งานโดยมีจุดประสงค์ดังนี้

1. เข้าใจถึงทฤษฎี พื้นฐานของ COM ลักษณะภายในตัว COM ออบเจ็กต์ และองค์ประกอบที่เกี่ยวข้อง
2. รู้วิธีการสร้าง COM ออบเจ็กต์ ทั้งแบบ in-process, out-of-process และแบบ remote server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เข้าใจถึงทฤษฎีพื้นฐานของสภาพแวดล้อม COM+ รันไทม์และบริการต่าง ๆ ที่มาพร้อมกันกับ DNA
4. สามารถออกแบบระบบงานเชิงคอมโพเนนต์ และสร้างระบบงานที่ออกแบบในลักษณะคอมโพเนนต์ โดยใช้เทคโนโลยี COM + ร่วมกับ DNA



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### การวิเคราะห์ ออกแบบและการพัฒนาระบบเชิงวัตถุ

#### 2.1 ความหมายของการโปรแกรมเชิงออบเจกต์ (Object-Oriented Programming)

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

การเขียนโปรแกรมแบบออบเจกต์ คือ วิธีการเขียนโปรแกรมคอมพิวเตอร์ที่ใช้แนวความคิดในการจำลองวัตถุในโลกแห่งความเป็นจริงด้วยออบเจกต์ซึ่งทำให้ได้โครงสร้างของโปรแกรมเป็นระเบียบมากขึ้นและเอื้ออำนวยต่อการพัฒนาโปรแกรมในรุ่นต่อไป

#### 2.2 คุณสมบัติของการโปรแกรมเชิงออบเจกต์

การโปรแกรมเชิงออบเจกต์มีแนวคิดพื้นฐานที่ควรทำความเข้าใจดังต่อไปนี้

##### 2.2.1 เอกลักษณ์ (Abstraction)

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

##### 2.2.2 การซ่อนรายละเอียด (Encapsulation)

เอ็นแคปซูลชันเป็นการรวบรวมข้อมูลและโปรซีเจอร์หรือฟังก์ชันเข้าด้วยกันเป็นออบเจกต์ โดยรายละเอียดเกี่ยวกับข้อมูล ส่วนประกอบ โครงสร้างและการทำงานภายในของออบเจกต์จะถูกซ่อนอยู่จากผู้ใช้งาน (Client) โดยผู้ใช้งานจะเห็นเพียงอินเตอร์เฟซของออบเจกต์ซึ่งการทำงานภายในของออบเจกต์จะซ่อนอยู่ภายใต้อินเตอร์เฟซ ตัวอย่างเช่น ในการพิมพ์งานจะมีคีย์บอร์ดเป็นอินเตอร์เฟซซึ่งใช้ในการสร้างตัวอักษรในโปรแกรมเวิร์ดโปรเซสเซอร์ ส่วนการทำงานภายในที่ทำการแปลงจากการกดคีย์บอร์ดไปเป็นตัวอักษรบนหน้าจอจะถูกซ่อนอยู่ภายใต้อินเตอร์เฟซ

##### 2.2.3 การสืบทอด (Inheritance)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2.4 โพลิมอร์ฟิซึม (Polymorphism)

โพลิมอร์ฟิซึมเป็นความสามารถของออบเจกต์ในการทำงานเมื่อได้รับคำสั่งเดียวกันจาก โปรแกรม โดยออบเจกต์จะทำงานตามแบบของตัวเอง ตัวอย่างเช่น ออบเจกต์คนกับออบเจกต์สุนัขสามารถที่จะเดินได้แต่วิธีการเดินของทั้งสองออบเจกต์จะแตกต่างกัน เป็นต้น การที่ออบเจกต์จะสามารถทำงานได้ในลักษณะนี้ คลาสต้นแบบจะต้องประกาศเมธอด (Method) ที่จะใช้เป็นเวอร์ชวลเมธอด

### 2.2.4.1 เมธอดแบบสแตติก (Static Method)

เมธอดแบบสแตติกมีลักษณะการทำงานคล้ายกับการทำงานของ โพรซีเจอร์และฟังก์ชันทั่วไปใน ภาษาปาสคาล กล่าวคือ เมื่อเมธอดเหล่านี้เรียกใช้เมธอดอื่น ก็จะมีการเรียกใช้เมธอดเดิมนั้นเสมอ ที่เป็นเช่นนี้ก็เพราะได้มีการผูกติดกันตั้งแต่ตอนคอมไพล์โปรแกรมซึ่งเรียกว่า เออร์ลีไบน์ดิง (Early binding) ผลของเออร์ลีไบน์ดิงก็คือ เมธอดที่ทำการเรียกเมธอดนั้นจากที่ใดและตำแหน่งการเรียกเมธอดนั้นจะคงที่ไม่สามารถเปลี่ยนแปลงได้

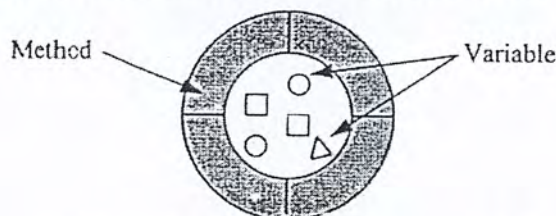
### 2.2.4.2 เมธอดแบบเวอร์ชวล (Virtual Method)

เมธอดแบบนี้จะไม่มีการผูกติดกับเมธอดอื่นตั้งแต่ตอนคอมไพล์โปรแกรม แต่จะมีการผูกติดกัน ตอนรันโปรแกรม ซึ่งเรียกว่า เลตไบน์ดิง (Late binding) ผลของเลตไบน์ดิงก็คือ เมธอดที่ทำการเรียก เมธอดอื่นที่ไม่รู้ว่าจะเรียกเมธอดนั้นจากที่ไหนแต่จะรู้ในตอนรันโปรแกรม

## 2.3 ออบเจกต์ (Object)

ออบเจกต์คือสิ่งที่เกิดจากการรวมกันของข้อมูลและวิธีการจัดการกับข้อมูลนั้น ออบเจกต์เป็นส่วนสำคัญในการเขียนโปรแกรมแบบออบเจกต์ โดยตัวอย่างของออบเจกต์ในโลกแห่งความจริง เช่น คนเป็น สิ่งมีชีวิตที่มีคุณสมบัติ เช่น ความสูงและน้ำหนัก เป็นต้น และมีพฤติกรรม เช่น สามารถเดินได้และสามารถวิ่งได้ เป็นต้น สำหรับการโปรแกรมเชิงออบเจกต์นั้นเราสามารถที่จะแทนคุณสมบัติด้วยตัวแปร (Variable) และแทนพฤติกรรมด้วยเมธอด

จากรูปที่ 2-1 เห็นได้ว่าตัวแปรของออบเจกต์ถูกจัดให้อยู่ตรงกลางของวงกลมออบเจกต์และมีเมธอด ล้อมรอบเพื่อทำหน้าที่ซ่อนส่วนกลางของออบเจกต์ไว้ ประโยชน์ของการซ่อนรายละเอียดก็คือ ถ้าหาก ต้องการเปลี่ยนแปลงข้อมูลภายในออบเจกต์ ก็สามารถกระทำได้โดยไม่ส่งผลกระทบต่อออบเจกต์อื่นๆ และประโยชน์อีกข้อหนึ่ง ก็คือ สามารถจัดการกับข้อมูลภายในออบเจกต์ได้ว่าข้อมูลส่วนไหนต้องการที่จะเปิดเผยหรือไม่ ซึ่งสามารถกำหนดได้โดยประกาศคุณสมบัติให้กับออบเจกต์

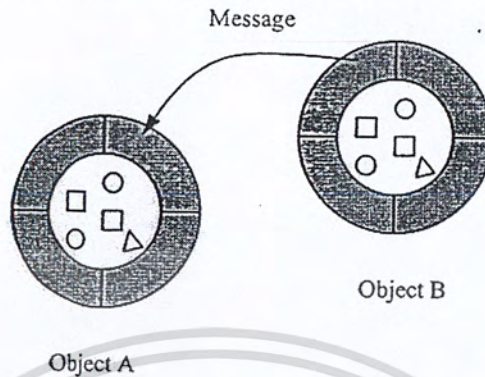


รูปที่ 2-1 การจำลองส่วนประกอบของออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 ข่าวสาร (Message)

การติดต่อสื่อสารกันระหว่างออบเจกต์สามารถทำได้โดยการรับส่งข่าวสาร ตัวอย่างเช่น เมื่อออบเจกต์ A ต้องการทำงานกับออบเจกต์ B ออบเจกต์ A ก็จะส่งข่าวสารไปบอกแก่ออบเจกต์ B ดังรูปที่ 2-2



รูปที่ 2-2 การส่งข่าวสารระหว่างออบเจกต์

## 2.5 คลาสและอินสแตนซ์ (Class and Instance)

คลาสคือชนิดของออบเจกต์ซึ่งจะแสดงถึงคุณสมบัติและคุณลักษณะของออบเจกต์ คลาสจะถูกสร้างขึ้นโดยการเขียนรายละเอียดของคลาส ซึ่งคลาสหนึ่งสามารถที่จะเป็นต้นแบบให้แก่ออบเจกต์ได้หลายออบเจกต์ การที่เราสร้างออบเจกต์ขึ้นมา เราเรียกว่าอินสแตนซ์ (Instance) โดยเราต้องทำการสร้างอินสแตนซ์ของคลาสนั้นมาก่อนที่จะมีการใช้ตัวแปรหรือเมธอดของคลาสนั้น ซึ่งจะเรียกอินสแตนซ์ว่าเป็นออบเจกต์เมื่อมีการเปลี่ยนแปลงข้อมูลหรือเรียกใช้เมธอดของอินสแตนซ์นั้นเกิดขึ้น



รูปที่ 2-3 รายละเอียดของคลาส

## 2.6 ตัวอย่างการออกแบบออบเจกต์

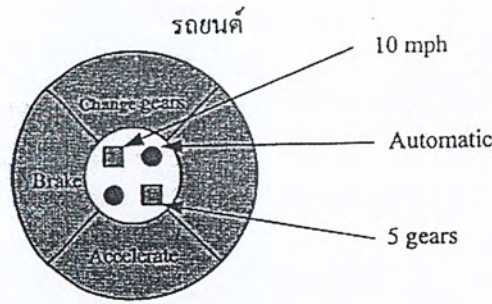
ในตัวอย่างนี้จะทำการออกแบบออบเจกต์รถยนต์ที่มีการกำหนดตัวแปรดังต่อไปนี้  
สถานะของรถยนต์

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

เมธอดของรถยนต์

1. การเบรก
2. การเร่งความเร็ว
3. การเปลี่ยนเกียร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



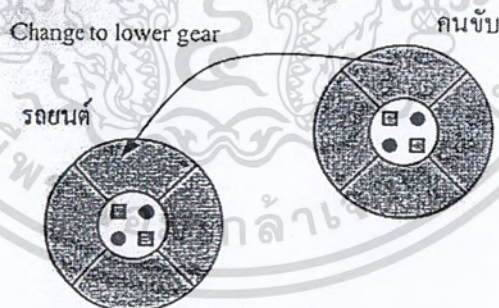
รูปที่ 2-4 ไดอะแกรมของตัวอย่างการออกแบบ

จากรูปที่ 2-4 ตัวแปรของออบเจกต์จะอยู่ตรงส่วนกลางของออบเจกต์ โดยเมธอดจะอยู่รอบ ๆ ตัวแปรของออบเจกต์และซ่อนตัวแปรของออบเจกต์จากออบเจกต์อื่นๆ เช่น เมื่อเราต้องการที่จะเปลี่ยนเกียร์ของรถยนต์ เราก็ไม่จำเป็นต้องรู้ระบบของเกียร์ทำงานอย่างไรบ้าง เพียงแต่รู้วิธีการที่จะเลื่อนคันโยกสำหรับเปลี่ยนเกียร์ก็พอแล้ว

### เมสเสจของรถยนต์

ในการทำงานของออบเจกต์ ออบเจกต์จะต้องได้รับข้อมูลที่เพียงพอเพื่อที่ออบเจกต์จะรู้ว่าต้องทำอะไร จากตัวอย่างของการออกแบบ เมื่อต้องการเปลี่ยนเกียร์จะต้องทราบว่าเปลี่ยนเกียร์เป็นเกียร์อะไร โดยข้อมูลเหล่านี้จะถูกส่งไปกับเมสเสจเหมือนพารามิเตอร์ ส่วนประกอบของเมสเสจประกอบด้วย

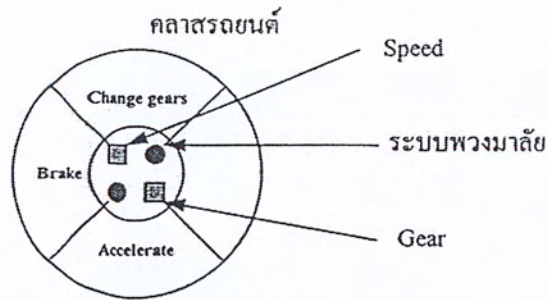
1. ออบเจกต์ที่จะส่งเมสเสจไปให้ (รถยนต์)
2. ชื่อของเมธอดที่จะทำ (เปลี่ยนเกียร์)
3. พารามิเตอร์ต่างๆ ที่เมธอดต้องใช้



รูปที่ 2-5 การส่งเมสเสจที่มีพารามิเตอร์

### คลาสของรถยนต์

เราต้องสร้างคลาสของรถยนต์โดยประกาศตัวแปรเพื่อใช้เก็บสถานะต่าง ๆ เช่น ความเร็วของรถยนต์ สถานะของเกียร์ปัจจุบันและระบบของพวงมาลัย เป็นต้น และสร้างเมธอดต่าง ๆ ที่ใช้ในการทำงานของออบเจกต์ เช่น การเบรก การเร่งความเร็ว และการเปลี่ยนเกียร์ เป็นต้น

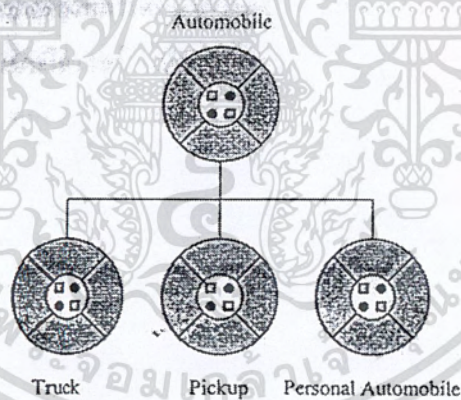


รูปที่ 2-6 คลาสไดอะแกรมของรถยนต์

ค่าของตัวแปรจะถูกกำหนดโดยอินสแตนซ์ของคลาส ดังนั้นหลังจากที่ทำการสร้างคลาสของรถยนต์แล้วจะต้องทำการสร้างอินสแตนซ์ของคลาสด้วย ถ้าทำการสร้างอินสแตนซ์ของคลาสแล้วตัวแปรที่ประกาศไว้โดยคลาสจะต้องเก็บไว้ในหน่วยความจำ จากนั้นจะสามารถใช้เมธอดเพื่อกำหนดค่าให้ตัวแปรได้

## 2.7 การสืบทอด

ในการเขียนโปรแกรมแบบออบเจกต์จะมีการสืบทอดคลาสอย่างเป็นทางการเป็นลำดับชั้นและกำหนดให้คลาสหนึ่งเป็นคลาสต้นแบบของคลาสอื่นๆได้ เช่น รถกระบะ รถบรรทุกและรถยนต์ส่วนบุคคลเป็นคลาสที่ทำการสืบทอดมาจากคลาสรถยนต์ โดยจะเรียนคลาสทั้งสามแบบเป็นสับคลาส (Sub Class) และเรียกคลาสรถยนต์ว่าเป็นซูเปอร์คลาส (Super class)



รูปที่ 2-7 การสืบทอดคลาสรถยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### Component Object Model (COM)

##### 3.1 ซอฟต์แวร์คอมโพเนนต์

เป็นกลุ่มของ คลาสซึ่งอยู่ร่วมกันและอยู่ในรูปแบบที่ใช้งานได้โดยไม่ต้องทำการคอมไพล์ใหม่ (อาจจะ เป็นไบนารี หรือเป็น ไฟล์ DLL) ติดต่อกับระบบภายนอกโดยทางอินเตอร์เฟส ซึ่งมีลักษณะคล้าย แอป แทรคคลาส (Abstract Class) ซึ่งเป็นคลาสที่ถูกประกาศเพียงแต่เมธอดไม่มีการอิมพลิเมนต์ เมธอดภายใน และไม่มีการประกาศคุณสมบัติ (Property) ต่างๆ ของคลาสนั้นด้วย

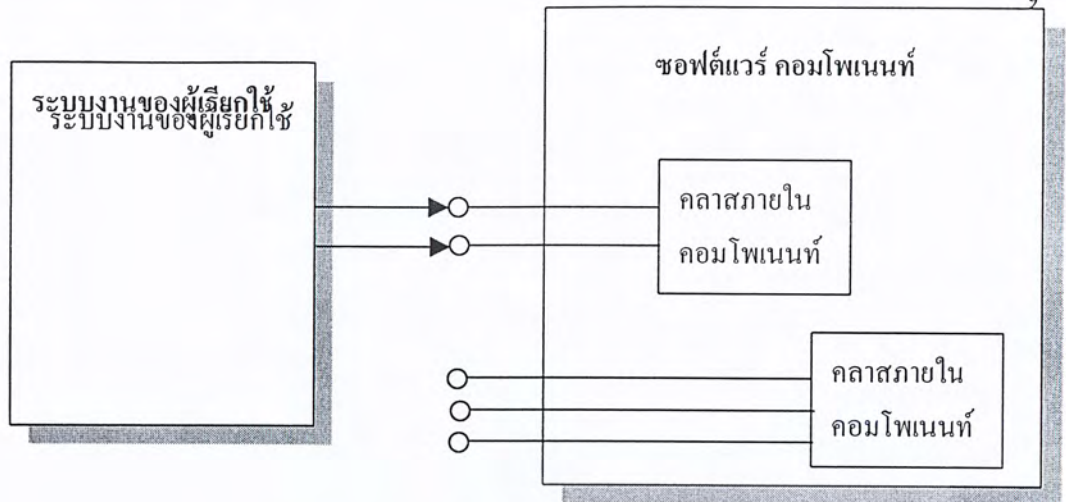
สำหรับ COM แล้ว อินเตอร์เฟสนั้นเป็นองค์ประกอบสำคัญในการสร้างความเป็นอิสระระหว่างผู้เรียกใช้ กับคอมโพเนนต์ ตัวอินเตอร์เฟสนั้นจะถูกประกาศโดยใช้ภาษาเป็นมาตรฐานในการสร้างอินเทอร์เฟซซึ่งเรียกว่า Interface Definition Language (IDL) ซึ่งทำให้การพัฒนา COM คอมโพเนนต์นั้นสามารถพัฒนาจาก ภาษาโปรแกรมภาษาใด ๆ ก็ได้ อย่างไรก็ตามเนื่องจากโครงสร้างพื้นฐานของ COM คอมโพเนนต์ที่จำเป็น ต้องมีการเรียกใช้ผ่านทางพอยต์เตอร์ ดังนั้นภาษาที่จะสร้าง และเรียกใช้งานก็ควรจะมีความสามารถที่จะ เรียกใช้พอยต์เตอร์เหล่านั้นได้ ซึ่งไม่ได้หมายความว่าภาษาเหล่านั้นจะต้องมีการเรียกใช้ตัวแปรพอยต์เตอร์ ได้โดยตรง เช่น Visual Basic นั้นไม่มีความสามารถที่จะเรียกใช้ตัวแปร พอยต์เตอร์ได้โดยตรง แต่ก็ สามารถที่จะสร้างและเรียกใช้ COM คอมโพเนนต์ ได้ เนื่องจากตัวภาษานั้นสามารถเรียกใช้ได้เพียงแต่ ไม่สามารถเรียกใช้ได้โดยผ่านทางตัวแปรพอยต์เตอร์เท่านั้น

ในการพัฒนาคอมโพเนนต์ ก็จะทำให้คลาสภายในคอมโพเนนต์นั้นทำการอิมพลิเมนต์ แอปแทรคคลาสหรือ อินเตอร์เฟสนั้น และฝ่ายผู้ใช้คอมโพเนนต์ นั้นก็จะติดต่อกับคลาสภายในคอมโพเนนต์ผ่านทางอินเตอร์เฟส ดังนั้นเมื่อมีการเปลี่ยนแปลงลักษณะภายในของคลาสในคอมโพเนนต์ก็จะมีผลกระทบต่อผู้ เรียกใช้ เนื่องจาก การเรียกใช้คอมโพเนนต์ก็ยังเรียกผ่านทางอินเตอร์เฟสที่ยังไม่เปลี่ยนแปลง ซึ่งก็เป็นเหตุ ผลสำคัญที่ทำให้การกำหนดอินเตอร์เฟสนั้น ไม่ควรมีการเปลี่ยนแปลง (immutable)

เมื่อผู้เรียกใช้ต้องการเปลี่ยนคอมโพเนนต์ จากตัวเดิมที่ใช้อยู่ไปเป็นตัวใหม่ก็สามารถเปลี่ยนได้โดยไม่ต้อง แก่ซอร์สโค้ด เพียงแต่คอมโพเนนต์ตัวใหม่นั้นจะต้องมีอินเตอร์เฟสเหมือนคอมโพเนนต์ตัวเก่าเท่านั้น ซึ่ง ทำให้การพัฒนาโปรแกรมนั้นเป็นอิสระต่อกันมากกว่าการพัฒนาโปรแกรมแบบเดิม

แต่ละคลาสนั้นสามารถอิมพลิเมนต์อินเตอร์เฟสได้มากกว่าหนึ่งอินเตอร์เฟส ซึ่งจะแตกต่างจากที่คลาสต่าง ๆ นั้นสามารถสืบทอดคุณสมบัติได้จากคลาสใดคลาสหนึ่ง เนื่องจากหลักการ เขียนโปรแกรมเชิงคอมโพ เนนต์นั้นไม่สนับสนุนการสืบทอดคุณสมบัติมากกว่าหนึ่ง (Multiple Inheritance) แต่สนับสนุนหลักการ การมีหลายอินเตอร์เฟส (Multiple Interface)

จากที่ได้กล่าวมาเบื้องต้น ลักษณะอย่างคร่าว ๆ ของซอฟต์แวร์คอมโพเนนต์จะมีลักษณะดังรูปที่ 3-1



รูปที่ 3-1 แสดงลักษณะของซอฟต์แวร์คอมโพเนนท์

### 3.2 COM คอมโพเนนท์

COM เป็นซอฟต์แวร์คอมโพเนนท์โมดูลที่เป็นอิสระทางภาษาไม่ขึ้นกับภาษาที่ใช้พัฒนา เราสามารถสร้างออบเจกต์ (Object) แล้วใช้โดยภาษาอื่นๆ ได้หลายภาษา สามารถใช้ภายในเครื่องเดียวกัน หรือใช้ผ่านเครือข่ายมาจากเครื่องอื่นก็ได้

ลักษณะที่เห็นได้ชัดของ COM ก็คือการสื่อสารระหว่างคอมโพเนนท์, ระหว่างแอปพลิเคชัน และระหว่างไคลเอนต์กับเซิร์ฟเวอร์ ผ่านทางอินเตอร์เฟส (Interface) แอปพลิเคชันต่างๆ สามารถเข้าถึง COM คอมโพเนนท์และอินเตอร์เฟสของมันที่มีได้ในเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรือเครื่องคอมพิวเตอร์ที่มีอยู่ในเครือข่ายได้ ในกรณีที่ใช้ผ่านเครือข่ายนั้นจะใช้ Distribute COM (DCOM)

ในการใช้งาน COM มีส่วนประกอบอยู่ 3 ส่วนคือ

1. COM เซิร์ฟเวอร์ เป็นตัวให้บริการ ในการทำงานไคลเอนต์จะเรียกของการทำงานจาก COM เซิร์ฟเวอร์โดยผ่านทางอินเตอร์เฟส
2. COM อินเตอร์เฟส เป็นตัวกลางในการติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์
3. COM ไคลเอนต์ เป็นตัวเรียกใช้การทำงานของ COM เซิร์ฟเวอร์

#### 3.2.1 COM เซิร์ฟเวอร์

COM เซิร์ฟเวอร์ เป็นแอปพลิเคชัน หรือไลบรารี (Library) ที่เตรียมบริการให้แก่ไคลเอนต์ ใน COM เซิร์ฟเวอร์หนึ่งตัวประกอบไปด้วยหนึ่ง หรือหลายๆ COM ออบเจกต์

ไคลเอนต์ไม่จำเป็นต้องรู้วิธีการให้บริการ COM ออบเจกต์ และไม่ต้องรู้ว่า COM ออบเจกต์อยู่ที่ใด ตัว COM เองจะจัดการในการเข้าถึงออบเจกต์

เมื่อไคลเอนต์ร้องขอบริการจาก COM ออบเจกต์ ตัวไคลเอนต์จะส่ง Class identifier (CLSID) ไปยัง COM จากนั้น COM จะใช้ CLSID ในการเตรียมที่ที่เหมาะสมในการนำโค้ด (Code) เข้ามาไว้ในหน่วยความจำ แล้วทำการกำหนดค่าเริ่มต้นให้แก่ออบเจกต์อินสแตนซ์เพื่อไคลเอนต์ในการนำไปใช้ ส่วนตัว

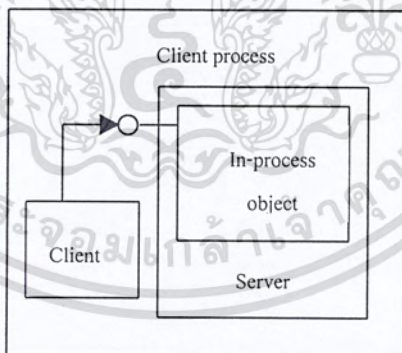
COM เซิร์ฟเวอร์ต้องทำการลงทะเบียน (Register) ด้วย CLSID เข้าในระบบ แล้วเตรียม Class factory

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจ็กต์ (IClassFactory) เพื่อการสร้างอินสแตนซ์ที่ต้องการตาม CLSID และเตรียมวิธีในการนำอินสแตนซ์ออกจากหน่วยความจำเมื่อไคลเอนต์ไม่ใช้อินสแตนซ์นั้นแล้ว

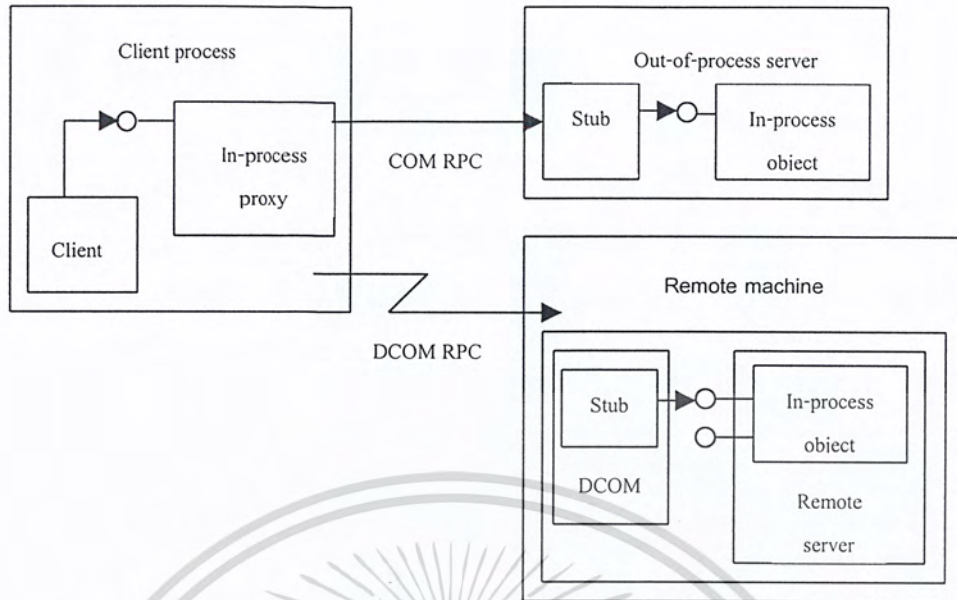
### ประเภทของ COM เซิร์ฟเวอร์มี 3 แบบดังนี้

- **In-process server** เป็นไลบรารี (DLL) จะรัน (Run) ภายในโพรเซส (Process) เดียวกันกับไคลเอนต์ เช่น แอ็กทีฟเอ็กซ์คอนโทรล (ActiveX control) ที่ฝังในเว็บเพจที่สามารถแสดงผลทางอินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet explorer) หรือ เน็ตส์เคป (Netscape) โดยแอ็กทีฟเอ็กซ์คอนโทรลจะถูกดาวน์โหลดมาไว้ที่ไคลเอนต์และทำงานในโพรเซสเดียวกันกับเว็บเบราว์เซอร์ ในการติดต่อของไคลเอนต์กับ In-process server จะเรียกใช้ไปโดยตรงไปยัง COM อินเทอร์เฟซ
- **Out-of-process server** หรือ Local server เป็นแอปพลิเคชัน (EXE) ทำการรันในคนละโพรเซส แต่อยู่ในเครื่องเดียวกัน เช่น เวิร์ด มี เอ็กเซล ฝังอยู่ในทั้งสองแอปพลิเคชันรันในเครื่องเดียวกันแต่คนละโพรเซส Out-of-process server จะใช้ COM ในการติดต่อกับไคลเอนต์
- **Remote server** เป็นไลบรารี หรือแอปพลิเคชันก็ได้ รันอยู่บนคนละเครื่องกับไคลเอนต์ เช่น ในการติดต่อของ ดาตาเบสแอปพลิเคชัน กับ แอปพลิเคชันเซิร์ฟเวอร์ที่อยู่กันคนละเครื่องในเครือข่าย Remote server จะใช้ Distribute COM (DCOM) ในการติดต่อกับไคลเอนต์



รูปที่ 3-2 แสดง In-process server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 แสดง Out-of-process server และ Remote server

ทั้งในแบบ Out-of-Process Server และ Remote Server จะต้องมีวิธีการพิเศษซึ่งเรียกว่า RPC (Remote Procedure Call) ซึ่งใช้ในการเรียกใช้โปรแกรมเมอร์ โดยผ่านทางพร็อกซี ซึ่งอยู่ภายนอกโปรเซส แต่เป็น in-process ในมุมมองของ ไคลเอนต์ ความแตกต่างระหว่างตำแหน่งทั้งสองคือ ใน Out-of-Process นั้นจะใช้ COM เป็นตัวกลางในการติดต่อ ส่วน ในแบบ Remote machine จะใช้ DCOM (Distributed COM) เป็นตัวส่งผ่าน RPC นั้นให้กับตัวเซิร์ฟเวอร์

### 3.2.2 COM อินเทอร์เฟซ



รูปที่ 3-4 แสดง ลักษณะของ COM

COM ไคลเอนต์สื่อสารกับออบเจกต์ผ่านทาง COM อินเทอร์เฟซ ซึ่งเป็นกลุ่มของรูทีนเอง (Routine) ที่ช่วยในการติดต่อระหว่างผู้ให้บริการ (Server Object) และผู้ขอใช้บริการ (Client) ภายในอินเทอร์เฟซ จะไม่มีโค้ด แต่จะมีเพียง รายเอ็ดย่างคร่าวๆ ของ แต่ละโปรเซส ซึ่งได้แก่ ชื่อ เมธอด และออบเจกต์ แต่ละตัวสามารถมี อินเทอร์เฟซ ได้หลายตัวซึ่ง ทุกตัวต้องประกอบด้วย อินเทอร์เฟซ อย่างน้อย 2 ตัว หนึ่งคือ อินเทอร์เฟซ พื้นฐานซึ่งเป็นตัวที่ ออบเจกต์ทุกตัวต้องมีคือ IUnknown ซึ่งมี เมธอด พื้นฐานคือ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QueryInterface (), AddRef () และ Release () และอินเทอร์เฟซ ที่เหลือคือ อินเทอร์เฟซซึ่งผู้พัฒนาพัฒนาให้อยู่ในรูปแบบซึ่งตัวเองต้องการ อินเทอร์เฟซ เหล่านี้จะสืบทอดมาจาก IUnknown

โดยอินเทอร์เฟซมีลักษณะดังนี้

- สร้างขึ้นมาแล้วจะแก้ไขหรือเปลี่ยนอินเทอร์เฟซไม่ได้ ในแต่ละอินเทอร์เฟซจะเตรียมเซตของฟังก์ชันให้เราเรียกใช้ได้ การจะเพิ่มฟังก์ชันทำได้โดยการเพิ่มอินเทอร์เฟซเข้าไปใหม่
- ชื่อของ COM อินเทอร์เฟซจะขึ้นต้นด้วย “I” ตัวใหญ่แล้วตามด้วยชื่อของอินเทอร์เฟซนั้น
- ในแต่ละอินเทอร์เฟซนั้นจะมีหมายเลขที่รับรองว่าไม่มีการซ้ำเรียกว่า Globally Unique Identifier (GUID) มีขนาด 128 บิต หรือ 32 ไบต์
- อินเทอร์เฟซเป็นอิสระทางภาษาคือจะใช้ภาษาใดก็ได้ในการสร้างแต่ภาษาที่ใช้ต้องสนับสนุนการใช้พอยน์เตอร์และสามารถเรียกฟังก์ชันผ่านทางพอยน์เตอร์ได้
- อินเทอร์เฟซไม่ใช่ออบเจกต์ แต่เป็นทางที่จะเข้าถึงออบเจกต์ได้ ตัวโคลเอนต์ไม่สามารถเข้าถึงข้อมูลได้โดยตรงแต่จะเข้าถึงผ่านอินเทอร์เฟซพอยน์เตอร์
- อินเทอร์เฟซต้องได้รับการสืบทอดมาจาก IUnknown เสมอ

### 3.2.1.1 ลักษณะของ IUnknown

ทุกๆ COM ออบเจกต์ต้องมีอินเทอร์เฟซพื้นฐานที่ชื่อ IUnknown ซึ่งภายในมีเมธอดดังนี้

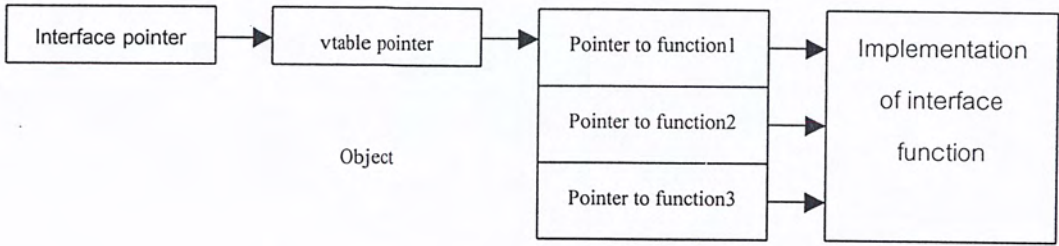
- **QueryInterface** ทำหน้าที่เตรียมพอยน์เตอร์ชี้ไปยังอินเทอร์เฟซที่ออบเจกต์นั้นสนับสนุนอยู่
- **AddRef** และ **Release** ทำหน้าที่ในการนับโดยจะเก็บช่วงชีวิตของออบเจกต์แล้วคอยดูว่าออบเจกต์ตัวใดสามารถถูกลบออกไปได้ถ้าไม่ได้ถูกใช้เป็นเวลานานๆ

ในการทำงาน โคลเอนต์ได้รับพอยน์เตอร์ไปยังอินเทอร์เฟซต่างๆ ผ่าน IUnknown โดยใช้เมธอด QueryInterface ตัว QueryInterface นั้นจะรู้ข้อมูลเกี่ยวกับทุกๆ อินเทอร์เฟซในเซิร์ฟเวอร์แล้วจะให้พอยน์เตอร์แก่โคลเอนต์ในการเรียกใช้เมธอดจากอินเทอร์เฟซ

ในการนับช่วงชีวิตของออบเจกต์ก็จะใช้ AddRef และ Release ในการนับออบเจกต์ที่มีค่าอ้างอิงไม่เป็นศูนย์จะอยู่ในหน่วยความจำและถ้าค่าอ้างอิงเป็นศูนย์เมื่อไหร่ออบเจกต์นั้นก็จะถูกนำออกไป

พอยน์เตอร์ของอินเทอร์เฟซจะมีขนาด 32 บิต ชี้ไปยังอินสแตนซ์ (Instance) ของออบเจกต์ เพื่อการใช้เมธอดที่มีในอินเทอร์เฟซ โดยการทำงานจะใช้อาร์เรย์ของพอยน์เตอร์ชี้ไปยังเมธอด เรียกว่า vtable

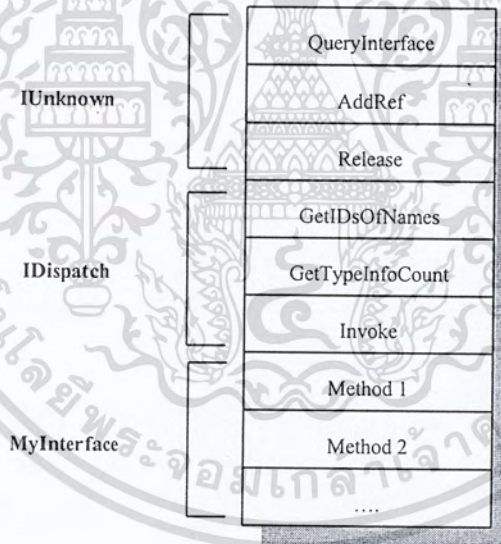
vtable จะถูกใช้ร่วมกันโดยทุกๆ อินสแตนซ์ของออบเจกต์คลาส และพอยน์เตอร์ของอินเทอร์เฟซจะชี้ไปยัง vtable ดังรูปที่ 3-5



รูปที่ 3-5 แสดง ลักษณะของ vtable

3.2.3 คู่อินเตอร์เฟส (Dual Interface)

เป็นอีกรูปแบบหนึ่งของการเรียกใช้อินเตอร์เฟส สำหรับโดยคอมโพเนนต์ที่จะมีลักษณะเป็น Dual Interface นี้จะเป็นอินเตอร์เฟสที่มีอินเตอร์เฟส IDispatch เพิ่มขึ้นอีกอินเตอร์เฟสหนึ่งอินเตอร์เฟส ดังรูปที่ 3-6 จะแสดงลักษณะอินเตอร์เฟส IMyInterface ซึ่งมีลักษณะเป็น Dual Interface ของ COM คอมโพเนนต์หนึ่ง จะเป็นว่า 3 เมธอดแรกนั้นเป็นเมธอดของอินเตอร์เฟส IUnknown ส่วนอีก 4 เมธอดถัดไปนั้นเป็นของอินเตอร์เฟส IDispatch หลังจากนั้นจึงเป็นเมธอดของอินเตอร์เฟส IMyInterface



รูปที่ 3-6 แสดงลักษณะของอินเตอร์เฟสที่เป็น คู่อินเตอร์เฟส

ในการเรียกใช้งาน COM คอมโพเนนต์ประเภทนี้นั้นจะสามารถเรียกใช้งานได้ใน 2 รูปแบบเนื่องจากอินเตอร์เฟสที่เป็นคู่อินเตอร์เฟสนั้นจะมีลักษณะทั้งสองอย่างอยู่ในตัวมัน ซึ่งได้แก่

3.2.3.1 )เรียกใช้งานแบบเออิลิบาย (Early Binding) ที่เรียกใช้ตอนคอมไพล์ใหม่โดยผ่านการเรียกไปยังเมธอดของ COM คอมโพเนนต์โดยตรงผ่านทาง VTable หรือคัสตอมอินเตอร์เฟส (Custom Interface) ซึ่งเป็นลักษณะพื้นฐานของ COM คอมโพเนนต์ทั่วไปซึ่งมีข้อดีกว่าการเรียกอีกรูปแบบหนึ่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเรียกใช้ผ่าน VTable นั้นจะมีการตรวจสอบชื่อเมธอด และพารามิเตอร์ซึ่งทำให้สามารถควบคุมการเกิดความผิดพลาดเบื้องต้นได้
- เข้าถึง COM คอมโพเนนต์ได้เร็วกว่าสำหรับเซิร์ฟเวอร์ชนิดอินโพรเซสเซิร์ฟเวอร์นั้น

นอกจากสำหรับ COM คอมโพเนนต์ทั่วไปหากไม่สนับสนุนอินเตอร์เฟส IDispatch การเรียกใช้งานจะเรียกโดยใช้วิธีโดยอัตโนมัติ

3.2.3.2 )เรียกใช้งานแบบเลทบาย (Late Binding) ซึ่งจะทำการเรียกใช้ตอนรันไทม์โดยผ่านทางอินเตอร์เฟส IDispatch ในการเรียกใช้ผู้เรียกใช้ต้องทำการสร้างออบเจกต์ที่เรียกอินเตอร์เฟส IUnknown เสียก่อนหลังจากนั้นจึงเรียกไปยังอินเตอร์เฟส IDispatch อีกที IDispatch จะเก็บคุณสมบัติและเมธอดต่าง ๆ ของมันไว้ภายในโดยใช้ ตัวบ่งชี้ dispatch (dispID) ซึ่งเป็น GUID การจะเข้าถึงชื่อของเมธอด และคุณสมบัติต่าง ๆ ของอินเตอร์เฟสเหล่านั้นได้ โดยผ่านตัวบ่งชี้ ซึ่งจะถูกสร้างเมื่ออยู่ในขณะรันไทม์ ร่วมกับเมธอด GetIDsOfNames ส่วนการเรียกใช้จะใช้เมธอด Invoke ข้อดีของการใช้วิธีนี้ก็คือตัวไคลเอนต์จะเป็นอิสระต่อตัวคอมโพเนนต์เนื่องจาก จะเรียกใช้จริง ๆ ตอนรันไทม์เท่านั้น ในการทำงานจริงหากมีการเปลี่ยนแปลง ตัวคอมโพเนนต์จึงไม่จำเป็นต้องคอมไพล์ตัวไคลเอนต์ใหม่ (อย่างไรก็ดีต้องไม่มีการเปลี่ยนแปลงอินเตอร์เฟสเก่า แต่สามารถเพิ่มเมธอดใหม่ เข้าไปในคอมโพเนนต์ใหม่ได้ ) นอกจากนี้การเรียกใช้ COM คอมโพเนนต์ในผลิตภัณฑ์ Microsoft Visual Basic จะเป็นการเรียกโดยวิธีนี้เท่านั้น

### 3.2.4 โคลาส ( Coclases ) และ คลาสแฟกทอรี (Class factory)

COM ออบเจกต์ เป็นอินสแตนซ์ของโคลาส ตัวโคลาสเองจะมีออบเจกต์ที่เรียกคลาสแฟกทอรีเมื่อไรที่มีการร้องขอบริการจากไคลเอนต์ตัวคลาสแฟกทอรี จะสร้าง และลงทะเบียนออบเจกต์อินสแตนซ์ให้กับไคลเอนต์ที่ทำการร้องขอมา และถ้ามีไคลเอนต์ตัวอื่นต้องการบริการอีกคลาสแฟกทอรี ก็จะทำการสร้างให้อีก

### 3.2.5 มาร์แชลลิ่งแมกนานิซึม ( Marshaling mechanism )

เป็นวิธีการที่ไคลเอนต์ทำการเรียกไปยังรีโมตออบเจกต์ในโพรเซสอื่น หรือบนคอมพิวเตอร์คนละเครื่อง วิธีการมาร์แชลลิ่งมีดังนี้

- ใช้อินเตอร์เฟสพอยน์เตอร์ในโพรเซสของเซิร์ฟเวอร์ และสร้างพ็อกซีพอยน์เตอร์ชี้ไปยังโค้ดในไคลเอนต์โพรเซส
- ส่งอาร์กิวเมนต์ (Argument) ของการเรียกอินเตอร์เฟสจากไคลเอนต์ และวางอาร์กิวเมนต์ลงในโพรเซสของรีโมตออบเจกต์

สำหรับการเรียกอินเตอร์เฟส ตัวไคลเอนต์จะทำการวาง (Push) อาร์กิวเมนต์ลงในสแตค และทำการเรียก

ฟังก์ชันผ่านทางอินเตอร์เฟสพอยน์เตอร์ ถ้าการเรียกออบเจกต์ไม่เป็นแบบ In-process server การเรียกจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านไปยังพรอกซี ตัวพรอกซีจะทำการห่อ (Packs) อาร์กิวเมนต์ให้เป็น มาร์แชลลิงแพ็คเกจ แล้วทำการส่ง ไปยังรีโมตออบเจกต์ หลังจากนั้น Stub จะทำการแกะห่อ (Unpacks) แล้ววางอาร์กิวเมนต์ลงในสแตกแล้ว จึงเรียกใช้ออบเจกต์

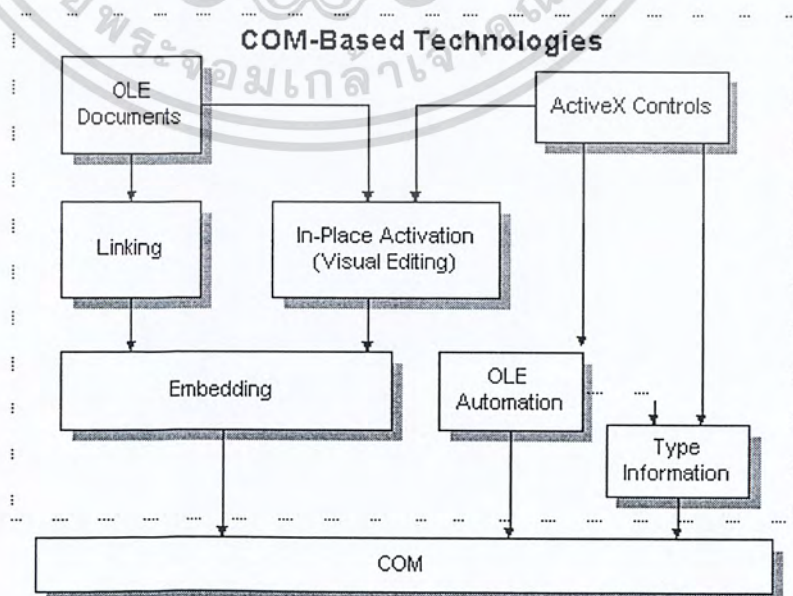
### 3.3 COM ไคลเอนต์

เป็นผู้ขอการบริการจาก COM เซิร์ฟเวอร์ ผ่านทาง COM อินเตอร์เฟส นับเป็นส่วนที่สำคัญส่วน หนึ่งในการออกแบบ COM แอปพลิเคชัน สิ่งที่สำคัญของส่วนนี้คือ ไคลเอนต์นั้นไม่ต้องการรู้ว่า COM เซิร์ฟเวอร์ นั้นทำงานอย่างไร และอยู่ที่ไหนขั้นตอนเหล่านั้น ทั้งอินเตอร์เฟส และ เซิร์ฟเวอร์ต้องปกปิด เพื่อให้ไคลเอนต์ นั้นสะดวกในการนำไปใช้

### 3.4 ส่วยขยาย COM

COM ได้ออกแบบให้มีการติดต่อกันของฟังก์ชัน และขยายการติดต่อกับฟังก์ชันออกไปในวงกว้าง ตัว COM เองได้มีการขยายออกไปเฉพาะทางตามความต้องการในการใช้งาน เช่น อย่างแรกที่ต้องพูดถึงคือ เทคโนโลยี OLE และบริการหลายๆ อย่างของวินโดวส์

- ออโตเมชัน เซิร์ฟเวอร์ เป็นความสามารถของแอปพลิเคชันในการควบคุมออบเจกต์ใน แอปพลิเคชันตัวอื่น ในทางโปรแกรมมิ่งพูดได้ว่า ออโตเมชัน เซิร์ฟเวอร์ เป็นออบเจกต์ที่ สามารถควบคุม โดยเอ็กซีคิวทีเบิล (EXE) ในเวลารัน
- ออโตเมชัน คอนโทรลเลอร์ (COM clients) เป็นไคลเอนต์ของ ออโตเมชัน เซิร์ฟเวอร์ ผู้ใช้ สามารถเขียนสคริปต์ทำการควบคุม ออโตเมชัน เซิร์ฟเวอร์ ได้
- ActiveX controls เป็นแบบเฉพาะของ In-process COM เซิร์ฟเวอร์ โดยจะฝังใน แอปพลิเคชันของไคลเอนต์
- Type libraries เป็น โครงสร้างข้อมูลแบบหนึ่งใช้เก็บทรัพยากร (Resource) บอกรายละเอียด ของรูปแบบเกี่ยวกับออบเจกต์และอินเตอร์เฟสต่างๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3-7 แสดง แสดงส่วนขยาย COM  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Active Server Pages** เป็น ActiveX คอมโพเนนท์ในการสร้าง ไดนามิกเว็บเพจ
- **Active Document** เป็นออบเจกต์ที่สนับสนุน Linking และ Embedding , Drag และ Drop , Visual editing ตัวอย่างเช่น ผลึกภัณฑ์ไมโครซอฟต์เวิร์ด และ ไมโครซอฟท์เอ็กเซล
- **Visual Cross-Process Object** เป็นออบเจกต์ซึ่งสามารถถ่ายโอนข้ามโพรเซส

### 3.4.1 Automation sever and controller

อโโตเมชัน อ้างถึงความสามารถของแอปพลิเคชันในการควบคุมแอปพลิเคชันอื่น เหมือนกับมาโครที่สามารถใช้กับแอปพลิเคชันหลายๆ แอปพลิเคชันในเวลาเดียวกัน อโโตเมชัน สามารถใช้ได้ทั้งบน in-process, local และ remote เซิร์ฟเวอร์

อโโตเมชัน มีลักษณะที่เด่นๆ 2จุดคือ

- อโโตเมชัน ออบเจกต์ต้องสามารถกำหนดเขตของคุณสมบัติและ คำสั่ง แล้วต้องสามารถเตรียมข้อมูลเกี่ยวกับอินเตอร์เฟสของออบเจกต์เช่น เมธอดของอินเตอร์เฟส และ อาร์กิวเมนต์ของออบเจกต์ธรรมดาแล้วจะสามารถหาข้อมูลได้ในโทปโกลบรัลรี ตัว อโโตเมชัน เซิร์ฟเวอร์ สามารถสร้างข้อมูลของรูปแบบได้โดยอัตโนมัติ
- อโโตเมชัน ออบเจกต์ต้องสร้างเมธอดที่สามารถให้แอปพลิเคชันอื่นๆ มาใช้ได้ ในกรณีนี้คือต้องมี IDispatch เราสามารถดูเมธอดทั้งหมดและ คุณสมบัติได้ผ่านทางอินเตอร์เฟสนี้ ผู้พัฒนาจะใช้ อโโตเมชัน ในการสร้างและใช้ OLE ออบเจกต์ที่รันได้ในทุกๆ โพรเซสเพราะอินเตอร์เฟส IDispatch สามารถใช้มาร์แชลลิง

### 3.4.2 ActiveX controls

ActiveX controls เป็นตัวคอนโทรลที่มองเห็นได้สามารถรันใน in-process เซิร์ฟเวอร์ ได้อย่างเดียว และสามารถต่อกับแอปพลิเคชันที่เป็นคอนเทนเนอร์ ของ ActiveX control ตัวมันเองนั้น ไม่ได้เป็นแอปพลิเคชัน แต่เป็นไลบรารีที่สามารถถูกเรียกใช้ได้ในแอปพลิเคชันต่างๆ และสามารถถูกเรียกใช้งานจากหลายๆ แอปพลิเคชัน พร้อม ๆ กันได้ งานที่ใช้ ActiveX control กันมากก็คืองานประเภทเว็บไซค์ ต่าง ๆ

### 3.4.3 Type libraries

เป็นทางที่จะรู้ถึง ข้อมูลเกี่ยวกับรูปแบบ (Type information) ของออบเจกต์ ข้อมูลเกี่ยวกับรูปแบบจะเก็บอยู่ใน Type libraries ในนั้นจะมีข้อมูลเกี่ยวกับออบเจกต์ และอินเตอร์เฟส เช่น มีอินเตอร์เฟสอะไรบ้างในออบเจกต์ มีฟังก์ชันอะไรบ้างในอินเตอร์เฟส และบอกถึงชนิดและจำนวนอาร์กิวเมนต์อะไรบ้างที่จะต้องส่งให้ฟังก์ชันรวมถึง หมายเลขที่ไม่ซ้ำสำหรับ Coclases (CLSIDs) และ IIDs, รูทีนที่ส่งออกมาโดย อโโตเมชัน หรือ ActiveX เซิร์ฟเวอร์ แต่ไม่ใช่เมธอดของอินเตอร์เฟส, ข้อมูลเกี่ยวกับ enumeration, record, union, alias

การสร้าง Type libraries สามารถสร้างโดยการใช้ทุล (Tool) โดยเขียนสคริปต์ใน Interface Definition Language (IDL) หรือ Object Description Language (ODL) แล้วรันสคริปต์ผ่านคอมไพเลอร์ แต่อาจใช้ Type Library editor มาช่วยในการสร้างก็ได้ทำให้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type libraries เป็นสิ่งที่สำคัญสำหรับแต่ละเซตของออบเจกต์ในการทำให้ผู้ใช้เข้าใจ ตัวอย่างความสำคัญที่ต้องใช้ Type libraries เช่น ActiveX controls ต้องการ Type libraries ที่ต้องรวมทรัพยากรใน DLL ที่เก็บ ActiveX controls, ออบเจกต์ที่สนับสนุน vtable ของอินเตอร์เฟสที่สร้างขึ้นต้องอธิบายใน Type libraries เพราะ vtable จะต้องอ้างอิงในเวลาคอมไพล์, แอปพลิเคชันที่ใช้ ออโตเมชัน เซิร์ฟเวอร์ ต้องมี Type libraries สำหรับไคลเอนต์ ฯลฯ

ข้อได้เปรียบในการใช้โทปโบลารีคือ ช่วยตรวจสอบรูปแบบขณะคอมไพล์, สามารถดูโบลารีได้เมื่อไคลเอนต์ต้องการรู้ลักษณะของออบเจกต์ ฯลฯ

#### 3.4.4 Active Server Page (ASP)

เป็นเทคโนโลยีที่ใช้ในการสร้างเว็บเพจโดยใช้คอมโพเนนต์ของ ActiveX เซิร์ฟเวอร์ ใน Active server page เราสามารถฝัง ActiveX control ลงในเว็บเพจได้แล้วเรียกได้ตลอดเวลาที่เซิร์ฟเวอร์โหลดเว็บเพจ ทางด้านเซิร์ฟเวอร์ Active server page เป็น ActiveX คอมโพเนนต์ที่สามารถพัฒนาได้ด้วยภาษาต่างๆ เช่น Delphi, C++, Java, VB ส่วนทางด้านไคลเอนต์ ASP จะถูกมองเป็นเอกสาร HTML และสามารถดูได้ด้วยเว็บเบราว์เซอร์ต่างๆ

#### 3.4.5 Active Document

แอคทีฟด็อกคิวเมนต์ อ้างอิง OLE ด็อกคิวเมนต์ เป็นเซตของบริการของ COMเกี่ยวกับการ linking และ embedding, drag และ drop และ visual editing ตัว แอคทีฟด็อกคิวเมนต์ นั้นสามารถรวมข้อมูลหรือออบเจกต์ที่มีรูปแบบที่ต่างกันเช่น เสียง, ตัวหนังสือ, รูปภาพ ออบเจกต์ของ แอคทีฟด็อกคิวเมนต์ สามารถเป็นคอนเทนเนอร์ของเอกสารหรือเซิร์ฟเวอร์ของเอกสาร เราสามารถใช้ TOleContainer ซึ่งเป็นพื้นฐานของคอนเทนเนอร์ของ แอคทีฟด็อกคิวเมนต์ ในการสร้างออบเจกต์ของ แอคทีฟด็อกคิวเมนต์ เซิร์ฟเวอร์ ตัว แอคทีฟด็อกคิวเมนต์ เองนั้นสนับสนุนการใช้มาร์แชร์ลิงสำหรับการข้ามโพรเซสของแอปพลิเคชัน แต่ไม่สามารถรันในรีโมท เซิร์ฟเวอร์ได้

#### 3.4.6 Visual cross-process objects

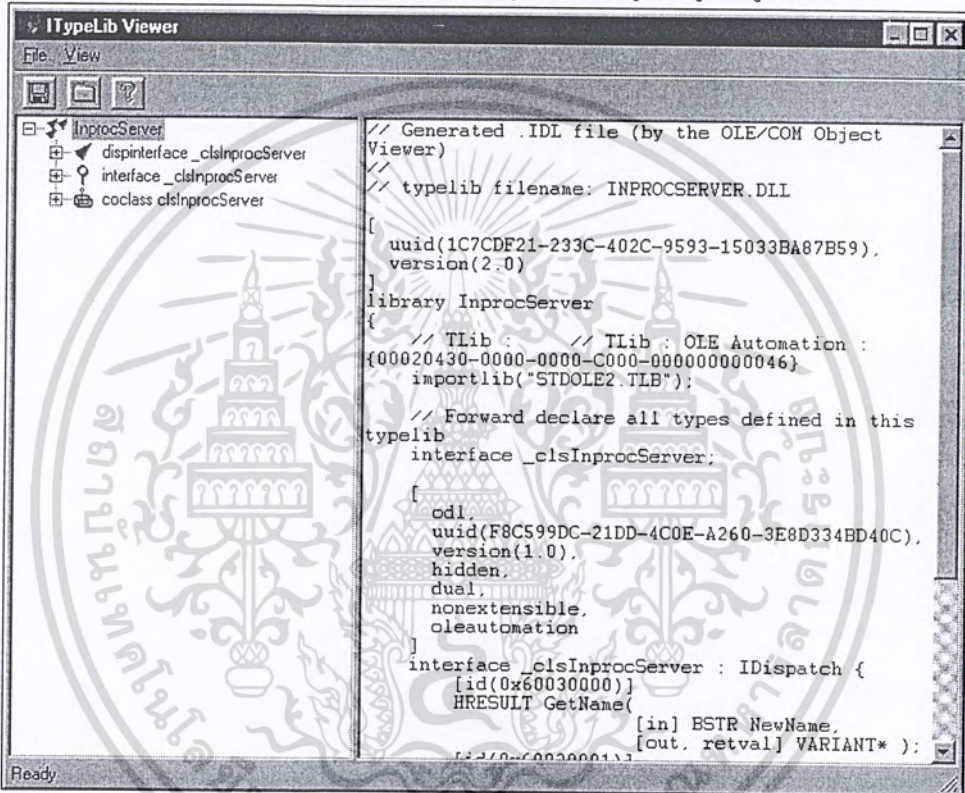
ออโตเมชันออบเจกต์, แอคทีฟด็อกคิวเมนต์, ActiveX control เป็นการใช้ออบเจกต์ต่างๆ ไปอย่างน้อย OLE และ ActiveX ออบเจกต์ ก็เป็นสิ่งที่มองเห็นและจัดการแบบข้ามโพรเซส รูปแบบของออบเจกต์เป็นการยากในการสร้างเพราะโพรโตคอลที่ใช้ในการจัดการออบเจกต์แบบข้ามโพรเซสจะใช้อินเตอร์เฟสของแอคทีฟด็อกคิวเมนต์ ดังนั้นการสร้างอินเตอร์เฟสและการทำมาร์แชร์ลิงต่างๆ ต้องทำเอง ที่นี้สามารถทำได้ด้วยการใช้ คู่อินเตอร์เฟส IDispatch จะเตรียมมาร์แชร์ลิงให้อัตโนมัตซึ่งเป็นวิธีที่แนะนำเพราะง่ายในการสร้าง อีกวิธีคือการเขียนมาร์แชร์ลิงคลาสโดยใช้ IMarshal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การพัฒนาคอมโพเนนต์โดยใช้วิซวลเบสิก

ในวิซวลเบสิก เราสามารถสร้างคอมโพเนนต์ด้วย ActiveX control, ActiveX Document, ActiveX EXE และ ActiveX DLL โดยวิซวลเบสิกได้ถูกออกแบบให้ซ่อนความซับซ้อนในการสร้างคอมโพเนนต์ โดยจะสร้างไทป์ไลบรารี (Type Library) ให้โดยอัตโนมัติเมื่อคอมไพล์คอมโพเนนต์ ดังนั้นเราจึงไม่สามารถกำหนด GUID หรือรายละเอียดของอินเตอร์เฟซด้วยตนเองได้ แต่สามารถดูรายละเอียดของไทป์ไลบรารีได้ โดยใช้โปรแกรม OLE View ซึ่งอยู่ในวิซวลสตูดิโอชุดรูปที่ 4-1



รูปที่ 4-1 การใช้โปรแกรม OLE View แสดงไทป์ไลบรารีของไฟล์ InprocServer.dll

#### 4.1 ชนิดของคอมโพเนนต์ในวิซวลเบสิก

วิซวลเบสิกสร้างคอมโพเนนต์ประเภทต่าง ๆ ดังนี้

1. อินโพรเซส ใช้ ActiveX DLL
2. เอท้อฟโพรเซส ใช้ ActiveX EXE
3. รีโมทเซิร์ฟเวอร์ ใช้ ActiveX DLL และ ActiveX EXE

ส่วน ActiveX Control และ ActiveX Document ใช้ในการสร้างคอมโพเนนต์ที่ใช้งานบนเว็บ

#### 4.2 อินเตอร์เฟซกับวิซวลเบสิก

ชื่ออินเตอร์เฟซของออบเจกต์ที่สร้างจากวิซวลเบสิก คือชื่อของคลาสโมดูลโดยมีเครื่องหมาย \_ นำ

หน้า เช่น หากในโปรเจกต์มีคลาสโมดูลชื่อ clsInprocServer ชื่ออินเตอร์เฟซก็คือ \_clsInprocServer

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเท่านั้น เมื่อผู้ใดเห็นเอกสารนี้แล้ว กรุณา  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นต้น ส่วนชื่อเมธอดภายในอินเตอร์เฟซก็คือฟังก์ชันหรือโพรซีเจอร์ต่าง ๆ ที่กำหนดไว้ภายในคลาสโมดูลที่ประกาศไว้เป็น Public

#### 4.3 การสร้าง COM ออบเจกต์ในวิซวลเบสิก

การสร้าง COM ออบเจกต์ในวิซวลเบสิกสามารถทำได้ 2 วิธีคือ

##### 1. ใช้คำสั่ง New ซึ่งมีรูปแบบ

```
Dim [ชื่อออบเจกต์] As New [ชื่อคลาสโมดูล]
```

หรือใช้คำสั่ง

```
Dim [ชื่อออบเจกต์] As [ชื่อคลาสโมดูล]
```

```
Set [ชื่อออบเจกต์] = New [ชื่อคลาสโมดูล]
```

ซึ่งวิธีนี้หลังนี้สามารถทำลายออบเจกต์ได้เมื่อไม่ต้องการใช้ด้วยคำสั่ง

```
Set [ชื่อออบเจกต์] = Nothing
```

##### 2. ใช้คำสั่ง CreateObject ซึ่งมีรูปแบบ

```
Dim [ชื่อออบเจกต์] As Object
```

```
Set [ชื่อออบเจกต์] = CreateObject("[ชื่อไฟล์ .dll หรือ .exe].[ชื่อคลาสโมดูล]")
```

#### 4.4 ตัวอย่างการพัฒนาคอมโพเนนต์ซอฟต์แวร์

##### ตัวอย่างที่ 4-1 พัฒนาคอมโพเนนต์ซอฟต์แวร์ประเภทอินโพรเซส (In-process)

จุดประสงค์

1. สามารถสร้างคอมโพเนนต์ประเภทอินโพรเซสได้
2. ทดสอบคุณสมบัติต่าง ๆ ของเทคโนโลยี COM อันได้แก่
  - การแก้ปัญหาด้านเวอร์ชันของซอฟต์แวร์
  - การไม่ขึ้นกับภาษาที่ใช้พัฒนา

เครื่องมือที่ใช้พัฒนา

1. วิซวลเบสิก 6.0 เอนเตอร์ไพรส์เอดิชัน
2. บอร์แลนด์เดลไฟ 5.0 เอนเตอร์ไพรส์

คำอธิบายโปรแกรม

1. ใช้วิซวลเบสิกสร้างคอมโพเนนต์ประเภทอินโพรเซส ซึ่งมีเมธอดทำหน้าที่รับข้อความและส่งคำทักทายพร้อมข้อความที่ได้รับ
2. สร้างไคลเอนท์แอปพลิเคชันด้วยวิซวลเบสิกที่มีการเรียกใช้คอมโพเนนต์ดังกล่าวโดยให้ผู้ใช้พิมพ์ชื่อลงในเท็กซ์บ็อกซ์ (Textbox) แล้วคลิกปุ่มเพื่อแสดงข้อความทักทาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

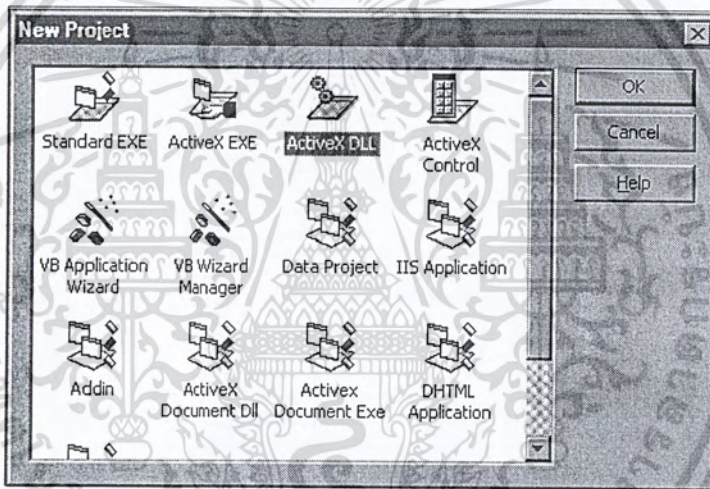
3. สร้างไคลเอนท์แอปพลิเคชันด้วยเคลฟไฟให้มีการทำงานเช่นเดียวกับไคลเอนท์แอปพลิเคชันที่เขียนด้วยวิซวลเบสิก
4. เปลี่ยนเวอร์ชันของคอมโพเนนต์ และใช้แอปพลิเคชันเดิมเรียกใช้คอมโพเนนต์เวอร์ชันใหม่ เพื่อแสดงให้เห็นว่าการเปลี่ยนเวอร์ชันไม่ทำให้แอปพลิเคชันเกิดปัญหาเรื่องการไม่เข้ากันระหว่างคอมโพเนนต์กับแอปพลิเคชัน

#### ขั้นตอนการพัฒนา

1. ออกแบบอินเตอร์เฟซที่จำเป็น ในที่นี่มีเพียงอินเตอร์เฟซเดียวคือ InprocServer ซึ่ง InprocServer ประกอบด้วยเมธอด 2 เมธอดคือ เมธอด GetName มีหน้าที่รับค่าสตริงมาเก็บไว้ในตัวแปร และเมธอด SayHello มีหน้าที่ส่งสตริง "Hello," ตามด้วยค่าของตัวแปรนั้น

2. ทำการสร้างคอมโพเนนต์โดยใช้วิซวลเบสิกดังนี้

2.1 เปิดโปรแกรมวิซวลเบสิก 6.0 แล้วเลือก *File > New Project* และเลือกประเภทโปรเจกต์ ActiveX DLL ดังรูปที่ 4-2



รูปที่ 4-2 หน้าจอเริ่มต้นในการสร้างคอมโพเนนต์ประเภทอินโพรเซสโดยใช้วิซวลเบสิก

2.2 ตั้งชื่อคลาส โมดูลชื่อ InprocServer เขียนโค้ดฟังก์ชัน GetName และฟังก์ชัน SayHello

ดังรูปที่ 4-3

```

Option Explicit
Private strName As String

Public Function GetName(ByVal NewName As String)
    strName = NewName
End Function

Public Function SayHello() As String
    SayHello = "Hello " & strName
End Function

```

รูปที่ 4-3 โค้ดของคลาสโมดูล InprocServer

2.3 ตั้งชื่อโปรเจกต์เป็น InprocServer แล้วทำการคอมไพล์โดยเลือก *File > Make InprocServer.dll* จะได้ไฟล์ InprocServer.dll ซึ่งเป็นคอมโพเนนต์ที่สามารถเรียกใช้ได้

2.4 บันทึกโปรเจกต์และปิดโปรแกรม

3. สร้างโปรแกรมทั่วไปที่จะเรียกใช้คอมโพเนนต์โดยใช้ Visual Basic มีขั้นตอนต่อไปนี้

3.1 เปิดโปรแกรม Visual Basic 6.0 แล้วเลือก *File > New Project* และเลือกประเภทโปรเจกต์ Standard EXE

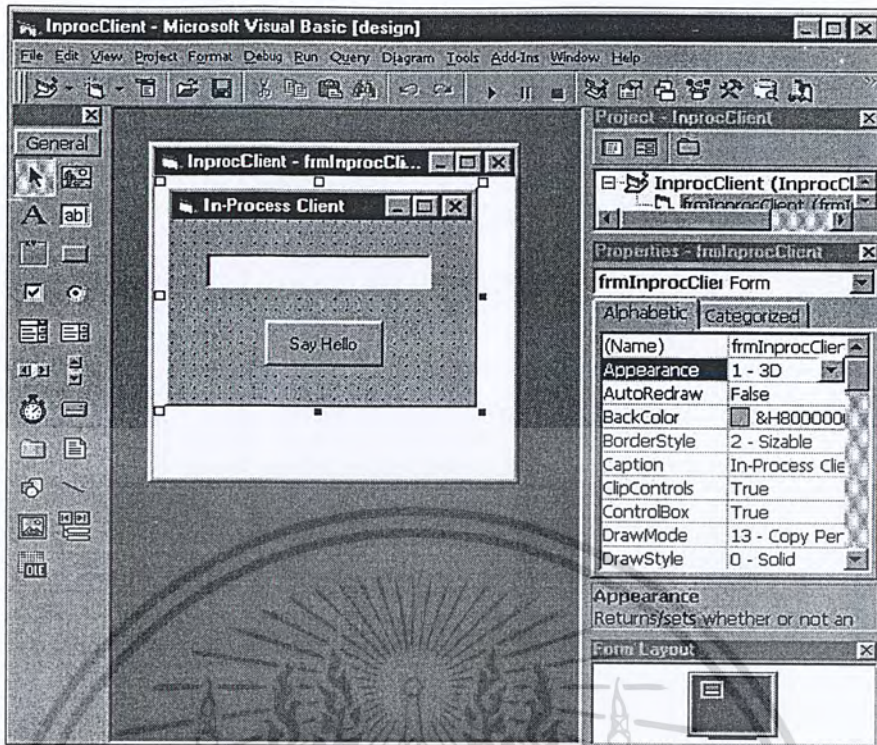
3.2 ออกแบบหน้าจอส่วนติดต่อกับผู้ใช้ดังรูปที่ 4-4 โดยรายละเอียดของคอนโทรลดังตาราง

ที่ 4-1

คอนโทรล	คุณสมบัติ	ค่า
Text1	Name	TxtName
	Text	<Blank>
Label1	Name	LblName
	Caption	Name:
Command1	Name	CmdSayHello
	Caption	Say Hello

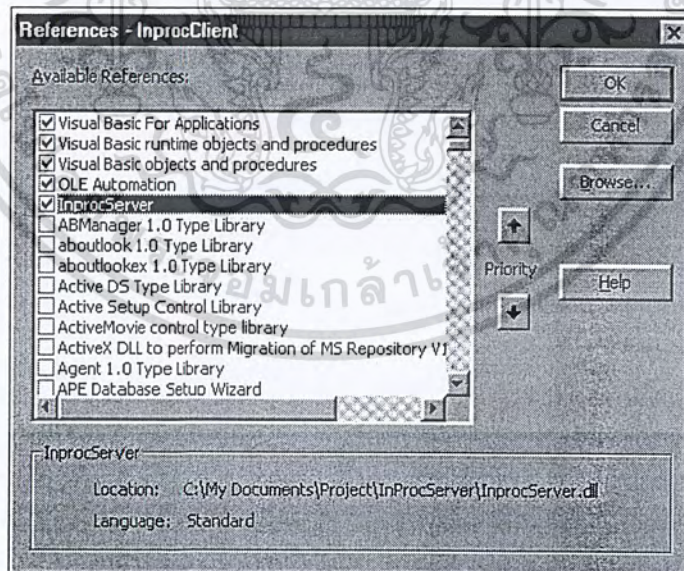
ตารางที่ 4-1 รายละเอียดของคอลโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-4 หน้าจอติดต่อกับผู้ใช้

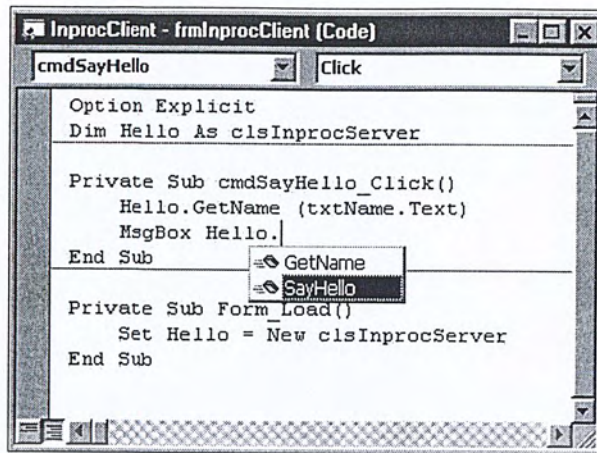
3.3 เรียกใช้คอมโพเนนต์ InprocServer1.dll โดยเลือก *Project > References...* จากนั้นเลือกไฟล์ InprocServer1.dll จะได้ผลดังรูปที่ 4-5



รูปที่ 4-5 การเรียกใช้คอมโพเนนต์ InprocServer.dll

3.4 เขียนโค้ดเพื่อเรียกใช้บริการต่าง ๆ ของคอมโพเนนต์ดังรูปที่ 4-6

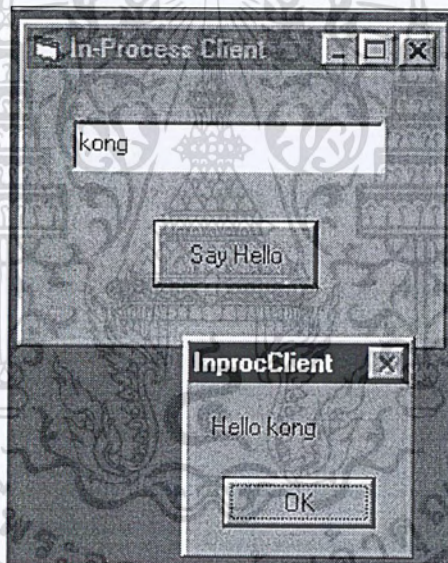
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 โค้ดของโปรแกรม

3.5 ทดลองรันโปรแกรม โดยเลือก *run > start* หรือกดปุ่ม F5

3.6 ใส่ชื่อและคลิกปุ่ม Say Hello จะได้ผลลัพธ์ดังรูปที่ 4-7



รูปที่ 4-7 ผลการทำงานของโปรแกรม

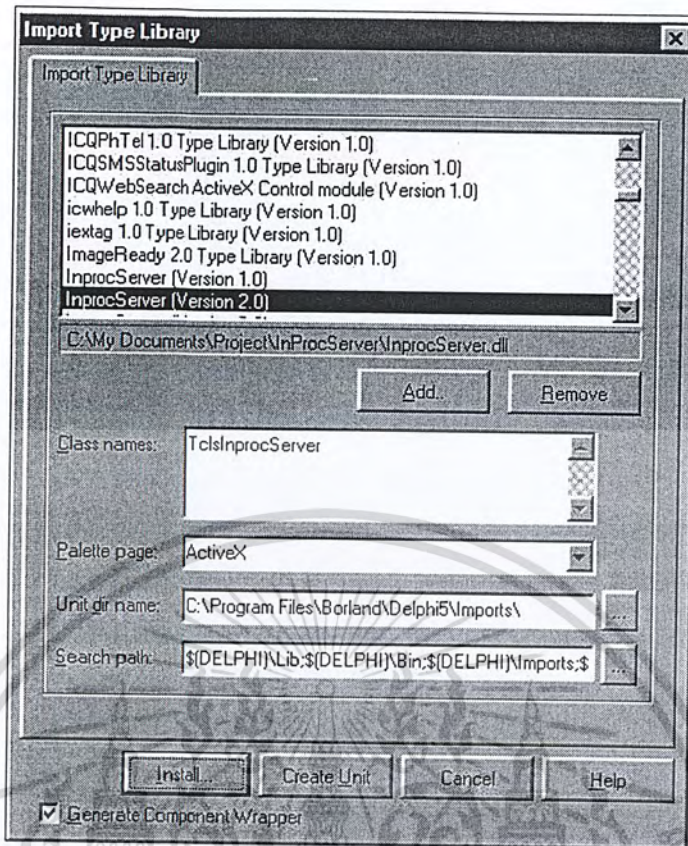
4. ทดสอบคุณสมบัติการไม่ขึ้นกับภาษาที่ใช้พัฒนา โดยใช้เคลฟสร้างไคลเอนท์แอปพลิเคชัน มีรายละเอียดดังนี้

4.1 เปิดโปรแกรมเดลไฟ คลิก *File > New Application*

4.2 ก่อนเรียกใช้คอมโพเนนต์ต้องทำการอิมพอร์ตไทป์ไลบรารี โดยคลิก *Project > Import Type Library* จะปรากฏหน้าต่าง Import Type Library คลิกปุ่ม Add เพื่อเลือกไฟล์ .dll ที่ต้องการ ในที่นี้คือ ไฟล์ InprocServer.dll จากนั้นคลิกปุ่ม Install ดังรูปที่ 4-8

4.3 เรียกใช้คอมโพเนนต์โดยเพิ่มชื่อของไฟล์ .dll ตามด้วย \_TLB ที่ส่วน uses เช่นในไคลเอนท์แอปพลิเคชันนี้ จะเพิ่มข้อความ InprocServer\_TLB ลงไปดังรูปที่ 4-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 หน้าต่าง Import Type Library

**uses**

Windows, Messages, SysUtils, Classes, Graphics, Controls,  
InprocServer\_TLB, StdCtrls;

รูปที่ 4-9 การเรียกใช้คอมโพเนนต์ InprocServer.dll

## 4.4 การสร้างออบเจกต์ทำได้ดังนี้

- ประกาศชื่อตัวแปรออบเจกต์โดยมีรูปแบบ

[ชื่อตัวแปร] : [ชื่ออินเตอร์เฟส]; เช่น Hello : \_clsInprocServer;

- การสร้างออบเจกต์โดยมีรูปแบบ

[ชื่อตัวแปร] := Co[ชื่อคลาส โมดูล].create; เช่น Hello :=

CoclsInprocServer.create

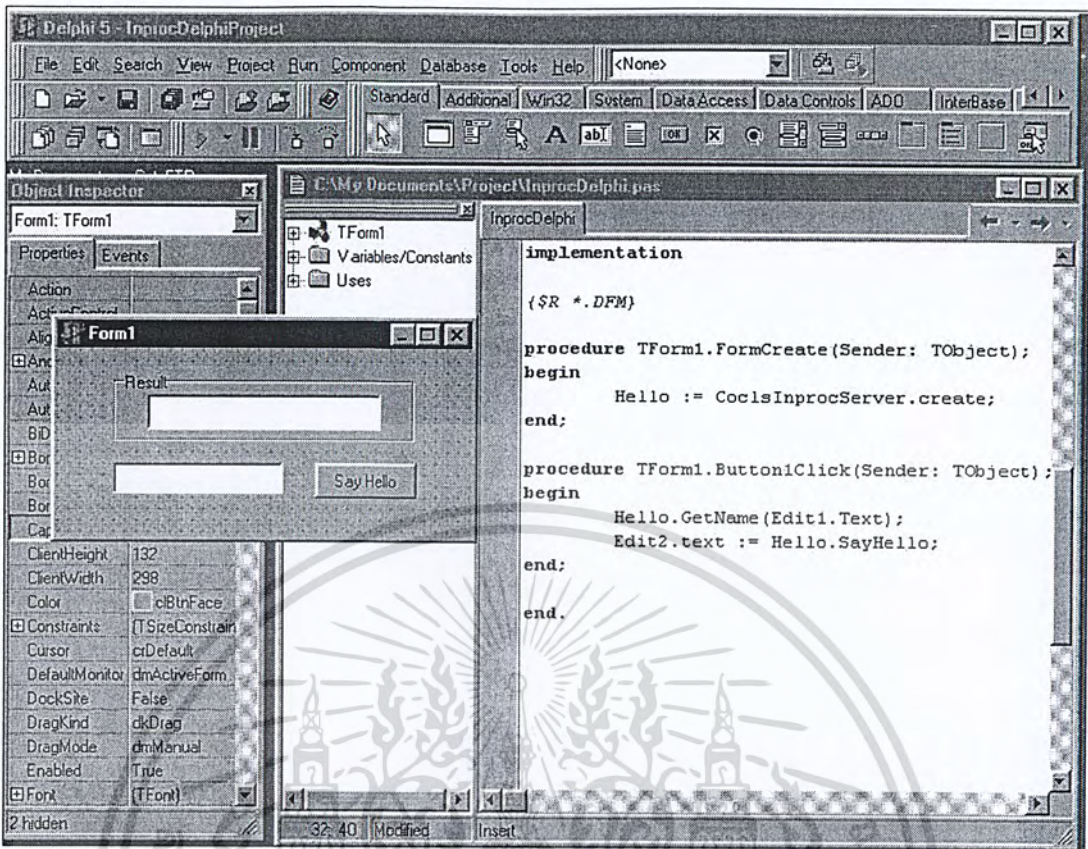
เมื่อสร้างออบเจกต์แล้วก็สามารถเรียกใช้เมธอดต่าง ๆ ได้ตามที่คอมโพเนนต์ประกาศ

## 4.5 ออกแบบหน้าจอส่วนติดต่อกับผู้ใช้ และเขียนโค้ดจัดการดังรูปที่ 4-10

## 4.6 ทดสอบโปรแกรม โดยคลิก Run &gt; Run หรือกด F9 ทดลองป้อนข้อความแล้วคลิก Say

Hello จะได้ผลการทำงานเหมือนโคลนที่แอปพลิเคชันที่สร้างจากวิซวลเบสิกดังรูปที่ 4-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-10 หน้าจอส่วนติดต่อกับผู้ใช้และโค้ด



รูปที่ 4-11 ผลการทำงานของไคลเอนท์แอปพลิเคชันที่สร้างจากเดลไฟ

#### ตัวอย่างที่ 4-2 การพัฒนาคอมโพเนนต์เซอร์ฟแวร์ประเภทเอาท์ออฟโพรเซส (Out-of-process)

##### จุดประสงค์

1. สามารถสร้างคอมโพเนนต์ประเภทเอาท์ออฟโพรเซสได้
2. ทดสอบคุณสมบัติการไม่ขึ้นกับภาษาที่ใช้พัฒนา

##### เครื่องมือที่ใช้ในการพัฒนา

1. วิซวลเบสิก 6.0 เอนเตอร์ไพรส์เอดิชัน
2. บอร์แลนด์เดลไฟ 5 เอนเตอร์ไพรส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### คำอธิบายโปรแกรม

1. ใช้วิชวลเบสิกสร้างคอมโพเนนต์ประเภทเอาท์โพรเซส เป็นเซิร์ฟเวอร์ทำหน้าที่รับข้อความและส่งข้อความที่ได้รับออกทางหน้าต่างของโปรแกรม
2. สร้างแอปพลิเคชันทำหน้าที่เป็นไคลเอนท์โดยใช้วิชวลเบสิก ทำการเรียกใช้เมธอดต่าง ๆ ของเซิร์ฟเวอร์ โดยรับสตรีมจากเซิร์ฟเวอร์ และสามารถแสดงสตรีมใหม่ออกทางหน้าต่างของโปรแกรมทั้งฝั่งไคลเอนท์และเซิร์ฟเวอร์ได้
3. สร้างแอปพลิเคชันทำหน้าที่เป็นไคลเอนท์โดยใช้เซลล์ไฟมีการทำงานเหมือนกับไคลเอนท์ในข้อที่ 2

### ขั้นตอนการพัฒนา

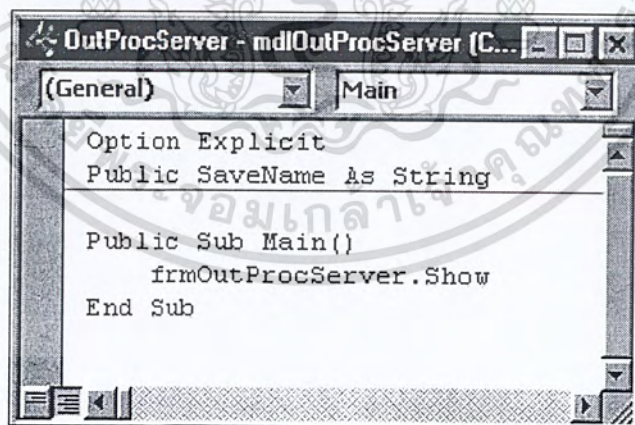
1. ออกแบบอินเตอร์เฟซที่จำเป็น ในที่นี้มีเพียงอินเตอร์เฟซเดียวคือ clsOutProcServer ซึ่งประกอบด้วย 2 เมธอดคือ เมธอด GetText ทำการรับสติงมาเก็บไว้ในตัวแปร และเมธอด SendText ทำการส่งค่าตัวแปรนั้น

2. ทำการสร้างคอมโพเนนต์โดยใช้วิชวลเบสิกดังนี้

2.1 เปิดโปรแกรมวิชวลเบสิก 6.0 เลือก *File > New Project* และเลือกประเภทโปรเจกต์ ActiveX EXE และตั้งชื่อโปรเจกต์เป็น prjOutProcServer.vbp

2.2 สร้างโมดูล mdlOutProcServer โดยคลิก *Project > Add Module* การสร้างโมดูลเพื่อใช้ในการประกาศตัวแปรที่มีลักษณะเป็นสเตติกคลาสดาต้า (Static Class Data) คือเป็นตัวแปรที่ทุกอินสแตนซ์ของคลาสดาต้าสามารถแชร์ร่วมกันได้ ซึ่งในที่นี้คือตัวแปร SaveName

2.3 เขียนโค้ดภายใน โมดูลดังรูปที่ 4-12



รูปที่ 4-12 โค้ดในโมดูล mdlOutProcServer

2.4 สร้างคลาสนิโมดูล clsOutProcServer โดยคลิก *Project > Add Class Module*

2.5 เขียนโค้ดภายในคลาสนิโมดูลดังรูปที่ 4-13

2.6 สร้างฟอร์ม โดยคลิก *Project > Add Form* เพื่อเป็นส่วนโต้ตอบกับผู้ใช้ เนื่องจากคอมโพเนนต์แบบเอาท์โพรเซสสามารถทำงานได้ด้วยตนเอง โดยมีการอ้างฟังก์ชันต่าง ๆ ในคลาสนิโมดูล clsOutProcServer ดังรูปที่ 4-14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Option Explicit
Procedure

Public Function GetText(ByVal NewName As String)
    SaveName = NewName
End Function

Public Function SendText()
    SendText = SaveName
End Function

```

รูปที่ 4-13 โค้ดในคลาสโมดูล clsOutProcServer

```

Option Explicit
Dim test As clsOutProcServer

Private Sub Form_Load()
    Set test = New clsOutProcServer
End Sub

Private Sub Timer1_Timer()
    txtResult = test.
End Sub

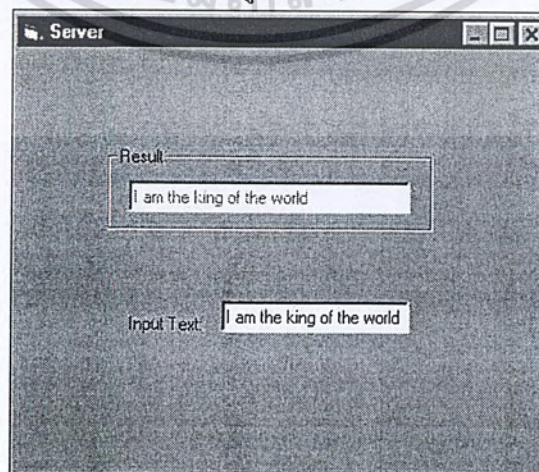
Private Sub txtText_Change()
    test.GetText (txtText)
    txtResult = test.SendText
End Sub

```

รูปที่ 4-14 โค้ดของฟอร์ม frmOutProcServer

2.7 ทดสอบคอมไพเลอร์โดยคลิก *Run > Start* หรือกด *F5*

2.8 ทดสอบโดยพิมพ์ข้อความลงในเท็กซ์บ็อกซ์ เซิร์ฟเวอร์แอปพลิเคชันจะทำงานโดยรับข้อความนั้นไปแสดงที่เท็กซ์บ็อกซ์อีกอันหนึ่ง ดังรูปที่ 4-15



รูปที่ 4-15 ผลการทดสอบแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 บันทึกโปรเจกต์และคอมไพล์โปรเจกต์โดยคลิก *File > Make prjOutProcSercer.exe* จะได้ไฟล์ prjOutProcServer ซึ่งเป็นคอมโพเนนต์แบบเอาต์ออฟโพรเซสที่ต้องการ

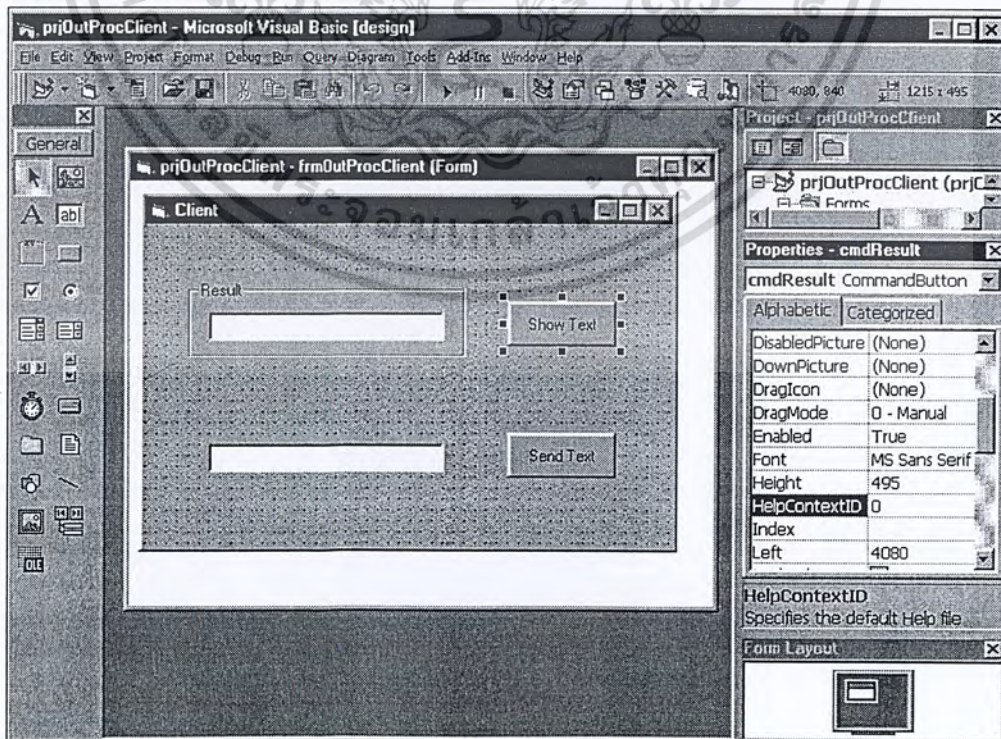
### 3. สร้างไคลเอนท์แอปพลิเคชันโดยใช้วิซวลเบสิก มีรายละเอียดดังต่อไปนี้

3.1 เปิดโปรแกรมวิซวลเบสิก 6.0 แล้วเลือก *File > New Project* และเลือกประเภทโปรเจกต์ Standard EXE และตั้งชื่อโปรเจกต์เป็น prjOutProcClient.vbp

3.2 ออกแบบหน้าจอส่วนติดต่อกับผู้ใช้ดังรูปที่ 4-16 โดยรายละเอียดของคอนโทรลดังตารางที่ 4-2

คอนโทรล	คุณสมบัติ	ค่า
Text1	Name	txtText
	Text	<Blank>
Text2	Name	TxtResult
	Text	<Blank>
Frame1	Name	FraResult
	Caption	Result
Command1	Name	CmdResult
	Caption	Show Text
Command2	Name	cmdSendText
	Caption	Send Text

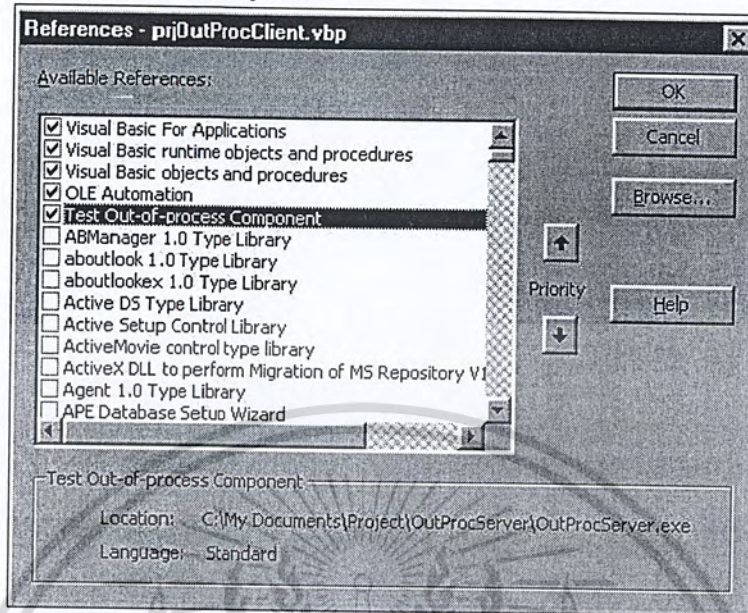
ตารางที่ 4-2 รายละเอียดของคอนโทรล



รูปที่ 4-16 หน้าจอส่วนติดต่อกับผู้ใช้

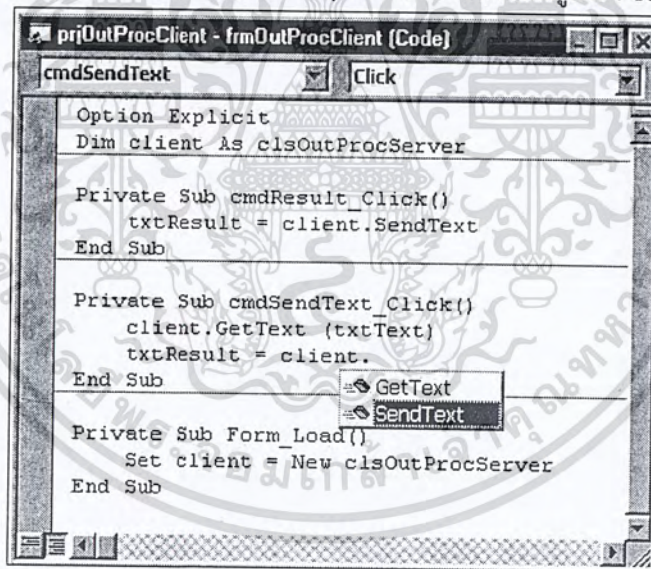
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิใช่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 เรียกใช้คอมโพเนนต์ prjOutProcServer.exe โดยเลือก *Project > References...* เลือกไฟล์ prjOutProcServer.exe จะ ได้ผลดังรูปที่ 4-17



รูปที่ 4-17 การเรียกใช้คอมโพเนนต์ prjOutProcServer.exe

3.4 เขียนโค้ดเพื่อเรียกใช้บริการต่าง ๆ ของคอมโพเนนต์ดังรูปที่ 4-18



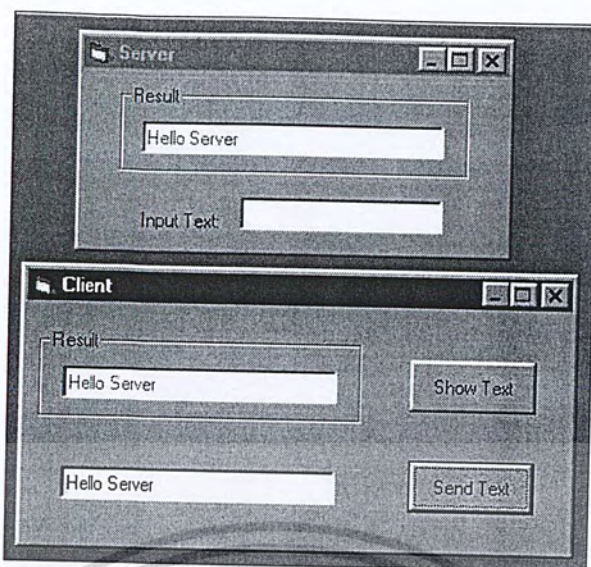
รูปที่ 4-18 โค้ดภายในฟอร์มที่ใช้ติดต่อกับผู้ใช้

3.5 ทดสอบไคลเอนท์แอปพลิเคชัน โดยคลิก *Run > Start* หรือกด *F5*

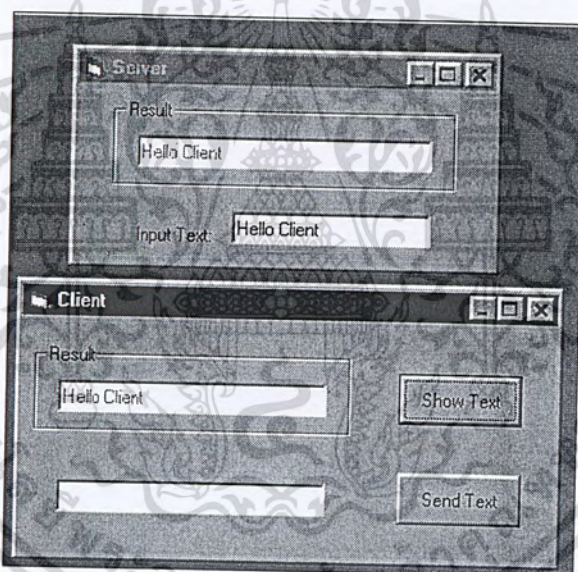
3.6 เมื่อรันไคลเอนท์แอปพลิเคชัน ส่วนของเซิร์ฟเวอร์แอปพลิเคชันจะปรากฏขึ้นมาทันที จากนั้นลองพิมพ์ข้อความลงในเท็กซ์บ็อกซ์ของไคลเอนท์ แล้วคลิก *Sent Text* ข้อความที่พิมพ์จะปรากฏที่เท็กซ์บ็อกซ์ทั้งไคลเอนท์และเซิร์ฟเวอร์ ดังรูปที่ 4-19

3.7 ทดลองพิมพ์ข้อความในเท็กซ์บ็อกซ์ของเซิร์ฟเวอร์ จากนั้นกลับมาคลิกปุ่ม *Show Text* ของไคลเอนท์ ข้อความในเท็กซ์บ็อกซ์ของเซิร์ฟเวอร์จะปรากฏที่เท็กซ์บ็อกซ์ของไคลเอนท์ดังรูปที่ 4-20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



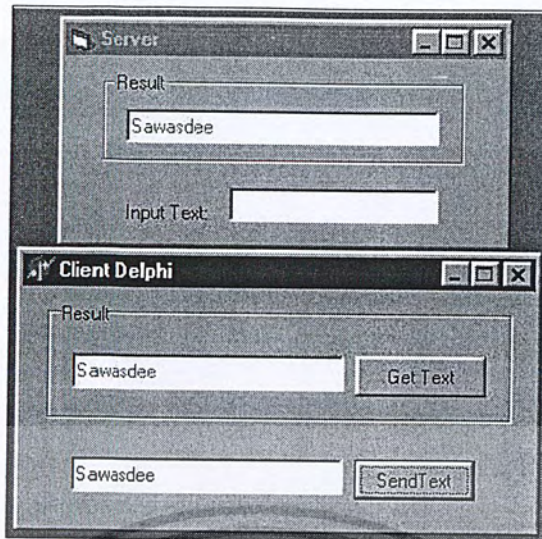
รูปที่ 4-19 ผลการทดลองเมื่อพิมพ์ข้อความในแท็บบ็อกซ์ของไคลเอนต์ แล้วคลิก *Send Text* ข้อความจะไปปรากฏที่แท็บบ็อกซ์ของเซิร์ฟเวอร์



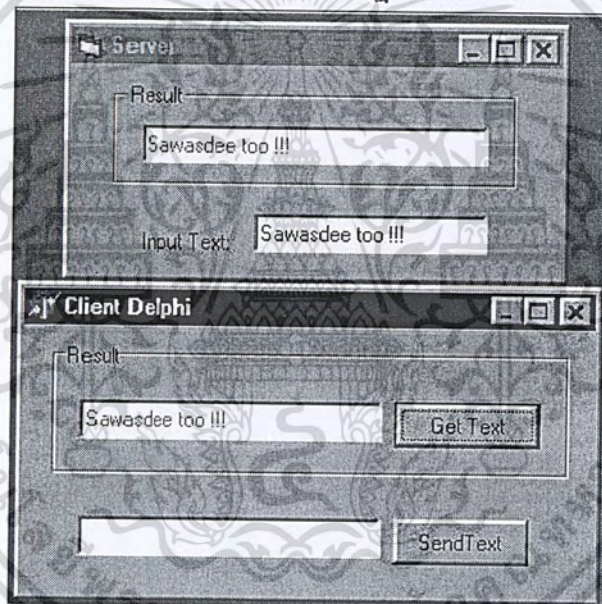
รูปที่ 4-20 ผลการทดลองเมื่อพิมพ์ข้อความลงในแท็บบ็อกซ์ของเซิร์ฟเวอร์ เมื่อไคลเอนต์คลิก *Show Text* ข้อความจะไปปรากฏที่แท็บบ็อกซ์ในเฟรม *Result*

4. ทดสอบคุณสมบัติการไม่ขึ้นกับภาษาที่ใช้พัฒนา โดยใช้เคลไพล์สร้างไคลเอนต์แอปพลิเคชัน การสร้างมีขั้นตอนเช่นเดียวกับการสร้างไคลเอนต์แอปพลิเคชันด้วยเคลไพล์ในการทดลองที่ 1 ทดลองรันไคลเอนต์จะได้ผลการทดลองดังรูปที่ 4-21 และรูปที่ 4-22 ซึ่งผลการทดลองเหมือนกับการทำงานของไคลเอนต์แอปพลิเคชันที่เขียนด้วยวิซวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-21 ผลการทดลองเมื่อพิมพ์ข้อความในเท็กซ์บ็อกซ์ของเดลไฟไคลเอนท์ แล้วคลิก *Send Text* ข้อความจะไปปรากฏที่เท็กซ์บ็อกซ์ของเซิร์ฟเวอร์



รูปที่ 4-22 ผลการทดลองเมื่อพิมพ์ข้อความลงในเท็กซ์บ็อกซ์ของเซิร์ฟเวอร์ เมื่อเดลไฟไคลเอนท์คลิก *Show Text* ข้อความจะไปปรากฏที่เท็กซ์บ็อกซ์ในเฟรม *Result*

### ตัวอย่างที่ 3 การพัฒนาคอมโพเนนต์ประเภทรีโมทเซิร์ฟเวอร์ (Remote Server)

#### จุดประสงค์

1. สามารถสร้างรีโมทคอมโพเนนต์ที่สามารถเรียกใช้ระหว่างเครื่องได้ เครื่องมือที่ใช้พัฒนา
  1. วิชาการเบสิก 6.0 เอนเตอร์ไพรส์เอดิชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### คำอธิบายโปรแกรม

1. ใช้วิซวลเบสิกสร้างรีโมทคอมโพเนนต์ เป็นเซิร์ฟเวอร์ซึ่งมีเมธอดทำหน้าที่รับข้อความและแสดงข้อความ “Welcome” ตามด้วยข้อความนั้นและเครื่องหมาย “!!!” โดยทำงานอยู่บนคอมพิวเตอร์ระบบปฏิบัติการวินโดวส์เอ็นที 4.0 เซิร์ฟเวอร์

2. สร้างไคลเอนท์แอปพลิเคชันซึ่งทำงานอยู่บนเครื่องกับคอมโพเนนต์ที่สร้าง ทำการติดต่อกับคอมโพเนนต์โดยใช้ DCOM และเรียกใช้เมธอดต่าง ๆ ของคอมโพเนนต์

### ขั้นตอนการพัฒนาโปรแกรม

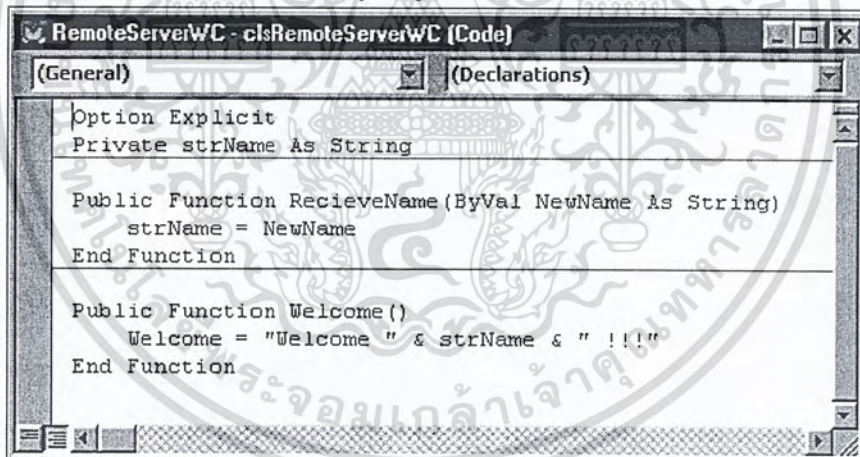
1. ออกแบบอินเตอร์เฟซ ซึ่งมีอินเตอร์เฟซเดียวคือ clsRemoteServerWC มีเมธอด 2 เมธอดคือ RecieveName() ทำหน้าที่รับค่าพารามิเตอร์มาเก็บไว้ในตัวแปรภายในคอมโพเนนต์ และเมธอด Welcome ทำหน้าที่ส่งข้อความ “Welcome” ตามด้วยค่าตัวแปรที่ได้จากเมธอด RecieveName() และต่อท้ายด้วยข้อความ “!!!”

2. ทำการสร้างคอมโพเนนต์บนระบบปฏิบัติการวินโดวส์เอ็นที 4.0 เซิร์ฟเวอร์ดังนี้

2.1 เปิดโปรแกรมวิซวลเบสิก 6.0 เลือก *File > New Project* และเลือกประเภทโปรเจกต์ ActiveX EXE ตั้งชื่อโปรเจกต์เป็น RemoteServerWC

2.2 สร้างคลาสโมดูล clsRemoteServerWC โดยคลิก *Project > Add Class Module*

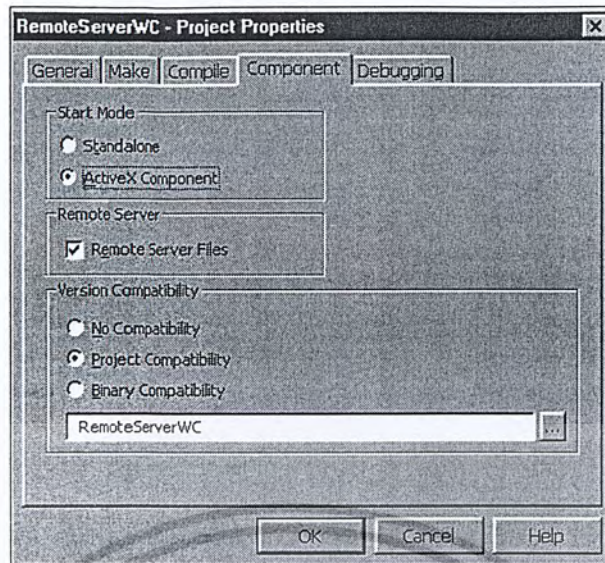
2.3 เขียนโค้ดภายในคลาสโมดูลดังรูปที่ 4-23



รูปที่ 4-23 โค้ดภายในคลาสโมดูล clsRemoteServerWC

2.4 คลิก *Project > RemoteServerWC Properties...* จะมีหน้าต่าง Project Properties ปรากฏขึ้นมา เลือกแท็บ Component และทำเครื่องหมายที่เช็กระบ็อกซ์ Remote Server Files ดังรูปที่ 4-24 จุดประสงค์เพื่อเมื่อทำการคอมไพล์โปรเจกต์ วิซวลเบสิกจะทำการสร้างไฟล์ .vbr และ .tlb ซึ่งมีชื่อเดียวกับโปรเจกต์ ไฟล์ .vbr มีข้อมูลที่วินโดวส์จิสตรีต้องการในการรันคอมโพเนนต์จากเครื่องอื่น

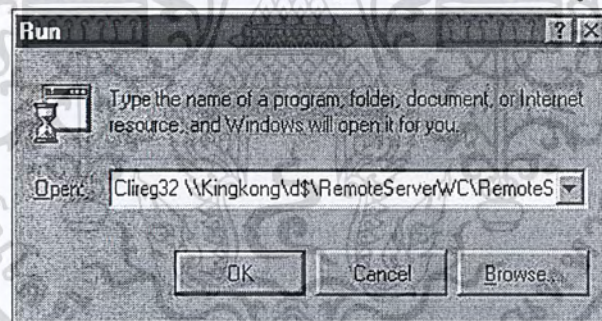
2.5 คลิก *File > Make RemoteServerWC.exe* เพื่อทำการสร้างไฟล์ RemoteServerWC.exe พร้อมกันนั้นจะได้ไฟล์ RemoteServerWC.tlb และ RemoteServerWC.vbr



รูปที่ 4-24 เช็บบ็อกซ์ Remote Server Files

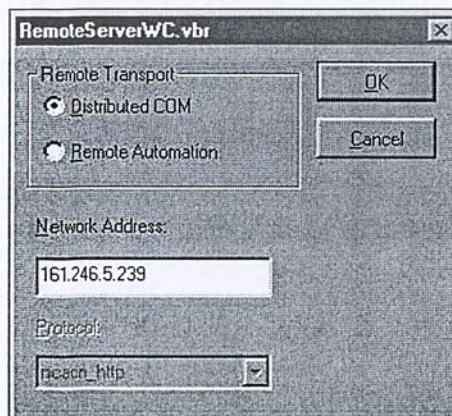
3. สร้างไคลเอนต์แอปพลิเคชันบนเครื่องอื่น โดยในการทดลองนี้ใช้เครื่องระบบปฏิบัติการวินโดว์ 98

3.1 รีจิสเตอร์รีโมทคอมโพเนนต์บนเครื่องไคลเอนต์ โดยใช้โปรแกรม Clireg32.exe บนเดสก์ท็อปของวินโดว์ 98 คลิกปุ่ม *Start > Run...* และพิมพ์คำสั่ง Clireg32 ตามด้วย path ของไฟล์ .vbr ที่อยู่บนเครื่องเซิร์ฟเวอร์ เช่น clireg32 \\[ชื่อเซิร์ฟเวอร์]\[path ของไฟล์ .vbr] ดังรูปที่ 4-25



รูปที่ 4-25 การรีจิสเตอร์รีโมทคอมโพเนนต์บนเครื่องไคลเอนต์ที่ใช้โปรแกรม Clireg32.exe

3.2 เมื่อคลิกปุ่ม OK จะปรากฏหน้าต่างดังรูปที่ 4-26 เลือก Distributed COM และในแท็บชื่้ออกซ์ Network Address พิมพ์ IP แอดเดรสของเครื่องเซิร์ฟเวอร์ จากนั้นคลิกปุ่ม OK



รูปที่ 4-26 โปรแกรม Clireg32.exe

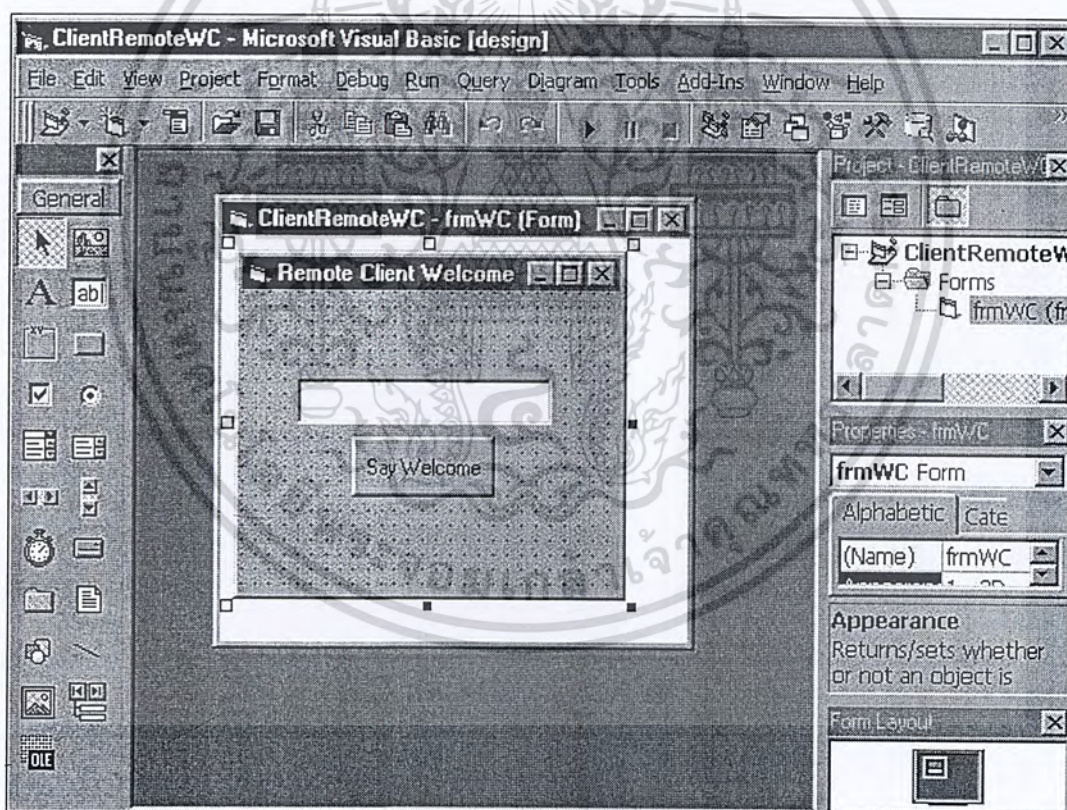
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในองค์กรที่ก๊อปปี้ให้ผู้อื่น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 สร้างไคลเอนท์แอปพลิเคชัน โดยเปิดโปรแกรมวิซวลเบสิก 6.0 คลิก *File > New Project* และเลือกประเภทโปรเจกต์ Standard EXE ตั้งชื่อโปรเจกต์เป็น ClientRemoteWC

3.4 ออกแบบหน้าจอส่วนติดต่อกับผู้ใช้งานดังรูปที่ 4-27 โดยรายละเอียดของคอนโทรลแสดงในตารางที่ 4-3

คอนโทรล	คุณสมบัติ	ค่า
Text1	Name	TxtName
	Text	<Blank>
Command1	Name	CmdSayWelcome
	Caption	Say Welcome
Form1	Name	FrmWC
	Caption	Remote Client Welcome

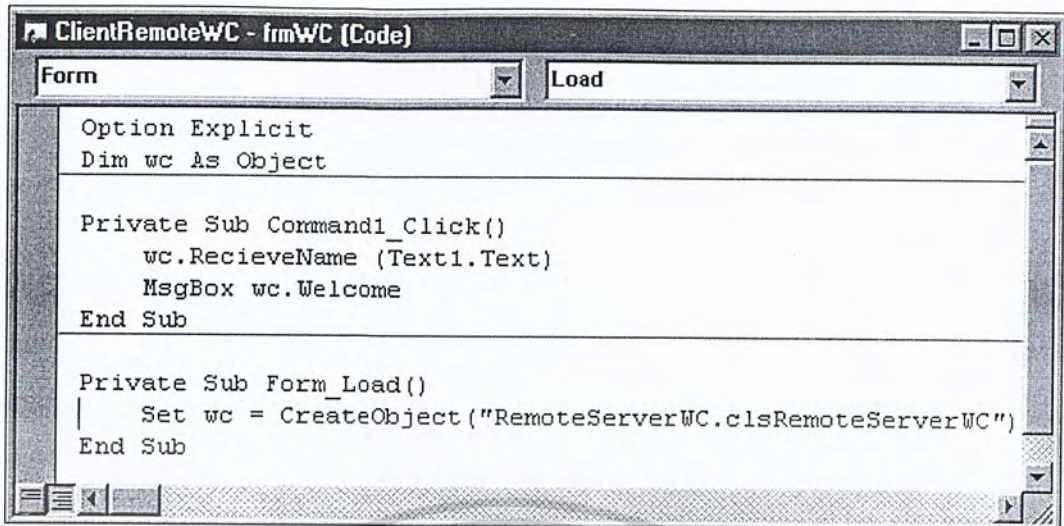
ตารางที่ 4-3 รายละเอียดของคอนโทรล



รูปที่ 4-27 หน้าจอส่วนติดต่อกับผู้ใช้งาน

3.5 เขียนโค้ดดังรูปที่ 4-28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

ClientRemoteWC - frmWC (Code)
Form
Load

Option Explicit
Dim wc As Object

Private Sub Command1_Click()
    wc.RecieveName (Text1.Text)
    MsgBox wc.Welcome
End Sub

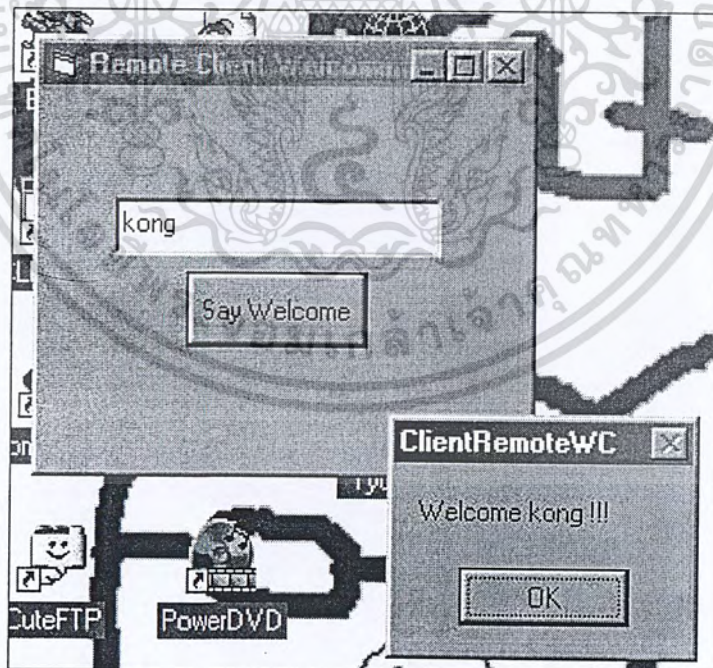
Private Sub Form_Load()
    Set wc = CreateObject("RemoteServerWC.clsRemoteServerWC")
End Sub

```

รูปที่ 4-28 โค้ดภายในฟอร์มที่ใช้ติดต่อกับผู้ใช้

จากรูปที่ 4-28 คำสั่งที่ใช้สร้างออบเจกต์คือ CreateObject() โดยพารามิเตอร์ภายในเป็นชื่อของคอมโพเนนต์ตามด้วยคลาสโมดูลที่ต้องการใช้งาน เมื่อสร้างออบเจกต์แล้ว จะสามารถเรียกใช้เมทอดต่าง ๆ ได้ ซึ่งในโปรแกรมนี้มี 2 เมทอดคือ RecieveName และ Welcome

3.6 ทดสอบการทำงานของโปรแกรมโดยคลิก *Run > Start* หรือพิมพ์ F5 ทดลองพิมพ์ข้อความลงในเท็กซ์บ็อกซ์แล้วคลิก Say Welcome จะได้ผลการทำงานดังรูปที่ 4-29



รูปที่ 4-29 ผลการทำงานของไคลเอนท์แอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### การพัฒนา COM โดยใช้ Borland Delphi 5.0

#### 5.1 กรณีศึกษาการพัฒนา คอมโพเนนต์ ออบเจกต์ ประเภท อิน – โพรเซส (In-Process COM Object) จุดประสงค์

1. เข้าใจถึงวิธีการในการพัฒนาคอมโพเนนต์ซอฟต์แวร์ประเภท อิน – โพรเซส
2. เข้าใจถึงวิธีการเรียกใช้คอมโพเนนต์ซอฟต์แวร์ประเภท อิน – โพรเซส จาก เครื่องมือพัฒนาตัวเดียวกัน
3. เข้าใจถึงวิธีการเรียกใช้คอมโพเนนต์ซอฟต์แวร์ประเภท อิน – โพรเซส จาก เครื่องมือพัฒนาอื่น

#### เครื่องมือที่ใช้พัฒนา

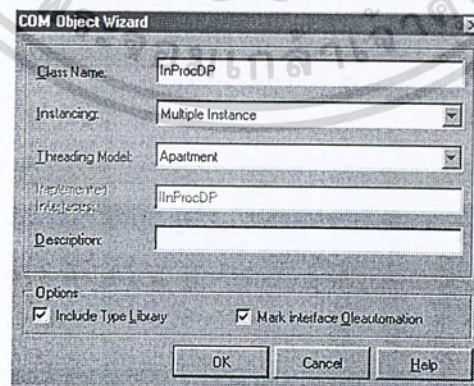
- Borland Delphi 5.0
- Microsoft Visual Basic 6.0

#### คำอธิบายโปรแกรม

- เป็นคอมโพเนนต์ออบเจกต์ ประเภท อิน-โพรเซส ซึ่งมี อินเตอร์เฟส 1 อินเตอร์เฟส
- ภายในอินเตอร์เฟสจะมี เมธอด 2 เมธอด ซึ่งได้แก่ในการรับค่าสตริงจากไคลเอนต์ และเมธอดในการส่งค่าสตริงคืน ไคลเอนต์

#### 5.1.1 ขั้นตอนการพัฒนา

1. สร้างโปรเจกต์ใหม่โดยเลือก *File > New* แล้วเลื่อนไปที่หน้า ActiveX แล้วเลือก ActiveX Library
2. สร้างคอมโพเนนต์ออบเจกต์ใหม่ โดยเลือก *File > New* แล้วเลื่อนไปที่หน้า ActiveX แล้วเลือก COM Object จะขึ้น COM Object Wizard ได้อะลือก ให้ใส่ข้อมูลจนมีลักษณะดัง รูปที่ 5-1

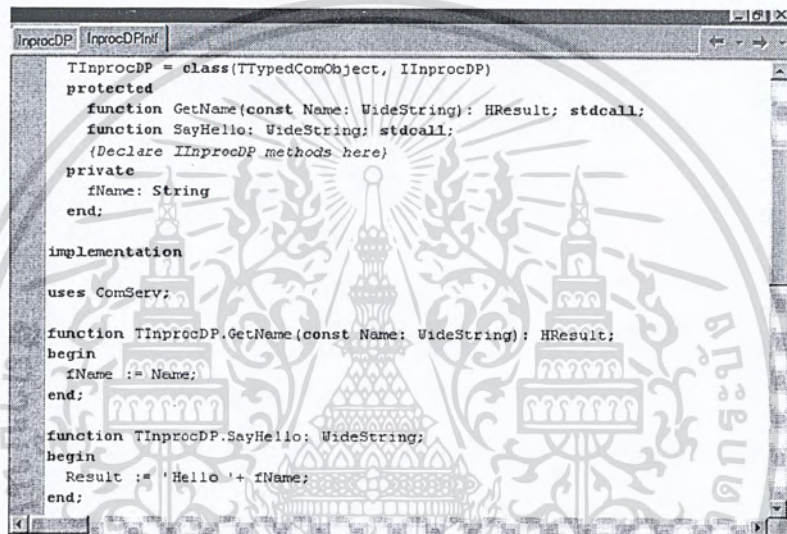


รูปที่ 5-1 แสดงหน้าจอของ COM Object Wizard ในการสร้างอิน-โพรเซส คอมโพเนนต์

3. กด OK จะทำให้เกิดการเรียกใช้ type library editor ให้เพิ่มเมธอด ลงไปใน IInProcDP ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมธอด GetName มี return value เป็น HRESULT, มีพารามิเตอร์ ชื่อ Name มีชนิดเป็น BSTR ซึ่งมี Modifier เป็น [in]
  - เมธอด SayHello มี return value เป็น HRESULT, มีพารามิเตอร์ ชื่อ Result มีชนิดเป็น BSTR \* ซึ่งมี Modifier เป็น [out, retval]
4. กด Refresh เพื่อทำการเปลี่ยนข้อมูลที่กำหนดไว้ใน type library editor ให้เป็น ซอร์สโค้ด
  5. เพิ่ม field เข้าไปใน Class TInprocDP 1 field โดยให้มีชนิดเป็นสตริง
  6. เพิ่มซอร์สโค้ดในส่วนของเมธอด SayHello และGetName ซึ่งลักษณะของซอร์สโค้ดที่ได้ถูกแก้ไขเรียบร้อยแล้วจะมีลักษณะดังรูปที่ 5-2



```

TInprocDP = class(TTypedComObject, IInprocDP)
protected
function GetName(const Name: WideString): HRESULT; stdcall;
function SayHello: WideString; stdcall;
(Declare IInprocDP methods here)
private
fName: String
end;

implementation
uses ComServ;

function TInprocDP.GetName(const Name: WideString): HRESULT;
begin
fName := Name;
end;

function TInprocDP.SayHello: WideString;
begin
Result := 'Hello ' + fName;
end;

```

รูปที่ 5-2 แสดงซอร์สโค้ดของตัวโปรแกรมของ อิน-โพรเซส คอมโพเนนต์

7. ทำการบันทึกไฟล์ทั้งหมดของโปรเจกต์โดย บันทึกไฟล์ของโปรเจกต์ไว้ที่ InprocDP.dpr และ ตัวซอร์สโค้ดจะบันทึกไว้ที่ InprocIntf.pas
8. ทำการรีจิสเตอร์อินเตอร์เฟสโดยเลือก *Run > Register ActiveX Server* หากไม่มีข้อผิดพลาดใด ๆ จะมีการแสดงไดอะล็อกที่บอกว่าการรีจิสเตอร์นั้นได้เสร็จสมบูรณ์แล้ว และจะพบว่าได้มีการสร้างไฟล์ที่ชื่อว่า InprocDP\_TLB.pas ขึ้นมาด้วย
9. ทำการสร้างไฟล์ InprocDP.dll โดยเลือก *Project > Build InprocDP*
10. เสร็จสิ้นการพัฒนาออบเจกต์ คอมโพเนนต์ประเภท อิน-โพรเซส

### 5.1.2 ขั้นตอนการเรียกใช้งาน ด้วย Borland Delphi 5.0

1. สร้างโปรเจกต์ใหม่โดยเลือก *File > New* แล้วเลือก Application จากหน้า New
2. อิมพอร์ตไฟล์ที่เป็น TLB ซึ่งในที่นี้คือ InprocIntf\_TLB.pas โดยเลือก *Project > Add to Project* แล้วเลือกไฟล์ InprocIntf\_TLB.pas จากตำแหน่งใดก็ตามที่บันทึกไว้หรือในไดเรกทอรีที่ได้ copy

เฉพาะส่วนที่เป็น ไฟล์ TLB อย่างเดียวก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพิ่มการเรียกใช้ InprocIntf\_TLB และ COMObj ในส่วนของ uses เช่น  
uses  
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
COMObj, InprocDP\_TLB, StdCtrls;
4. ประกาศตัวแปรประเภท InprocDP ในส่วน Private
5. เมื่อจะทำการเรียกใช้ตัวแปรที่เป็นคอมโพเนนต์ จะต้องมีการสร้างตัวแปรที่เป็นคอมโพเนนต์โดยใช้คำสั่ง

```
a := CreateCOMObject(Class_xxxx) as IntfName;
```

โดย a จะเป็นชื่ออินสแตนซ์ของคอมโพเนนต์

xxxx จะแทนชื่อของคอมโพเนนต์ที่ออบเจกต์เช่นในการทดลองนี้จะเป็น Class\_InprocDP

IntfName จะแทนชื่อของอินเตอร์เฟส เช่น

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    a := CreateCOMObject(Class_InprocDP) as InprocDP;
```

```
    a.GetName(Edit1.Text);
```

```
    Label1.Caption := a.SayHello;
```

```
end;
```

6. หลังจากนั้นจะสามารถใช้งานอินสแตนซ์ของคอมโพเนนต์ได้เหมือนกับออบเจกต์ทั่วไป

### 5.1.3 ขั้นตอนการเรียกใช้งาน ด้วย Microsoft Visual Basic 6.0

1. สร้างโปรเจกต์ใหม่โดยเลือก *File > New Project* แล้วเลือก Standard EXE
2. ทำการอ้างอิงไปยังไฟล์ที่เป็นอิน-โพรเซส โดยทำการเลือก *Project > References* จากนั้นให้ใช้ปุ่ม Browse เพื่อหาไฟล์ DLL ของอิน-โพรเซส เซิร์ฟเวอร์
3. ทำการปรับแต่งและเพิ่มเติม องค์ประกอบของโปรแกรมใหม่นี้ โดยการเรียกใช้อิน-โพรเซสจะใช้คำสั่ง ในลักษณะ

```
Dim a As New InProcDP เพื่อทำการสร้างอินสแตนซ์ใหม่
```

```
Label1.Caption = a.GetStr 'เพื่อเรียกใช้เมธอดของอินเตอร์เฟส
```

ซึ่งการนำไปใช้งานอาจ เป็นดังในรูปที่ 5-3

```

Project1 - Form1 (Code)
(General) (Declarations)

Option Explicit
Dim a As New InProcDP

Private Sub Command1_Click()
Label1.Caption = a.GetfStr
End Sub

Private Sub Command2_Click()
a.SetfStr (Text1.Text)
Label1.Caption = a.GetfStr
End Sub

Private Sub Form_Load()
Label1.Caption = a.GetfStr
End Sub

```

รูปที่ 5-3 แสดงการเรียกใช้ อิน-โพรเซส คอมโพเนนท์โดยใช้ Microsoft Visual Basic 6.0

4. ทำการเรียกใช้งานได้เหมือนกับออบเจกต์ทั่วไป

## 5.2 กรณีศึกษาการพัฒนา คอมโพเนนท์ ออบเจกต์ ประเภท เอาท์-โพรเซส (Out-Process COM Object)

### จุดประสงค์

1. เข้าใจถึงวิธีการในการพัฒนาคอมโพเนนท์ซอฟต์แวร์ประเภท เอาท์ - โพรเซส
2. เข้าใจถึงวิธีการเรียกใช้คอมโพเนนท์ซอฟต์แวร์ประเภท เอาท์ - โพรเซส จาก เครื่องมือพัฒนาตัวเดียวกัน
3. เข้าใจถึงวิธีการเรียกใช้คอมโพเนนท์ซอฟต์แวร์ประเภท เอาท์ - โพรเซส จาก เครื่องมือพัฒนาอื่น

### เครื่องมือที่ใช้พัฒนา

- Borland Delphi 5.0
- Microsoft Visual Basic 6.0

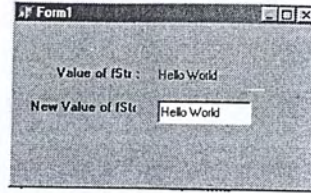
### คำอธิบายโปรแกรม

- เป็นคอมโพเนนท์ออบเจกต์ ประเภท เอาท์-โพรเซส ซึ่งมี ซึ่งติดต่อกับโปรแกรมประยุกต์ที่เก็บค่าตัวแปรชนิดสตริงเอาไว้โดยจะมีอินเตอร์เฟส ที่ติดต่อกับโปรแกรมนี้ภายในอินเตอร์เฟสจะมี เมธอด 2 เมธอด ซึ่งได้แก่ในการรับค่าสตริงจากไคลเอนต์ และเมธอดในการส่งค่าสตริงคืนไคลเอนต์

### 5.2.1 ขั้นตอนการพัฒนา

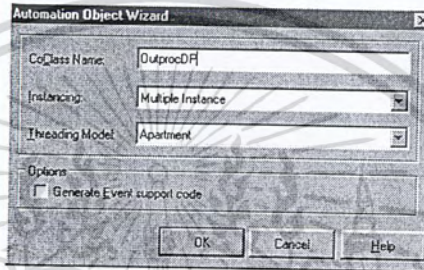
1. สร้าง โปรเจกต์ใหม่โดยเลือก *File > New* แล้วเลื่อนไปที่หน้า Application จากหน้า New
2. แต่งให้โปรแกรมประยุกต์ที่สร้างใหม่มี ลักษณะดังรูปโดยมีไฟล์ภายใน fstr เป็นชนิดสตริงโดยมีอนุญาตให้มีการเปลี่ยนแปลงค่า โดยมีค่าเท่าสตริงที่อยู่ในอิตีทบออกซ์ ดังรูปที่ 5-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



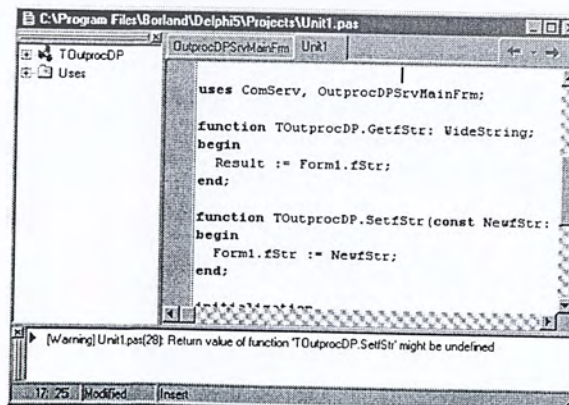
รูปที่ 5-4 แสดงรายละเอียดของตัวเอาต์-โพรเซส เซิร์ฟเวอร์

3. บันทึกโปรแกรมประยุกต์นี้ไว้ในไฟล์ OutprocDPSrvMainFrm.pas และไฟล์ของโปรเจกต์เก็บไว้ที่ OutprocDPSrv.dpr
4. สร้างคอมโพเนนต์ออบเจกต์ใหม่ โดยเลือก *File > New* แล้วเลื่อนไปที่หน้า ActiveX แล้วเลือก Automation Object จะขึ้น Automation Object Wizard ได้อะลือกใส่ข้อมูลจนมีลักษณะดังรูปที่ 5-5



รูปที่ 5-5 แสดงหน้าจอของ Automation Object Wizard ในการสร้างเอาต์-โพรเซส คอมโพเนนต์

5. กด OK จะทำให้เกิดการเรียกใช้ type library editor ให้เพิ่มเมธอด ลงไปใน IOutprocDP ดังนี้
  - เมธอด Setf Str มี return value เป็น HRESULT, มีพารามิเตอร์ ชื่อ NewfStr มีชนิดเป็น BSTR ซึ่งมี Modifier เป็น [in]
  - เมธอด Getf Str มี return value เป็น HRESULT, มีพารามิเตอร์ชื่อ Result มีชนิดเป็น BSTR \* ซึ่งมี Modifier เป็น [out, retval]
6. กด Refresh เพื่อทำการเปลี่ยนข้อมูลที่กำหนดไว้ใน type library editor ให้เป็น ซอร์สโค้ด
7. ทำการ เพิ่มชื่อ OutprocDPSrvMainFrm ลงในส่วนการประกาศ uses ของส่วน interface และใส่ซอร์สโค้ดสำหรับเมธอดทั้งสองของ IOutprocDP ดังรูปที่ 5-6



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
รูปที่ 5-6 แสดงรายละเอียดของเมธอดในตัวเอาต์-โพรเซสคอมโพเนนต์ที่โยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ทำการบันทึกไฟล์ของ IOutprocDP ไว้ที่ OutprocIntf.pas
9. ทำการรีจิสเตอร์อินเตอร์เฟสโดยเลือก Run โปรเจกต์และจะสังเกตพบว่ามีการสร้างไฟล์ที่ชื่อว่า OutprocDPSrv\_TLB.pas ขึ้นมาด้วย
10. ทำการสร้าง OutprocDPSrv.exe โดยเลือก *Project > Build OutprocDPSrv*
11. เสร็จสิ้นการพัฒนาออบเจกต์ คอมโพเนนท์ประเภท เอ้าท์-โพรเซส

### 5.2.2 ขั้นตอนการเรียกใช้งาน ด้วย Borland Delphi 5.0

1. สร้างโปรเจกต์ใหม่โดยเลือก *File > New* แล้วเลือก Application จากหน้า New
2. อิมพอร์ตไฟล์ที่เป็น TLB ซึ่งในที่นี้คือ OutprocIntf\_TLB.pas โดยเลือก *Project > Add to Project* แล้วเลือกไฟล์ OutprocIntf\_TLB.pas จากตำแหน่งไดเรกทอรีที่บันทึกไว้หรือในไดเรกทอรีที่ได้ copy เฉพาะส่วนที่เป็น ไฟล์ TLB อย่างเดียวก็ได้
3. เพิ่มการเรียกใช้ OutprocIntf\_TLB และ COMObj ในส่วนของ uses เช่น  
uses  
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
COMObj, OutprocDP\_TLB, StdCtrls;
4. ประกาศตัวแปรประเภท IOutprocDP ในส่วน Private
5. เมื่อจะทำการเรียกใช้ตัวแปรที่เป็นคอมโพเนนท์ จะต้องมีการสร้างตัวแปรที่เป็นคอมโพเนนท์โดยใช้คำสั่ง

```
a := CreateCOMObject(Class_xxxx) as IntfName;
```

โดยมีรายละเอียดเหมือนในการเรียกใช้ คอมโพเนนท์ออบเจกต์ชนิด อิน-โพรเซส เช่น

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
    a := CreateCOMObject(Class_OutprocDP) as IOutprocDP;
```

```
end;
```

6. หลังจากนั้นจะสามารถใช้งานอินสแตนซ์ของ คอมโพเนนท์ที่ได้เหมือนกับ ออบเจกต์ทั่วไป

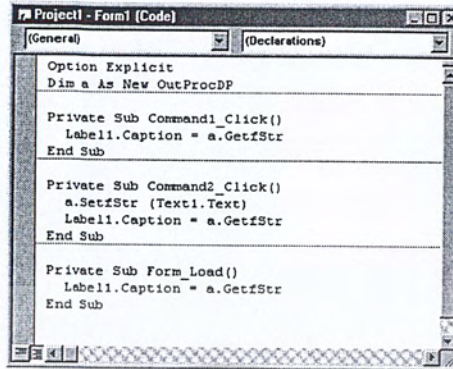
### 5.2.3 ขั้นตอนการเรียกใช้งาน ด้วย Microsoft Visual Basic 6.0

1. สร้างโปรเจกต์ใหม่โดยเลือก *File > New Project* แล้วเลือก Standard EXE
2. ทำการอ้างอิงไปยังไฟล์ที่เป็นเอ้าท์-โพรเซส โดยทำการเลือก *Project > References* จากนั้นให้ใช้ปุ่ม Browse เพื่อหาไฟล์ EXE ของเอ้าท์-โพรเซส เซิร์ฟเวอร์
3. ทำการปรับแต่งและเพิ่มเติม องค์ประกอบของโปรแกรมใหม่นี้ โดยการเรียกใช้เอ้าท์-โพรเซสจะใช้คำสั่ง ในลักษณะ

Dim a As New OutProcDP เพื่อทำการสร้างอินสแตนซ์ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label1.Caption = a.GetfStr เพื่อเรียกใช้เมธอดของอินเตอร์เฟซ  
ซึ่งการอาจเป็นดังในรูปที่ 5-7



```

Project1 - Form1 (Code)
(General) (Declarations)
Option Explicit
Dim a As New OutProcDP

Private Sub Command1_Click()
Label1.Caption = a.GetfStr
End Sub

Private Sub Command2_Click()
a.SetfStr (Text1.Text)
Label1.Caption = a.GetfStr
End Sub

Private Sub Form_Load()
Label1.Caption = a.GetfStr
End Sub

```

รูปที่ 5-7 แสดงตัวอย่างการเรียกใช้งานเอาท์-โพรเซสคอมโพเนนท์

4. ทำการเรียกใช้งานได้เหมือนกับออบเจกต์ทั่วไป

#### ข้อสังเกต

1. ในการใช้ เอาท์โพรเซสเซิร์ฟเวอร์นั้นเมื่อตัวไคลเอนต์ทำการรันโปรแกรมจะส่งผลให้ตัวเซิร์ฟเวอร์นั้นรันโดยอัตโนมัติ
2. การเรียกใช้คอมโพเนนท์ทั้งในรูปแบบ อิน-โพรเซสและ เอาท์-โพรเซส จะมีลักษณะการอ้างอิง และการเรียกใช้เมธอดที่อยู่ภายในคอมโพเนนท์เหมือนกัน

### 5.3 กรณีศึกษาการพัฒนา คอมโพเนนท์ ออบเจกต์ ประเภท รีโมท เซิร์ฟเวอร์ (Remote Server)

#### จุดประสงค์

1. เข้าใจถึงวิธีการในการพัฒนาคอมโพเนนท์ซอฟต์แวร์ประเภท รีโมทเซิร์ฟเวอร์
2. เข้าใจถึงวิธีการเรียกใช้คอมโพเนนท์ซอฟต์แวร์ประเภท รีโมท เซิร์ฟเวอร์ จาก เครื่องมือพัฒนาตัวเดียวกัน
3. เข้าใจถึงวิธีการเรียกใช้คอมโพเนนท์ซอฟต์แวร์ประเภทรีโมท เซิร์ฟเวอร์ จาก เครื่องมือพัฒนาอื่น

#### เครื่องมือที่ใช้พัฒนา

- Borland Delphi 5.0
- Microsoft Visual Basic 6.0

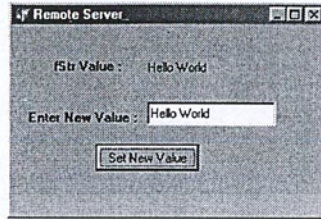
#### คำอธิบายโปรแกรม

- เป็นคอมโพเนนท์ออบเจกต์ ประเภท รีโมท เซิร์ฟเวอร์ ซึ่งมี อินเตอร์เฟซ 1 อินเตอร์เฟซ
- ภายในอินเตอร์เฟซจะมี เมธอด 2 เมธอด ซึ่งได้แก่ในการรับค่าสตริงจากไคลเอนต์ และเมธอดในการส่งค่าสตริงคืนไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.1 ขั้นตอนการพัฒนา

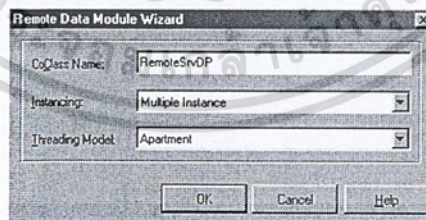
1. สร้างโปรเจกต์ใหม่โดยเลือก *File > New* แล้วเลื่อนไปที่หน้า Application จากหน้า New
2. แต่งให้โปรแกรมประยุกต์ที่สร้างใหม่มี ลักษณะดังรูปโดยมีฟิลด์ภายใน fStr เป็นชนิดสตริง โดยมีอนุญาตให้มีการเปลี่ยนแปลงค่าโดยมีค่าเท่าสตริงที่อยู่ในอติทบอกร์ ดังรูปที่ 5-8



รูปที่ 5-8 แสดงลักษณะของรีโมท เซิร์ฟเวอร์

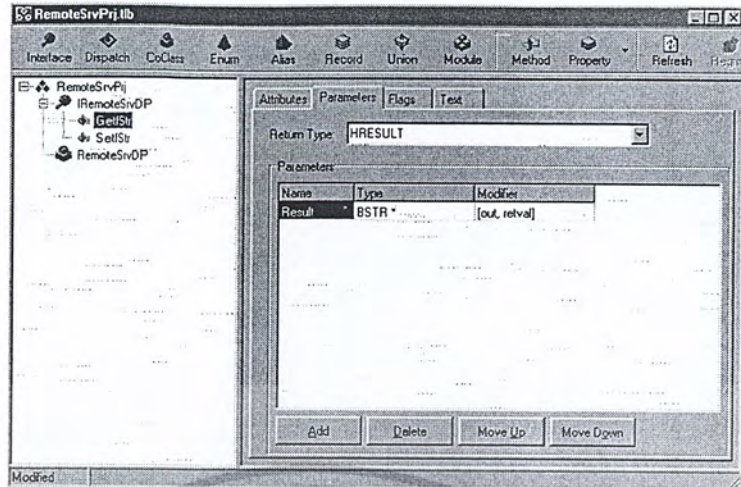
3. บันทึกโปรแกรมประยุกต์นี้ไว้ในไฟล์ RemoteSrvMainFrm.pas และไฟล์ของโปรเจกต์เก็บไว้ที่ RemoteSrv.dpr
4. เพิ่ม Remote Data Module โดยเลือก *File > New* หลังจากนั้นเลือก Remote Data Module จากหน้า Multitier ตัว Remote Data Module ที่สร้างขึ้นนั้นเป็นตัวอินเตอร์เฟสด้วย มันจะถูกเรียกใช้โดยตัวไคลเอนต์ตัวคอมโพเนนท์เซิร์ฟเวอร์ตัวนี้ เมื่อเลือกสร้าง Remote Data Module จะขึ้นหน้าจอของ Remote Data Module Wizard ให้ใส่รายละเอียดที่ต้องการไป ดังรูปที่ 5-9 หลังจากทำการปรับแต่งจนเหมาะสมแล้วให้กดปุ่ม OK เพื่อทำงานต่อ
5. กด OK จะทำให้เกิดการเรียกใช้ type library editor ให้เพิ่มเมธอด ลงไปใน IOUtprocDP ดังนี้
  - เมธอด Setf Str มี return value เป็น HRESULT, มีพารามิเตอร์ ชื่อ NewfStr มีชนิดเป็น BSTR ซึ่งมี Modifier เป็น [in]

เมธอด Getf Str มี return value เป็น HRESULT, มีพารามิเตอร์ ชื่อ Result มีชนิดเป็น BSTR \* ซึ่งมี Modifier เป็น [out, retval] ดังรูปที่ 5-10



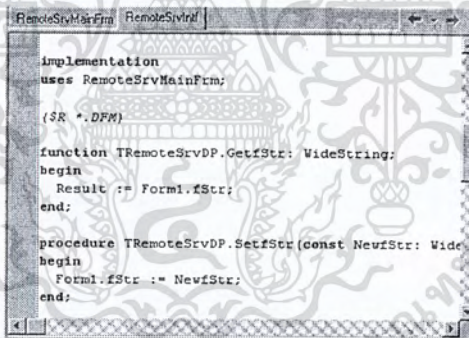
รูปที่ 5-9 แสดงหน้าจอของ Remote Data Module ในการสร้างรีโมท เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-10 แสดงหน้าจอของ *type library editor* ในการเพิ่ม เมธอดและคุณสมบัติต่างๆ ให้กับอินเตอร์เฟส

6. กด Refresh เพื่อทำการเปลี่ยนข้อมูลที่กำหนดไว้ใน *type library editor* ให้เป็น ชอร์ตโค้ด
7. ทำการ เพิ่มชื่อ RemoteSrvMainFrm ลงในส่วนการประกาศ *uses* ของส่วน *interface* และใส่ชอร์ตโค้ดสำหรับเมธอดทั้งสองของ *IRemoteSrvDP* ดังรูปที่ 5-11



รูปที่ 5-11 แสดงการเพิ่มเติมรายละเอียดในส่วน *Remote Data Module*

8. ทำการบันทึกไฟล์ของ *IRemoteSrvDP* ไว้ที่ *RemoteSrvDPIntf.pas*
9. ทำการรีจิสเตอร์อินเตอร์เฟสโดยเลือก *Run* โปรเจกต์และจะสังเกตเห็นว่าการสร้างไฟล์ที่ชื่อว่า *RemoteDPSrvPrj\_TLB.pas* ขึ้นมาด้วย
10. ทำการสร้างไฟล์ *RemoteDPSrvPrj\_TLB.pas* โดยการเลือก *Project > Build RemoteSrvPrj*
11. เสร็จสิ้นการพัฒนาออบเจกต์ คอมโพเนนท์ประเภท รีโมท เซิร์ฟเวอร์

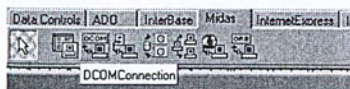
### 5.3.2 ขั้นตอนการเรียกใช้งาน ด้วย Borland Delphi 5.0

1. สร้างโปรเจกต์ใหม่โดยเลือก *File > New* แล้วเลือก *Application* จากหน้า *New*
2. ทำการรีจิสเตอร์ตัวเซิร์ฟเวอร์ ไว้ในเครื่องของไคลเอนต์โดยการ รันตัวโปรแกรมเซิร์ฟเวอร์แล้วต่อ

เอกสารนี้เป็นที่ด้วย /regserver เช่น \\Chead\c\Project\Report\RemoteDP\RemoteSrvPrj.exe /regserver ด้านการค้า

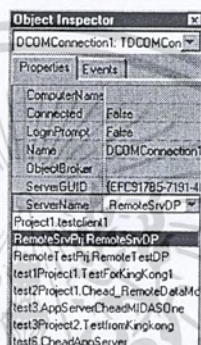
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการเพิ่มออบเจ็กต์ชนิด DCOMConnection จากหน้า Midas ลงในโปรเจกต์ ดังรูปที่ 4-12 ออบเจ็กต์ตัวนี้จะทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างเซิร์ฟเวอร์ และไคลเอนต์ดังจะได้อธิบายต่อไป



รูปที่ 5-12 แสดงการเลือก DCOMConnect คอมโพเนนท์

4. ทำการเลือกตัวเซิร์ฟเวอร์ที่ต้องการจะติดต่อผ่านตัว DCOMConnection โดยการตั้งค่าคุณสมบัติ ServerName ดังรูปที่ 5-13



รูปที่ 5-13 แสดงการเลือกตัวรีโมท เซิร์ฟเวอร์ในคุณสมบัติ ServerName

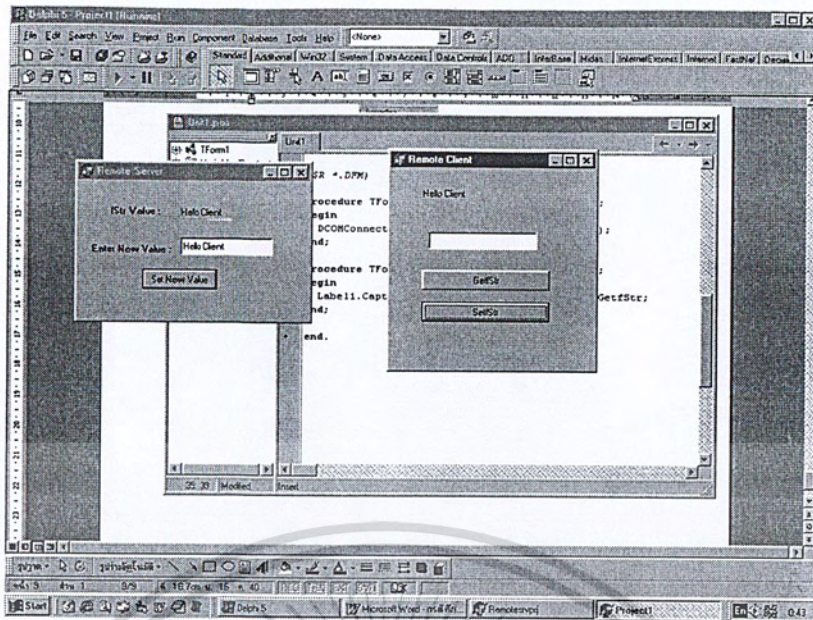
5. ทำการตั้งค่า Connection ให้เป็น true จะสังเกตได้ว่าตัวเซิร์ฟเวอร์นั้นจะถูกเรียกใช้งานโดยอัตโนมัติ
6. ในการเรียกใช้เซิร์ฟเวอร์นั้นไม่จำเป็นต้องสร้างออบเจ็กต์ใหม่แต่ อย่งใดแต่ละจะเรียกใช้ผ่านทางตัว DCOMConnection.AppServer แทน เช่นหากตัว DCOMConnection มีชื่อว่า DCOMConnection1 ก็สามารถใช้เมธอด GetStr ได้ดังนี้

```
DCOMConnection1.AppServer.SetfStr('Hello World');
```

7. หลังจากนั้นจะสามารถใช้งานอินสแตนซ์ของ คอมโพเนนท์ได้เหมือนกับ ออบเจ็กต์ทั่วไป  
ข้อสังเกต

1. เมื่อมีการเรียกตัวไคลเอนต์ทำงานตัวเซิร์ฟเวอร์จะถูกเรียกใช้งานโดยอัตโนมัติ ดังรูปที่ 4-14
2. หากก่อนการกำหนดค่าของ DCOMConnection นั้นไม่ได้มีการริจิสเตอร์ตัวเซิร์ฟเวอร์ในเครื่องไคลเอนต์ จะต้องกำหนดค่า ComputerName ของเครื่องServer เสียก่อนจึงจะเรียกใช้ได้ เนื่องจากการใส่ค่า ComputerName จะมีความหมายว่าตัวเซิร์ฟเวอร์นั้นอยู่ในเครื่องไคลเอนต์ หรือ ตัวเซิร์ฟเวอร์นั้นได้ถูกริจิสเตอร์ไว้ในเครื่องไคลเอนต์แล้ว
3. ในการใช้งานเมธอดของรีโมทเซิร์ฟเวอร์บนเครื่องไคลเอนต์โดยผ่าน DCOMConnection นั้นจะไม่สามารถใช้ type library ผ่านทางปุ่ม Ctrl + Space เพื่อดูรายละเอียดของเมธอดภายในตัวเซิร์ฟเวอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-14 แสดงการรันไคลเอนต์จะให้ตัวเซิร์ฟเวอร์รันด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### ActiveX Data Object (ADO)

ActiveX Data Object เป็นเซตของคอมโพเนนต์ใน Universal Data Access ช่วยให้โปรแกรมเมอร์เข้าถึงข้อมูลได้ ทั้งแบบที่เป็น Relational Database และ โครงสร้างข้อมูลแบบอื่นๆ เช่น เมลล์

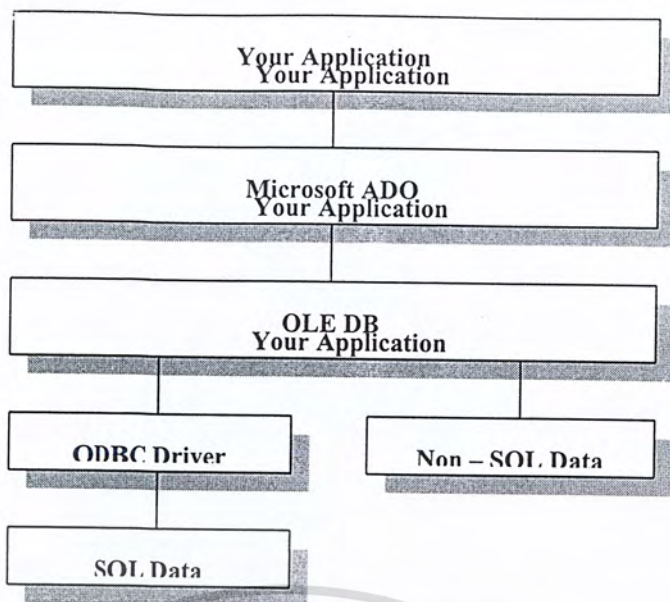
ในการใช้งานเราจะใช้ ADO ติดต่อกับ OLE DB เพื่อให้การทำงานมีประสิทธิภาพสูงขึ้น เพราะ OLE DB มีความสามารถสูงแต่การใช้งานนั้นใช้ได้ยาก จึงใช้ผ่าน ADO ซึ่งการใช้งาน ADO นั้นใช้ได้ง่ายกว่ามาก การทำงานของ ADO นั้นใช้แนวคิดของ Provider ในการเข้าถึงข้อมูล เช่น ถ้าเราใช้ ODBC ตัว Provider ก็เหมือนกับ ODBC Driver ตัวที่เป็น Provider จะต้องเผยให้เห็นถึงเซตของอินเตอร์เฟสให้ ADO สามารถเข้าถึง ทำให้ ADO สามารถเข้าถึงข้อมูลที่แตกต่างกันได้อย่างถูกต้อง

#### 6.1 ADO และ OLE DB

ADO เตรียม API ให้โปรแกรมเมอร์ใช้ในการทำงานได้อย่างมีประสิทธิภาพ และมีความเป็นไปได้ว่าการพัฒนาในอนาคตจะหันมาใช้ ADO กันอย่างแพร่หลาย

การใช้ ADO นั้นมีข้อได้เปรียบมากมาย เช่น ความเร็วสูง, ใช้หน่วยความจำน้อย, ประหยัดพื้นที่, ใช้เครือข่ายอย่างประหยัด, มีอินเตอร์เฟสที่มีประสิทธิภาพสูง จึงเป็นทางเลือกหนึ่งที่ดี ในการทำงาน ADO จะไปเรียก OLE DB ตัว OLE DB นี้ช่วยให้การเข้าถึงข้อมูลได้หลากหลายมากขึ้น แม้ว่าการใช้ ODBC จะมีความสามารถในการเข้าถึงข้อมูลได้หลายรูปแบบในระบบของ ดาตาเบส ต่างๆ แต่ก็มีข้อจำกัดเนื่องจาก ODBC ไม่ได้เตรียมการเข้าถึงข้อมูลไว้ทุกรูปแบบ จึงมีการใช้ OLE DB มาช่วยในการทำงานทำให้การเข้าถึงข้อมูลทำได้มากขึ้นคือช่วยให้สามารถเข้าถึงข้อมูลแบบที่ไม่ใช่ Relation ได้ด้วย แต่การใช้ OLE DB ก็มีปัญหาเหมือนกันคือการใช้งาน OLE DB นั้นทำได้ยาก มีความซับซ้อนในการใช้งานสูง และเพื่อการใช้งาน OLE DB ให้ได้มีประสิทธิภาพที่ดี และใช้ได้ง่ายขึ้นจึงใช้ ADO มาติดต่อกับ OLE DB อีกที

ADO จึงเป็นทางเลือกที่ดีสำหรับการเขียนโปรแกรมระดับสูงโดยผู้ใช้ไม่จำเป็นต้องเข้าใจการใช้ COM อินเตอร์เฟส ในระดับต่างๆ ทำให้การใช้ง่ายเป็นไปได้ง่าย



รูปที่ 6-1 แสดงการทำงานของ ADO

## 6.2 ออบเจ็กต์ ADO

สถาปัตยกรรมของ ADO ประกอบด้วยออบเจ็กต์เพียงไม่กี่ตัวที่สำคัญ ดังนี้

- Connection เป็นออบเจ็กต์ที่ใช้ในการติดต่อกับแหล่งข้อมูล โดยใช้ Connection strings มาเป็นตัวบอกว่าจะติดต่อกับข้อมูลชนิดใด ช่วยจัดการในการเชื่อมต่อและดูแลทรานแซกชัน
- Command จัดการกับแหล่งข้อมูล เช่นการ เพิ่ม, แก้ไข, ลบ ข้อมูล พวงง่ายๆ คือจัดการกับข้อมูลโดยไม่ต้องการคำสั่งคืนกลับ
- Recordset เป็นออบเจ็กต์ที่เก็บคำสั่งคืนมาจากการ Query

ADO ยังมีออบเจ็กต์อื่นๆ อีก คือ Error เป็นออบเจ็กต์ที่จัดการเกี่ยวกับความผิดพลาด, Property เป็นออบเจ็กต์ที่บอกถึงคุณสมบัติของตัวเอง เป็นต้น

## 6.3 คอมโพเนนต์ของ ADO ใน Borland Delphi 5.0

คอมโพเนนต์ ของ ADO ใน Delphi สืบทอดมาจากคลาส TDataset มีอยู่สามคอมโพเนนต์ที่สามารถใช้ในการเขียนแอปพลิเคชันที่ทำงานกับดาตาเบส ดังนี้

- TADODConnection เป็นคอมโพเนนต์ที่หุ้มออบเจ็กต์ ADO Connection ไว้ ช่วยในการสร้าง Connection strings, การ login, ทรานแซกชัน คล้ายๆ กับ TDatabase
- TADODCommand เป็นคอมโพเนนต์ที่หุ้มออบเจ็กต์ ADO Command ไว้ ทำงานแบบที่ไม่ต้องการคำสั่งคืนกลับ
- TADODDataSet เป็นคอมโพเนนต์ที่หุ้มออบเจ็กต์ สองตัวด้วยกันคือ ADO Command และ ADO Recordset ไว้ด้วยกันใช้ในการออกคำสั่งจัดการกับตารางหนึ่งตาราง หรือหลายๆ ตารางก็ได้ และรับเซตของเรคอร์ดที่ส่งคืนกลับมา

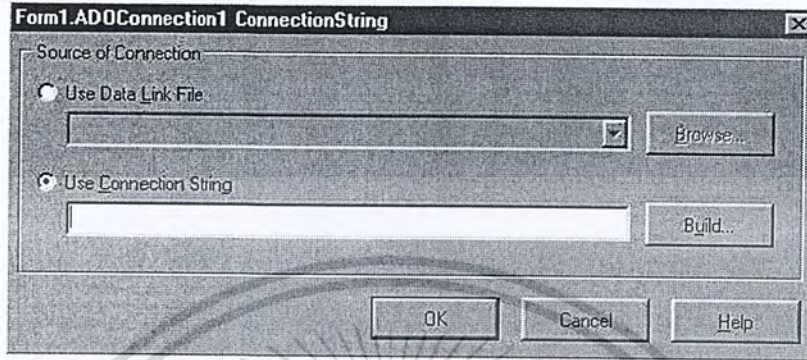
ทั้งสามคอมโพเนนต์จำเป็นในการสร้างแอปพลิเคชันด้วย Delphi แต่ก็มีคอมโพเนนต์อื่นๆ ให้ใช้ได้

อีกนอกจากสามตัวที่ได้กล่าวไป คือ TADODTable, TADODQuery, TADODStoredProc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

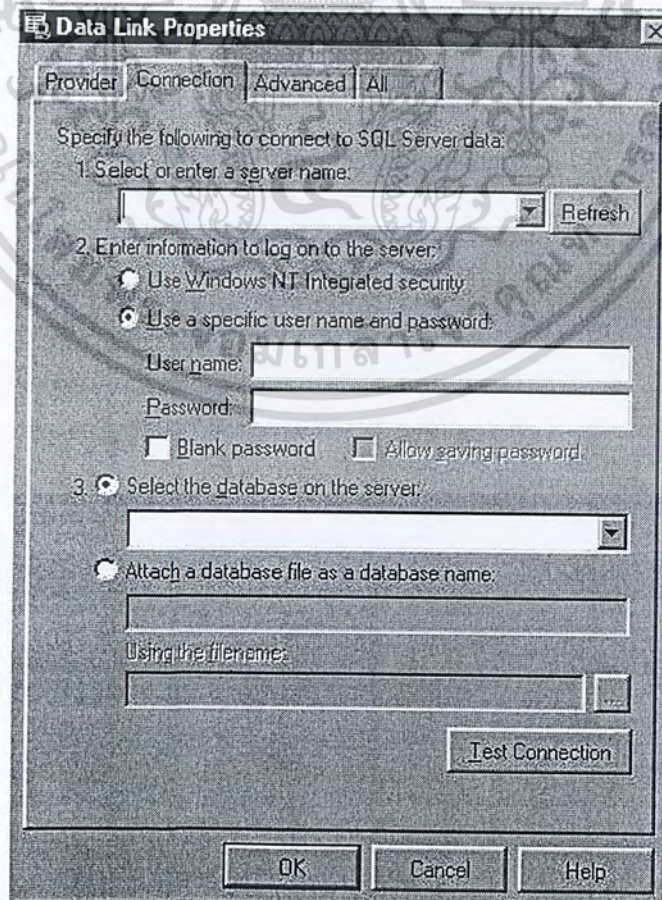
#### 6.4 การใช้คอมโพเนนต์ ADO ใน Borland Delphi 5.0

ได้กล่าวไปแล้วถึงแนวคิดของ ADO ต่อจากนี้เป็นการใช้งานคอมโพเนนต์ในการสร้างแอปพลิเคชันขึ้นมาให้ติดต่อกับดาตาเบสได้เริ่มจากการใช้ TADOConnection ในการติดต่อไปยังดาตาเบสที่เราต้องการ โดยการเลือก Connection strings ดังรูปที่ 6-2



รูปที่ 6-2 แสดงหน้าจอในการเลือกสร้าง Connection String

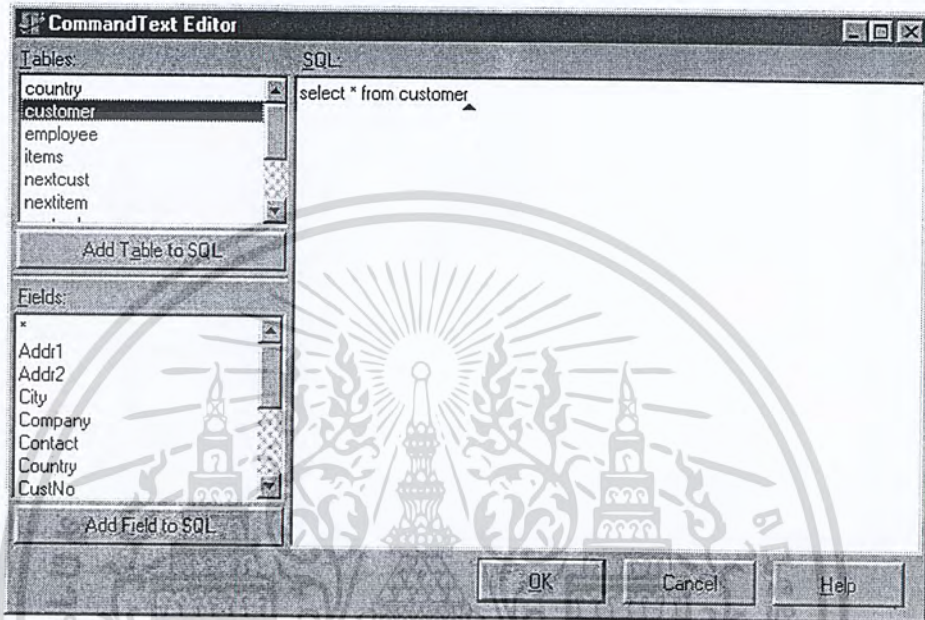
เลือก Use Connection string จะมีกล่องข้อความใหม่ปรากฏขึ้นมาใหม่ให้เลือก OLE DB Provider ที่เราต้องการใช้งาน เช่น Provider สำหรับ Oracle, Provider สำหรับ SQL Server เป็นต้น ต่อไปก็เป็นการกำหนดเงื่อนไขในการติดต่อ เช่น ชื่อเซิร์ฟเวอร์, ชื่อผู้ใช้, ชื่อดาตาเบส ดังรูปที่ 6-3



รูปที่ 6-3 แสดงหน้าจอในการสร้าง Connection String

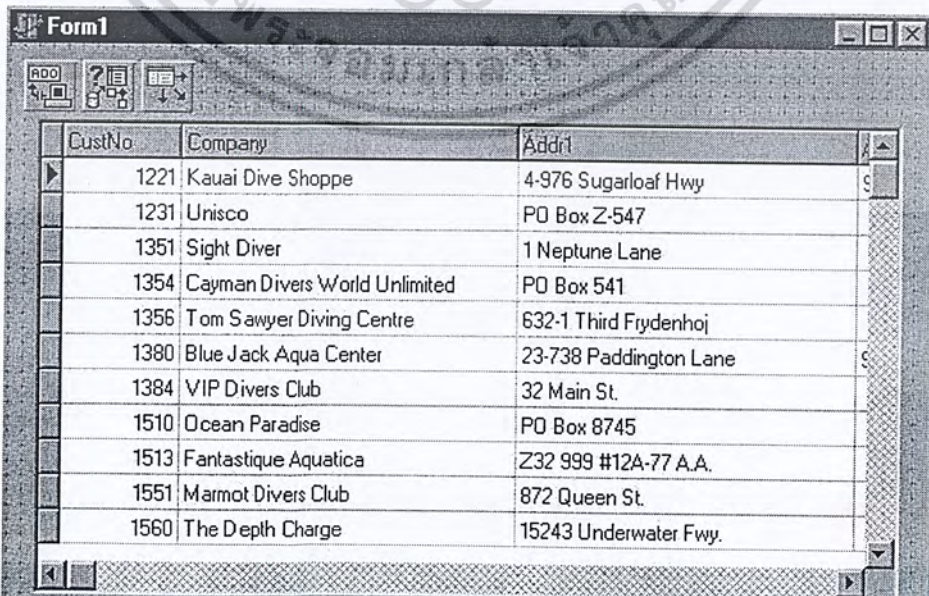
หลังจากตั้งค่าของ ADOConnection เสร็จเรียบร้อยแล้วเราสามารถเพิ่ม TADODataset เข้ามาใช้งานโดยตั้งค่า Connection เป็น ADOConnection ตัวคอมโพเนนต์ ADODataset ก็จะติดต่อกาตาเบสผ่านทาง ADOConnection ได้โดยกำหนดค่า CommandText เมื่อเลือกไปจะเห็น CommandText Editor ปรากฏขึ้นมา ดังรูปที่ 6-4

เราสามารถใส่คำสั่ง SQL ลงไปได้ตามต้องการต่อจากนั้น ก็ใช้ DataSource และ DBGrid ในการ



รูปที่ 6-4 แสดงหน้าจอในการสร้างคำสั่งภาษา SQL

แสดงเขตของข้อมูลที่ได้จากการ Query โดยกำหนดค่า DataSet ของ DataSource เป็น ADODataset และกำหนดค่า DataSource ของ DBGrid เป็น DataSource เท่านั้นก็จะเห็นเขตของข้อมูลที่เราร Query ออกมาดังรูปที่ 6-5



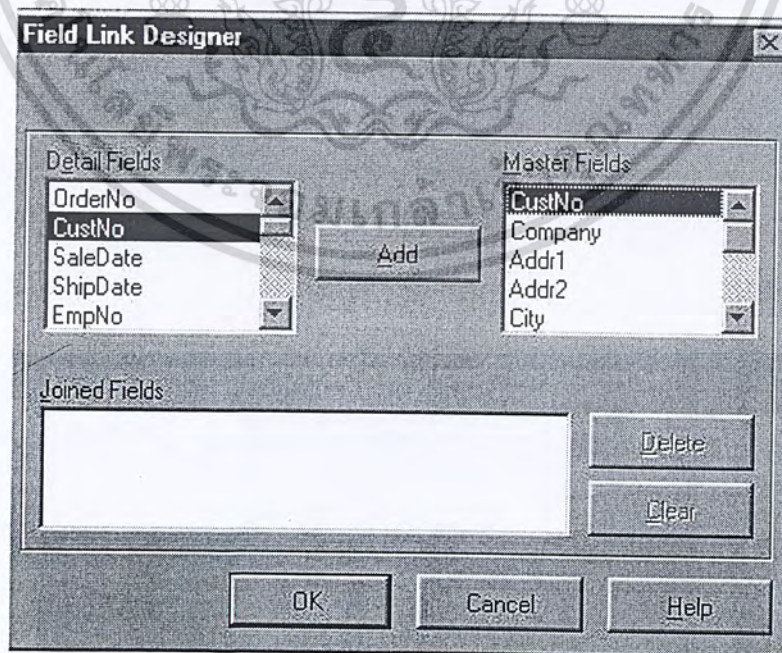
รูป 6-5 แสดงผลลัพธ์ จากโปรแกรมทดสอบที่ติดต่อกับฐานข้อมูลสำเร็จ

## 6.5 โครงสร้างแบบ Master/Detail

คอมโพเนนต์ ADODataset สนับสนุนการทำงานให้มีความสัมพันธ์แบบ Master/Detail รูปแบบการทำงานแบบ Master/Detail ก็คือการที่มีตารางหนึ่งเป็นตาราง Master มีความสัมพันธ์กับตารางอื่นที่เป็นตาราง Detail ซึ่งอาจมีได้มากกว่าหนึ่งตารางโดยที่ตารางที่เป็น Detail นั้นต้องมี Foreign key ที่เป็น Primary key ของตารางที่เป็น Master อยู่ในตารางด้วยทุกตาราง เพื่อเป็นตัวอ้างอิงในการรวมตาราง (Join) เข้าด้วยกัน ได้ถูกต้อง ตัวอย่างของการแสดงข้อมูลที่สามารถเป็นแบบ Master/Detail ได้ เช่น การเก็บข้อมูลของนักศึกษา มีตาราง Master เป็นข้อมูลเกี่ยวกับตัวนักศึกษาเอง รูปแบบของตาราง Master อาจมี รหัสนักศึกษา ชื่อ นามสกุล วัน/เดือน/ปี เกิด ที่อยู่ คณะ ภาควิชา เป็นต้น ส่วนตารางที่เป็น Detail ก็อาจมีหลายตาราง อย่างเช่น ตารางของการลงทะเบียนเรียน ตารางของผลการเรียน เป็นต้น เมื่อใช้โครงสร้างแบบ Master/Detail มาแสดงก็จะเห็นข้อมูลของนักศึกษาแต่ละคนได้อย่างชัดเจน คือเมื่อตารางที่เป็น Master มีการเลือกไปที่ข้อมูลในตาราง จะทำให้ตารางที่เป็น Detail เกิดการนำข้อมูลที่เกี่ยวข้องกับนักศึกษานั้นที่ถูกเลือกนำขึ้นมาแสดง โดยการรวมตาราง (Join) ตารางทุกตารางที่เป็น Detail ทำให้การแสดงผลข้อมูลเป็นไปได้ง่าย การทำงานแบบ Master/Detail นั้นเราสามารถใช้คำสั่งในการรวมตารางเองก็ได้

## 6.6 การใช้โครงสร้างแบบ Master/Detail ใน Borland Delphi 5.0

เริ่มแรกต้องมีตารางที่เป็น Master และ Detail กำหนดค่า CommandType ของ ADODataset ของทั้งสองตารางให้เป็น cmdTable และกำหนด CommandText เป็นชื่อตารางที่ต้องการ ตารางที่เป็น Master กำหนดเสร็จเรียบร้อยแล้ว ที่เหลือตั้งค่า DataSource ของ ADODataset ที่เป็น Detail ให้เป็น DataSource ของตาราง Master เสร็จแล้วกำหนดค่า MasterFields จะปรากฏกล่องข้อความดังรูปที่ 6-6



รูปที่ 6-6 แสดง หน้าจอของโปรแกรมตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือก Key ที่จะนำมาใช้เป็นตัวกลางในการรวมตารางเข้าด้วยกัน กด Add เป็นการเสร็จการกำหนดค่าของตาราง Detail เสร็จแล้วจะได้แอปพลิเคชันง่าย ดังรูปที่ 6-7

CustNo	Company	Addr1
1221	Kauai Dive Shoppe	4-976 Sugarloaf Hwy
1231	Unisco	PO Box Z-547
1351	Sight Diver	1 Neptune Lane
1354	Cayman Divers World Unlimited	PO Box 541
1356	Tom Sawyer Diving Centre	632-1 Third Frydenhoj
1380	Blue Jack Aqua Center	23-738 Paddington Lane

OrderNo	CustNo	SaleDate	ShipDate	EmpNo	ShipToContact
1023	1221	1/7/88	2/7/88	5	
1169	1221	6/7/94	6/7/94	12	
1269	1221	16/12/94	16/12/94	28	

รูปที่ 6-7 แสดงผลลัพธ์ของการทดสอบลักษณะ Master/Detail

จะเห็นได้ว่าตารางที่เป็น Detail (ด้านล่าง) จะขึ้นกับตารางที่เป็น Master (ด้านบน) ทั้งสองตารางสัมพันธ์กันด้วย CustNo

#### 6.7 การจัดการทรานแซกชันด้วย ADO

ในการทำงานเกี่ยวกับดาตาเบสเราอาจจะใช้ทรานแซกชันเข้ามาช่วย ทรานแซกชันเป็นการทำงานที่เป็นหนึ่งเดียวจะแบ่งแยกไม่ได้ พุดง่าย ๆ ก็คือ การทำงานที่เป็นทรานแซกชันถ้าทำแล้วต้องทำให้เสร็จทั้งหมดถ้าทำไม่เสร็จต้องเหมือนไม่เคยทำเลยยกเลิกที่ทำไม่เสร็จให้หมด

ใน ADO เองมีการจัดการการทำทรานแซกชันให้ได้ คือสามารถทำการ Start Transaction, Commit Transaction, Rollback Transaction ได้ช่วยให้การทำงานมีความถูกต้อง และความปลอดภัยมากขึ้นในการใช้งานคำสั่งเกี่ยวกับการจัดการทรานแซกชันนั้นมีเมธอดอยู่ในคอมโพเนนต์ ADOConnection ให้เราใช้คือ BeginTrans, CommitTrans, RollbackTrans ทั้งสามตัวใช้ในการกำหนด ทรานแซกชัน

## บทที่ 7

## อีเวนต์ และ คอลแบ็กของ COM

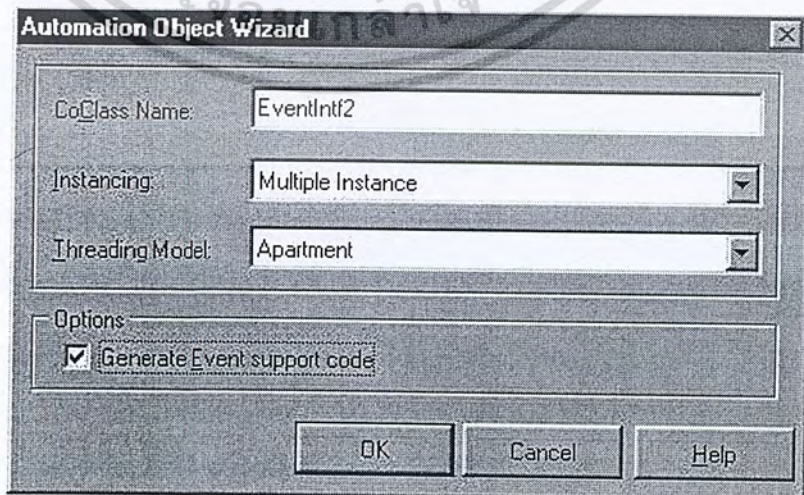
ในรูปแบบของโมเดลแบบโคลเอนต์/เซิร์ฟเวอร์ ตัวโคลเอนต์จะเรียกไปยังเซิร์ฟเวอร์เพื่อทำงานบางอย่างหรือเรียกดูข้อมูล แล้วตัวเซิร์ฟเวอร์จะทำการประมวลงานนั้น หรือส่งข้อมูลไปให้โคลเอนต์ซึ่งเป็นรูปแบบที่ดี แต่มีปัญหาเมื่อมีอีเวนต์ในฝั่งโคลเอนต์เป็นแบบอะซิงโครนัส หรือทำงานตามการใช้งานของผู้ใช้ ตัวอย่างเช่น โคลเอนต์ร้องขอไปยังเซิร์ฟเวอร์เพื่อดาวน์โหลดไฟล์ตัวโคลเอนต์นั้นไม่ต้องการรอจึงมีรูปแบบในการใช้อีเวนต์เกิดขึ้นโดยตัวโคลเอนต์จะทำการร้องขอไปที่เซิร์ฟเวอร์แล้วกลับมาทำงานต่อได้จนกระทั่งเซิร์ฟเวอร์แจ้งกลับมาที่โคลเอนต์ว่าการดาวน์โหลดเสร็จแล้ว หรืออีกตัวอย่างหนึ่งคือการใช้งานของผู้ใช้ เช่นการกดปุ่มจะเกิดอีเวนต์ขึ้นบอกไปยังเซิร์ฟเวอร์แล้วเซิร์ฟเวอร์ทำการเรียกเมธอดที่สนับสนุนอีเวนต์นั้นให้

ลักษณะของออโตเมชันอีเวนต์นั้นเป็นโมเดลที่มีประสิทธิภาพแต่ซับซ้อนการใช้อีเวนต์นั้นจะเรียกผ่านอินเตอร์เฟส โดยอินเตอร์เฟสนี้เรียกว่า อีเวนต์อินเตอร์เฟส หรือ เอาท์โกอิงอินเตอร์เฟส (Outgoing Interface) ที่เรียกว่า เอาท์โกอิงอินเตอร์เฟสเพราะว่าการเขียนเมธอดไม่ได้ทำที่เซิร์ฟเวอร์เหมือนอินเตอร์เฟสอื่นๆ แต่จะทำได้ที่โคลเอนต์ของเซิร์ฟเวอร์นั้นๆ โดยอินเตอร์เฟสนี้จะเหมือนกับอินเตอร์เฟสต่างๆ ไปเช่นมี IID เป็นต้น

ใน Delphi เราจะใช้ DispInterface ในการสร้างอีเวนต์ เพราะสามารถเข้ากันได้กับ Visual Basic ได้ ถ้าเราใช้อินเตอร์เฟสธรรมดาในการสร้างอีเวนต์จะทำงานได้เร็วขึ้นกว่า DispInterface แต่ไม่สามารถเข้ากันได้กับ Visual Basic ได้ และต้องเขียนโค้ดมากขึ้น

### 7.1 การสร้างออโตเมชันเซิร์ฟเวอร์ในการใช้งานอีเวนต์

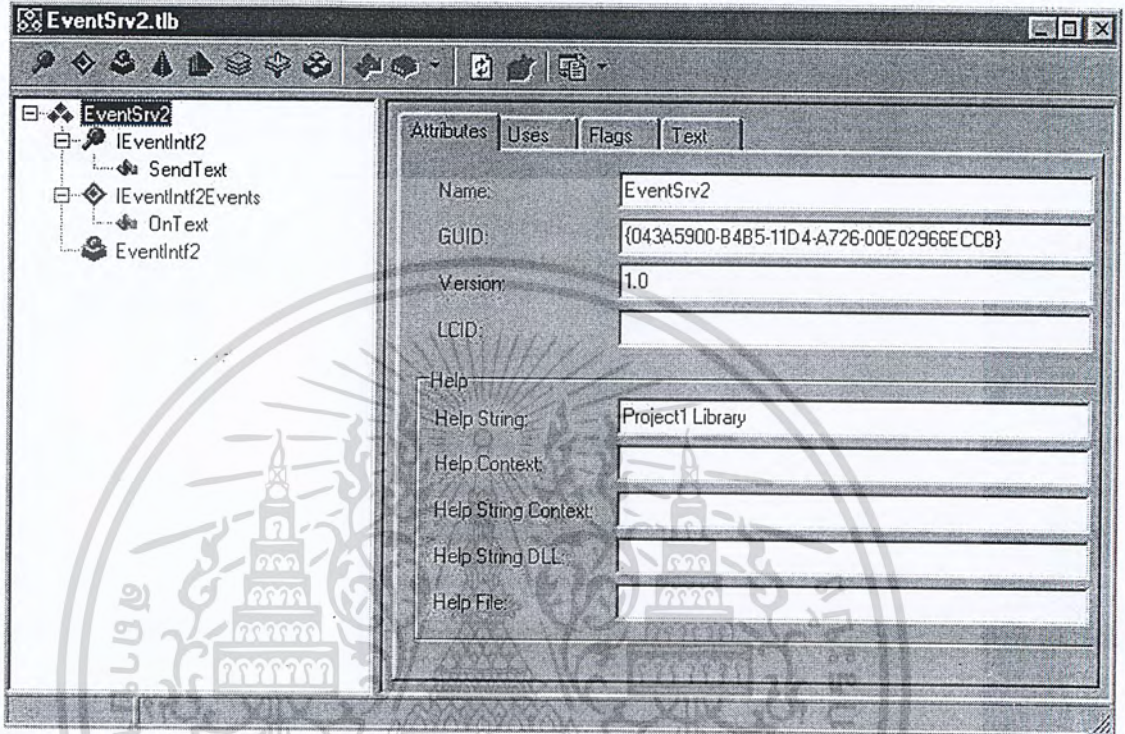
เริ่มจากการสร้างแอปพลิเคชันง่ายๆ มาหนึ่งแอปพลิเคชันแล้วมาการเพิ่มออโตเมชันออบเจกต์เข้าไปตั้งชื่อแล้วเช็คที่เช็คบ็อก Generate Event support code ด้วย ดังรูปที่ 7-1



รูปที่ 7-1 แสดงหน้าจอในการเพิ่มคุณสมบัติอีเวนต์เข้ากับคอมโพเนนท์ใหม่ใน Borland Delphi 5.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Borland Delphi 5.0 จะสร้างอินเตอร์เฟสให้สองอินเตอร์เฟสสำหรับ COM ออบเจ็กต์หนึ่ง และสำหรับอีเวนต์อีกหนึ่งในไทป์ไลบรารีแล้วทำการเพิ่มเมธอดไปยัง IEventIntf2 ชื่อ SendText ให้มี พารามิเตอร์ชื่อ Text เป็นชนิด WideString ต่อไปทำการเพิ่มอีเวนต์ใน IEventIntf2Events ชื่อ OnText มี พารามิเตอร์ชื่อ Text ชนิด WideString แล้วกดปุ่ม Refresh จะเห็นไทป์ไลบรารี ดังรูปที่ 7-2



รูปที่ 7-2 แสดงการกำหนดเมธอดของ อินเตอร์เฟส IeventIntf2 และ IEventIntf2Events

ทำการเขียนโค้ดในเมธอด SendText ดังนี้

```

Procedure TeventIntf2.SendText(const Text : WideString)
Begin
    FEvents.OnText(Text);
End;

```

ทำการบันทึกเป็นการเสร็จสิ้นการสร้างเซิร์ฟเวอร์

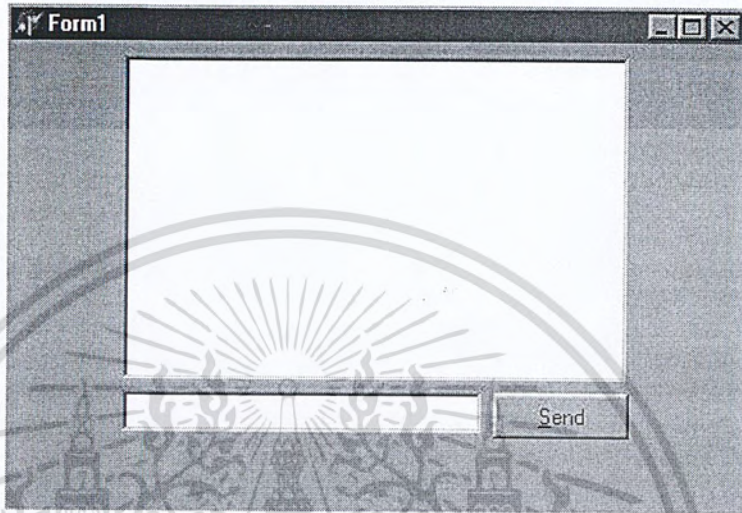
### 7.2 การสร้างไคลเอนต์ในการใช้งานอีเวนต์

การสร้างไคลเอนต์ทำได้โดยสร้างแอปพลิเคชันใหม่ขึ้นมาแล้วทำการอิมพอร์ตไทป์ไลบรารีเข้ามา จะได้คอมโพเนนต์ชื่อ TEventIntf2 ออกมาบนคอมโพเนนต์พาเลตทำการเรียกใช้ได้เหมือนคอมโพเนนต์ทั่วไป

ตัว TEventIntf2 ที่เราสร้างขึ้นจะมีคุณสมบัติที่น่าสนใจอยู่ 3 ตัว คือ AutoConnect เป็นการกำหนดให้แอปพลิเคชันทำการติดต่อกับเซิร์ฟเวอร์โดยอัตโนมัติเมื่อถูกเรียกใช้งาน ตัวที่สองคือ เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของบริษัท ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ConnectKind ใช้บอกว่าจะติดต่อกับเซิร์ฟเวอร์อย่างไร เช่น ติดต่อกับรีโมตเซิร์ฟเวอร์, ติดต่อกับอินสแตนซ์ที่มีอยู่แล้ว หรือทำการสร้างอินสแตนซ์ใหม่ เป็นต้น ส่วนตัวที่ตามคือ RemoteMachineName ใช้ในการติดต่อกับรีโมตเซิร์ฟเวอร์

เราจะสร้างแอปพลิเคชันของไคลเอนต์ขึ้นมาง่ายๆ โดยเรียกใช้ตัว TEventIntf2 วางบนฟอร์ม แล้ววางคอมโพเนนต์ต่างๆให้เป็น แอปพลิเคชันอย่างง่ายๆ ดังรูปที่ 7-3



รูปที่ 7-3 แสดงโปรแกรมทดสอบการเป็นไคลเอนต์ ของ คอมโพเนนต์ TEventIntf2

ภายในคอมโพเนนต์ TEventIntf2 จะมีอีเวนต์ให้เราใช้ได้คือ อีเวนต์ OnText ที่เราสร้างขึ้นทำการใช้ได้เหมือนกับการใช้อีเวนต์ของคอมโพเนนต์ทั่วไป โดยดับเบิลคลิกที่ชื่ออีเวนต์ OnText แล้วเขียนโค้ดดังนี้

```

Procedure TForm1.ButtonClick(Sender : TObject)
Begin
    EventIntf21.SenderText(Edit1.Text)
End;

Procedure TForm1.EventIntf2Event1Text(Sender : TObject)
Begin
    Memo1.Lines.Add(Text);
End;

```

การทำงานของไคลเอนต์เป็นการทำงานแบบง่ายๆ โดยเรียกเมธอด SendText ของเซิร์ฟเวอร์แล้วเซิร์ฟเวอร์จะทำการ Callback กลับมาที่ไคลเอนต์ทำให้เกิดการเขียนตัวอักษรขึ้นบนเมโม

### 7.3 การสร้างเซิร์ฟเวอร์แบบหลายไคลเอนต์

ตัวอย่างข้างต้นที่แสดงไปเป็นเซิร์ฟเวอร์ที่จัดการได้กับไคลเอนต์ได้ตัวเดียว ต่อไปเราจะดัดแปลงเซิร์ฟเวอร์ให้สามารถรองรับไคลเอนต์ให้เข้ามาใช้งานออบเจกต์ที่เซิร์ฟเวอร์ได้ทีละหลายๆ ตัวเป็นตัวอย่างการทำ Chat เซิร์ฟเวอร์อย่างง่าย

ขั้นตอนการแปลงทำดังนี้ แก้ไขเมธอด Initialize จาก ckSingle เป็น ckMulti และทำการเรียกเมธอด RegisterActiveXObject ดังนี้

```
Procedure TEventMultSrv.Initialize;
Begin
    Inherited Initialize;
    FconnectionPoints := TconnectionPoints.Create(Self);
    If AutoFactory.EventTypeInfo <> nil then
        FConnectionPoint := FconnectionPoints.CreateConnectionPoint(
            AutoFactory.EventIID, ckMulti, EventConnect )
    Else FConnectionPoint := nil

    RegisterActiveObject ( Self as IUnknown, CLASS_EventMultSrv,
        ACTIVEOBJECT_WEAK, FObjectID );
End;
```

เนื่องจากเซิร์ฟเวอร์ต้องจัดการกับไคลเอนต์หลายตัวจึงต้องรู้ว่ามีส่วนการติดต่อมายังเซิร์ฟเวอร์อย่างไรโดยการใช้เมธอด GetEnumerator และต้องมีการทำลายออบเจกต์เมื่อเลิกใช้ด้วยตัวทำลาย Destroy การเขียน โค้ดภายในเป็นดังนี้

```
Destructor TEventMultSrv.Destroy;
Begin
    RevokeActiveObject ( FObjectID, nil );
    Inherited Destroy;
End;

Function TEventMultSrv.GetEnumerator : IEnumConnections;
Var
    Container : IConnectionPointContainer;
    ConnectionPoint : IConnectionPoint;
Begin
    OleCheck(QueryInterface(IConnectionPointContainer, Container));
    OleCheck(Container.FindConnectionPoint(AutoFactory.EventIID, ConnectionPoint));
    ConnectionPoint.EnumConnections(Result);
End;
```

มาถึงเมธอด SendText ก็ต้องแก้ไขให้รองรับกับการ กระจายข้อความให้กับทุกๆ ไคลเอนต์ที่ต่อต่อกับออบเจกต์ของเซิร์ฟเวอร์ ดังนี้

```
Function TEventMultSrv.SendText ( const Text: WideString);
var
    Enum : IEnumConnections;
    ConnectData: TConnectData
```

```

    Fetched : Cardinal;
Begin
    Enum := GetEnumerator;
    If Enum <> nil then
    Begin
        While Enum.Next(1, ConnectData, @fetched) = s_ok do
            If ConnectData.pUnk <> nil then
                (ConnectData.pUnk as leventMultSrvEvents).OnText(Text);
        End;
    End;
End;

```

เป็นการเสร็จสิ้นการสร้าง Chat เซิร์ฟเวอร์อย่างง่าย ๆ ส่วนโคลเอนต์ไม่ต้องแก้ไขสามารถนำมาใช้กับเซิร์ฟเวอร์นี้ได้เลย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

### ไมโครเซอร์ฟิตรานแซกชันเซิร์ฟเวอร์

การพัฒนาระบบงานโดยใช้สถาปัตยกรรมเอ็นทีเยอร์ส(N-Tiers Architecture) เราจะให้คอมโพเนนต์แอปพลิเคชันต่าง ๆ ทำงานอยู่บนมิดเดิลเทียร์เซิร์ฟเวอร์ (Middle-Tier Server) ซึ่งเป็นอิสระจากส่วนงานพีริเซ็นเตชันและงานด้านฐานข้อมูล คอมโพเนนต์ที่เกี่ยวกับงานพีริเซ็นเตชันจะทำงานอยู่ที่เดสก์ท็อปไคลเอนต์หรือเว็บเบราว์เซอร์ ส่วนคอมโพเนนต์ที่เกี่ยวกับข้อมูลก็จะทำงานอยู่บนดาตาเบสเซิร์ฟเวอร์ ข้อดีในการแยกแอปพลิเคชันคอมโพเนนต์เป็นอิสระจากส่วนพีริเซ็นเตชันและงานฐานข้อมูลคือ

- สามารถนำแอปพลิเคชันคอมโพเนนต์กลับมาใช้ใหม่ (Reusable) ได้
- แอปพลิเคชันคอมโพเนนต์สามารถทำงานบนหลาย ๆ เครื่องพร้อมกันได้
- แอปพลิเคชันคอมโพเนนต์สามารถใช้การเชื่อมต่อดาตาเบส (Database Connection) ร่วมกันได้

#### 8.1 การพัฒนาระบบงานเอ็นทีเยอร์สโดยใช้ไมโครเซอร์ฟิตรานแซกชันเซิร์ฟเวอร์

เอ็มทีเอสเป็นแอปพลิเคชันที่ใช้ในการพัฒนาระบบงานเชิงคอมโพเนนต์ มีรันไทม์เอนไวรอนเมนต์ (Run-time environment) ใช้ควบคุมแอปพลิเคชันในการเรียกใช้คอมโพเนนต์ทั้งที่มีลักษณะเป็นทรานแซกชันหรือไม่ใช่ทรานแซกชันก็ตาม เอ็มทีเอสทำให้การพัฒนาและจัดการระบบงานเอ็นทีเยอร์สที่สร้างบนพื้นฐานของคอมโพเนนต์ ออบเจกต์โมเดลมีความสะดวกขึ้น แอปพลิเคชันคอมโพเนนต์จะทำงานภายใต้การควบคุมของ MTS คอมโพเนนต์เหล่านี้จะถูกเรียกใช้จากไคลเอนต์หลายประเภท เช่น เว็บเบราว์เซอร์ เอเอสพีสคริปต์ที่ทำงานอยู่ภายใน ไอไอเอส หรือแม้กระทั่งแอปพลิเคชันทั่วไปที่เขียนโดยวิซวลเบสิก เป็นต้น

#### 8.2 ข้อดีในการใช้เอ็มทีเอส

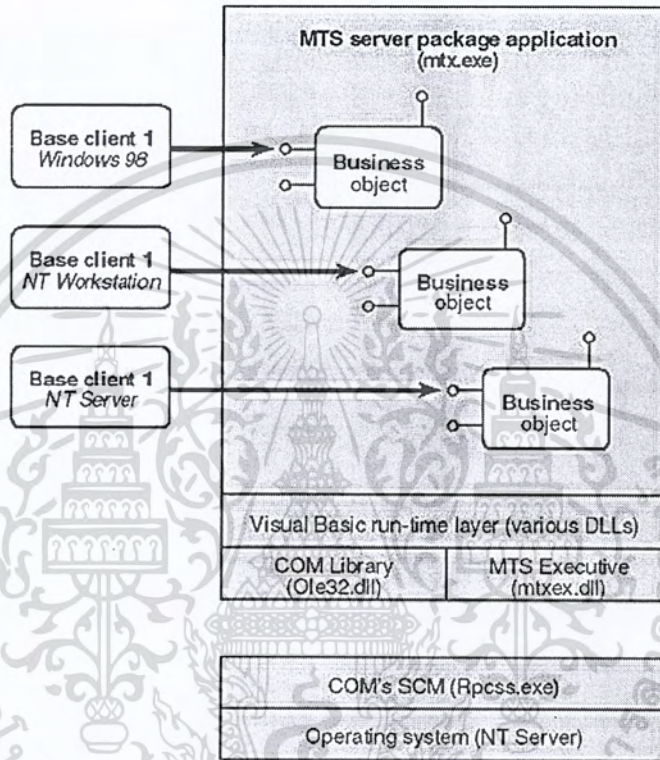
เอ็มทีเอสทำให้การพัฒนาระบบงานมีประสิทธิภาพ พึ่งกันในงานธุรกิจต่าง ๆ สามารถรองรับการใช้งานจากผู้ที่มีจำนวนมากได้ ช่วยลดความซ้ำซ้อนและทำให้ง่ายในการพัฒนาแอปพลิเคชัน โดยเราสามารถพัฒนาคอมโพเนนต์ประเภทอินโพรเซสทำงานบนมิดเดิลเทียร์เซิร์ฟเวอร์ภายใต้การควบคุมของเอ็มทีเอส เมื่อไคลเอนต์แอปพลิเคชันต้องการเรียกใช้คอมโพเนนต์ก็จะติดต่อกับเอ็มทีเอสโดยใช้ซีไอ-เอ็มซีไออื่น ๆ ในการใช้ MTS คือ

- สามารถติดต่อกับฐานข้อมูลและทรัพยากรอื่น ๆ เช่น เมนเฟรมแอปพลิเคชัน ในลักษณะที่เป็นทรานแซกชันได้
- จัดการคอมโพเนนต์ได้ง่าย สามารถใช้การแดรกแอนด์ดรอป (drag-and-drop) คอมโพเนนต์ผ่านทางเอ็มทีเอสเอ็กซ์พลอเรอร์ และมียูทิลิตี้ต่าง ๆ ที่ช่วยในการรีจิสเตอร์คอมโพเนนต์บนไคลเอนต์ได้โดยอัตโนมัติ

### 8.3 เอ็มทีเอสคอมโพเนนต์

เอ็มทีเอสมีรันไทม์เอนไวรอนเมนต์ที่ใช้รันออบเจกต์ รูปที่ 8-1 แสดงความสัมพันธ์ระหว่างไคลเอนต์ แอปพลิเคชันและออบเจกต์ที่รันอยู่ในรันไทม์เอนไวรอนเมนต์

ถึงแม้ว่าคอมโพเนนต์ประเภทอินโทรเซสสามารถใช้เป็น เอ็มทีเอสคอมโพเนนต์ได้ แต่จะไม่สามารถใช้เซอร์วิสต่าง ๆ ของเอ็มทีเอส ได้อย่างมีประสิทธิภาพ ดังนั้นเราจึงควรปรับปรุงคอมโพเนนต์ให้ใช้คุณสมบัติของเอ็มทีเอสอย่างเต็มที่ ซึ่งเอ็มทีเอสต้องการคอมโพเนนต์ที่มีลักษณะดังนี้



รูปที่ 8-1 ความสัมพันธ์ระหว่างไคลเอนต์แอปพลิเคชันและออบเจกต์ที่รันอยู่ในรันไทม์เอนไวรอนเมนต์

- คอมโพเนนต์จะต้องมีออบเจกต์คลาสมาตรฐานที่อิมพลีเมนต์มาจากอินเตอร์เฟซ IclassFactory
- เมธอด IclassFactory::CreateInstance ของออบเจกต์คลาสจะต้องรีเทิร์นอินสแตนซ์ของออบเจกต์ใหม่ทุกครั้งเมื่อเมธอดนี้ถูกเรียกใช้
- คอมโพเนนต์จะต้องเปิดเผยออบเจกต์คลาสโดยใช้ฟังก์ชัน DllGetClassObject
- คอมโพเนนต์ที่รันในเอ็มทีเอส ต้องไม่เอะกรึกเกิดกับคอมโพเนนต์ที่ไม่ได้รันในเอ็มทีเอส
- คอมโพเนนต์ต้องใช้ฟังก์ชัน DllRegisterServer ในการรีจิสเตอร์ ProgID, ClsID, อินเตอร์เฟซ และไทป์ไลบรารี ไว้ที่ HKEY\_CLASSES\_ROOT ของวินโดวส์รีจิสทรี
- อินเตอร์เฟซของคอกโพเนนต์ และ ซีไอเอ็มคลาสทั้งหมดจะต้องอธิบายอยู่ในไทป์ไลบรารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 8.4 บริการต่าง ๆ ของ MTS

#### คอมโพเนนต์ทรานแซกชัน (Component Transaction)

คอมโพเนนต์ทรานแซกชันมีลักษณะคล้ายกับดาตาเบสทรานแซกชัน คอมโพเนนต์ทรานแซกชันประกอบด้วยวิธีการในการรันคอมโพเนนต์ตั้งแต่หนึ่งคอมโพเนนต์ขึ้นไป เอ็มทีเอสจะควบคุมการทำงานของคอมโพเนนต์ในการติดต่อกับทรัพยากรที่ทำงานแบบทรานแซกชัน เช่น ติดต่อกับฐานข้อมูล หากคอมโพเนนต์ยกเลิกการทำทรานแซกชัน เอ็มทีเอสจะโรลแบค (Rollback) สิ่งที่คอมโพเนนต์ได้ปฏิบัติไป

#### ความปลอดภัย

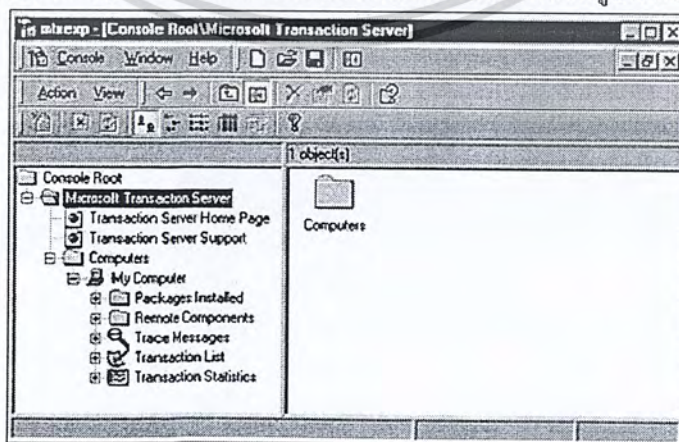
เอ็มทีเอสใช้ระบบการรักษาความปลอดภัยของวินโดวส์เอ็นทีในการตรวจสอบคุณสมบัติผู้ใช้ระบบ ส่วนเอ็มทีเอสเองมีโมเดลในการรักษาความปลอดภัย 2 แบบคือ Declarative Security และ Programmatic Security

#### การจัดการ (Administration)

เอ็มทีเอสมีเอ็มทีเอสเอ็กซ์พลอเรอร์ (MTS Explorer) เป็นยูสเซอร์อินเตอร์เฟซในการจัดการควบคุมแพคเกจ ปรับเปลี่ยนคุณสมบัติคอมโพเนนต์ในด้านความปลอดภัยและลักษณะการทำทรานแซกชัน ติดตามการทำงานของเซิร์ฟเวอร์ นอกจากนี้เอ็มทีเอสเอ็กซ์พลอเรอร์ยังมีคุณสมบัติอื่น ๆ ดังนี้

- ใช้งานง่าย สามารถใช้การแคชแคชคอมโพเนนต์จากภายนอกนำมาติดตั้งในเอ็นวีไวรอนเมนต์ของเอ็มทีเอสได้ทันที
- สามารถล็อกแพคเกจเพื่อป้องกันการเปลี่ยนแปลงแพคเกจขณะใช้งาน สามารถดูสถานะขณะทำงาน เปลี่ยนชื่อแพคเกจ และเพิ่มเงื่อนไขด้านความปลอดภัย
- มีความสามารถในการจัดการไอไอเอสแอปพลิเคชัน (IIS Application) ที่ทำงานภายใต้เอ็มทีเอส และจัดการแพคเกจที่ก่อสร้างโดยเอสเอ็นเอเชิร์ฟเวอร์ 4.0 (SNA Server 4.0)

เอ็มทีเอสเอ็กซ์พลอเรอร์เป็นส่วนหนึ่งของไมโครซอฟท์เมเนจเมนต์คอนโซล (Microsoft Management Console, MMC) มีลักษณะคล้ายกับวินโดวส์เอ็กซ์พลอเรอร์ ดังรูปที่ 8-2



รูปที่ 8-2 เอ็มทีเอสเอ็กซ์พลอเรอร์ซึ่งเป็นส่วนหนึ่งของไมโครซอฟท์เมเนจเมนต์คอนโซล

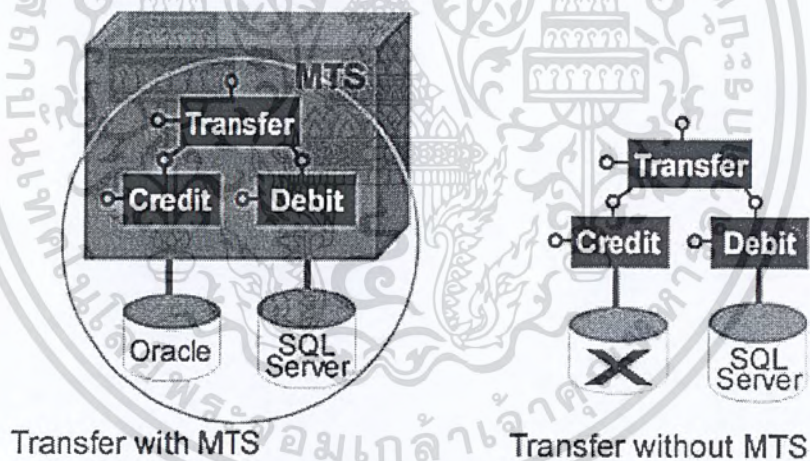
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังมีบริการอื่น ๆ เช่น การทำเทรดพูลลิ่งโดยอัตโนมัติ (Automatic Thread Pooling) ออบเจกต์โบรกเกอร์ (Object Broker) รีซอร์สพูลลิ่ง (Resource Pooling) โพรเซสไอโซเลชัน (Process Isolation) สนับสนุนคอมโพเนนต์เพจเจจ จัสอินไทม์แอคทีเวชัน (Just-in-Time Activation) เป็นต้น

### 8.5 ตัวอย่างของเอ็มทีเอสแอปพลิเคชัน

แอปพลิเคชันระบบธนาคารประกอบด้วย 3 คอมโพเนนต์ได้แก่ การฝาก การถอน และการโอนเงิน ซึ่งคอมโพเนนต์แต่ละอันถูกพัฒนาโดยนักพัฒนาต่างกัน และการโอนเงินทำงานโดยเรียกใช้คอมโพเนนต์การถอนและการฝาก

สมมติว่าการโอนเงิน เริ่มต้นจะต้องถอนเงินจากบัญชีก่อน แล้วจึงทำการฝาก หากการฝากเกิดปัญหาไม่สามารถทำงานได้ หากไม่ได้ทำงานแบบทรานแซกชันผู้พัฒนาคอมโพเนนต์การโอนเงินจะต้องเขียนโปรแกรมจัดการปัญหาให้บัญชีมียอดเงินเท่าเดิม ในกรณีตัวอย่างนี้การจัดการแก้ปัญหายังสามารถจัดการได้ง่ายเนื่องจากมีคอมโพเนนต์ในระบบไม่มากนัก หากระบบงานประกอบด้วยคอมโพเนนต์จำนวนมากที่ทำงานสัมพันธ์กัน หากการทำงานของคอมโพเนนต์ใดมีปัญหา การจัดการแก้ไขจะทำให้ยากและสิ้นเปลืองค่าใช้จ่าย รูปที่ 8-3 แสดงให้เห็นถึงการโอนเงินที่ทำงานในลักษณะเป็นทรานแซกชัน (โดยใช้เอ็มทีเอส) และไม่ได้เป็นทรานแซกชัน (ไม่ใช่เอ็มทีเอส)



รูปที่ 8-3 กระบวนการโอนเงินโดยใช้เอ็มทีเอสและไม่ใช้เอ็มทีเอส

เอ็มทีเอสจะควบคุมการทำทรานแซกชันให้โดยผู้พัฒนาโปรแกรมไม่จำเป็นต้องเขียน Begin Transaction และ End Transaction ในโค้ดของโปรแกรมเลย ผู้พัฒนาโปรแกรมสามารถใช้เครื่องมือและภาษาต่าง ๆ สร้างคอมโพเนนต์และให้ทำงานเป็นแบบทรานแซกชันได้โดยใช้เอ็มทีเอสควบคุม เมื่อคอมโพเนนต์เริ่มทำงาน เอ็มทีเอสก็จะเริ่มต้นทรานแซกชันให้หากคอมโพเนนต์ต้องการ

เมื่อคอมโพเนนต์มีการติดต่อทรัพยากรต่าง ๆ เช่น ฐานข้อมูลหรือเมนเฟรมแอปพลิเคชัน เอ็มทีเอสจะทำงานเป็นแบบทรานแซกชันโดยอัตโนมัติ หากคอมโพเนนต์ที่ทำงานแบบทรานแซกชันมีการเรียกใช้คอมโพเนนต์อื่น ๆ คอมโพเนนต์ที่ถูกเรียกใช้ก็จะอยู่ในทรานแซกชันเช่นกัน และ เอ็มทีเอสทำการควบคุมทรานแซกชันแบบทูเฟส-กอมมิต (Two-phase commit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปแล้วเอ็มทีเอสมีความสามารถหลักที่สำคัญ 3 ประการคือ

1. สนับสนุนการทำทรานแซกชัน
2. สนับสนุนการใช้ข้อมูลร่วมกัน
3. สนับสนุนด้านความปลอดภัย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 9 วินโดวส์ดีเอ็นเอ

### 9.1 ความหมาย

วินโดวส์ดีเอ็นเอ (Windows DNA) มาจากคำว่า Windows Distributed interNet Architecture เป็นแนวทางในการสร้างระบบงานของไมโครซอฟต์ หรือเป็นข้อเสนอในการสร้างระบบงานซึ่งเชื่อมต่อกับอินเทอร์เน็ต วินโดวส์ดีเอ็นเอไม่ใช่ข้อกำหนดคั้งนั้นในการพัฒนาระบบงานใด ๆ จึงอาจจะใช้เพียงบางส่วนของวินโดวส์ดีเอ็นเอ เพื่อพัฒนาระบบงานร่วมกับองค์ประกอบอื่น ด้วยก็ได้

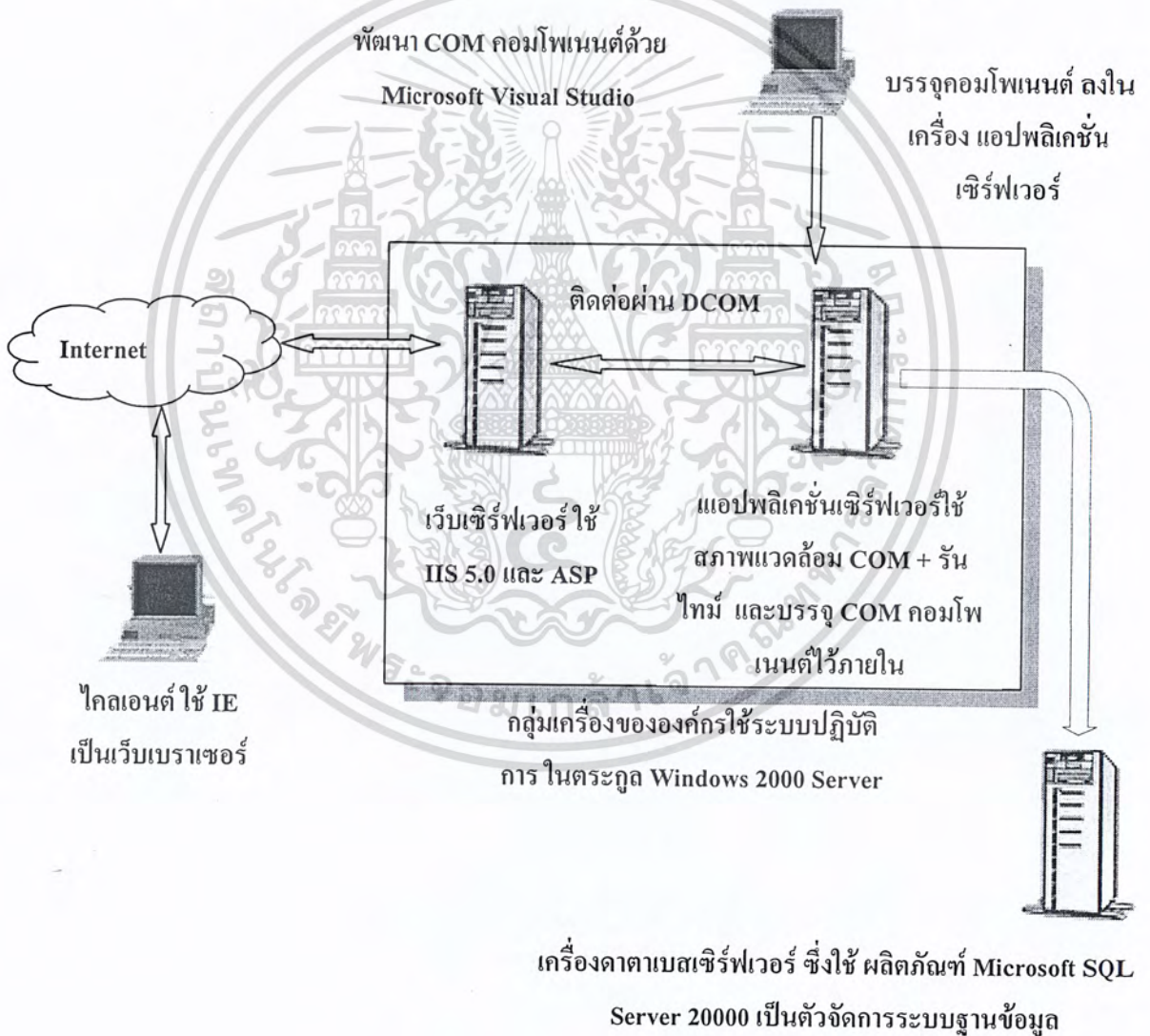
### 9.2 องค์ประกอบของวินโดวส์ดีเอ็นเอ

องค์ประกอบของวินโดวส์ดีเอ็นเอประกอบด้วย

- ส่วนซอฟต์แวร์คอมพิวเตอร์ ในส่วนนี้ทางไมโครซอฟต์นั้นเสนอให้ใช้คอมพิวเตอร์เน็ต เพื่อให้สามารถใช้บริการต่าง ๆ ของ สภาพแวดล้อมคอมพิวเตอร์ใหม่เช่นมี เซอร์เวอริสเป็นลักษณะ Both ทำให้สามารถเพิ่มประสิทธิภาพในการประมวลผล (Scalability) เป็นต้น
- มีแอปพลิเคชันเซิร์ฟเวอร์ หรือมีคอนเทนเนอร์ (Container) ของคอมพิวเตอร์เน็ตเป็น สภาพแวดล้อม คอมพิวเตอร์ใหม่มีมาพร้อมกับระบบปฏิบัติการ Windows 2000 ฎ Server, Windows 2000 ฎ Advance Server และ Windows 2000 ฎ Data Center เนื่องจากตัวสภาพแวดล้อมคอมพิวเตอร์นั้น สนับสนุนตัวคอมพิวเตอร์ซึ่งมีลักษณะเหมาะสมกับระบบสภาพแวดล้อมนี้ นอกจากนั้นยังมีบริการเพิ่มเติม เช่น การจัดการทรานแซกชัน หรือระบบความปลอดภัยแบบแบ่งลำดับชั้น (Fine-Gain Security) เป็นต้น
- มีเว็บเซิร์ฟเวอร์ เป็น Internet Information Server (IIS) เวอร์ชัน 5.0 ซึ่งมาพร้อมกับผลิตภัณฑ์ Microsoft Windows 2000 Server, Microsoft Windows 2000 ฎ Advance Server และ Microsoft Windows 2000 ฎ Data Center
- เครื่องมือในการพัฒนาคอมพิวเตอร์เน็ตนั้นทางไมโครซอฟต์ได้เสนอผลิตภัณฑ์ Microsoft Visual Studio ฎ ซึ่งประกอบด้วยโปรแกรม Visual Basic, Visual C++ และ Visual InterDev อยู่นอกจากนั้นยังมีโปรแกรมและเครื่องมือต่าง ๆ ที่จะช่วยให้การพัฒนาคอมพิวเตอร์เน็ตสะดวกยิ่งขึ้น และไม่เกิดปัญหาจากความเข้ากันได้ของคอมพิวเตอร์เน็ตและส่วนประกอบอื่น ๆ ของระบบ
- สำหรับภาษาสคริปต์ในฝั่งเซิร์ฟเวอร์ ไมโครซอฟต์เสนอ Active Server Page (ASP) ซึ่งใช้งานได้ไม่ยากนักเนื่องจากมีความคล้ายภาษา VBScript ซึ่งก็มีลักษณะและไวยากรณ์ของภาษาไม่ต่างจาก Visual Basic ซึ่งอยู่ในชุดผลิตภัณฑ์ Microsoft Visual Studio ฎ เท่าใดนัก เพื่อใช้ในการเรียกตัวคอมพิวเตอร์เน็ตซึ่งอยู่ภายในตัว แอปพลิเคชันเซิร์ฟเวอร์ นอกจากนี้ยังสามารถพัฒนาได้ด้วยโปรแกรม Microsoft Visual InterDev ซึ่งอยู่ในชุดผลิตภัณฑ์ Microsoft Visual Studio อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในส่วนการจัดการฐานข้อมูลนั้นก็จะเป็นตัวผลิตภัณฑ์ Microsoft SQL Server 2000 ซึ่งเข้ากันได้กับคอมพิวเตอร์ และยังมีคุณสมบัติใหม่ ๆ ซึ่งได้เพิ่มมาจาก เวอร์ชันก่อนหน้า คือตัวผลิตภัณฑ์ Microsoft SQL Server 7.0 อยู่พอสมควร และใช้งานได้ง่ายกว่าผลิตภัณฑ์ ระบบจัดการบริหาร ฐานข้อมูลตัวอื่นอยู่พอสมควร
  - ในการเชื่อมต่อและส่งผ่านข้อมูล นั้นก็จะใช้ดีค้อม เป็นตัวกลางในการติดต่อกับคอมพิวเตอร์ และ ส่วน ต่าง ๆ ภายในระบบ
  - ตัวเว็บเบราว์เซอร์ ที่อยู่ใน DNA นั้นก็จะเป็น Internet Explorer เนื่องจากมีความเข้ากันได้กับคอมพิวเตอร์ นอกจากนั้นตัว Internet Explorer นั้นก็มีเวอร์ชันที่ทำงานอยู่ในหลาย ๆ แพลตฟอร์ม และยังเป็นผลิตภัณฑ์ ที่แถมมากับระบบปฏิบัติการ Microsoft Windows เวอร์ชันใหม่ ๆ อยู่แล้ว
- จากที่ได้กล่าวมาเบื้องต้น ภาพรวมของระบบ Windows DNA จะมีลักษณะดังรูปที่ 9-1



รูปที่ 9-1 สถาปัตยกรรมของวินโดวส์ดีเอ็นเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 10

### สภาพแวดล้อม COM+ รันไทม์ และบริการของ COM+

เป็นกลุ่มของบริการจากระบบ ที่มีบริการต่าง ๆ สำหรับ COM ออบเจ็กต์ ดังนี้ fine-grained security, distributed transactions, thread synchronization, load balancing, asynchronous store และ forward method invocation นอกจากนี้แล้วยังมีการพัฒนาการขยายขนาด (scalability) โดยใช้ Just In Time (JIT) Activation และ การ pooling ของ ออบเจ็กต์ และ database connection

#### 10.1 สถาปัตยกรรม ของ COM+

##### 10.1.1 Attributed-Based Programming

หลักการของวิธีนี้คือ การที่สามารถทำการปรับแต่งคุณสมบัติของ COM ออบเจ็กต์ได้จากการปรับแต่งค่าของแอททริบิวต์ โดยที่รายละเอียดคนออกจากรันไทม์ COM+ รันไทม์ จะเป็นส่วนที่จัดการเพิ่มเติมเอง สำหรับในวินโดวส์ 2000 นั้นค่าแอททริบิวต์สำหรับ COM ออบเจ็กต์ต่างนั้นไม่ได้เก็บอยู่ในรีจิสทรีของระบบ แต่จะถูกเก็บไว้ใน configuration database ซึ่งเรียกว่า COM+ catalog นอกจากนั้นแล้ว Windows 2000 นั้นยังได้เตรียม Catalog Manager ไว้เพื่อความสะดวกในการใช้งานเรียกว่า Component Service Explorer

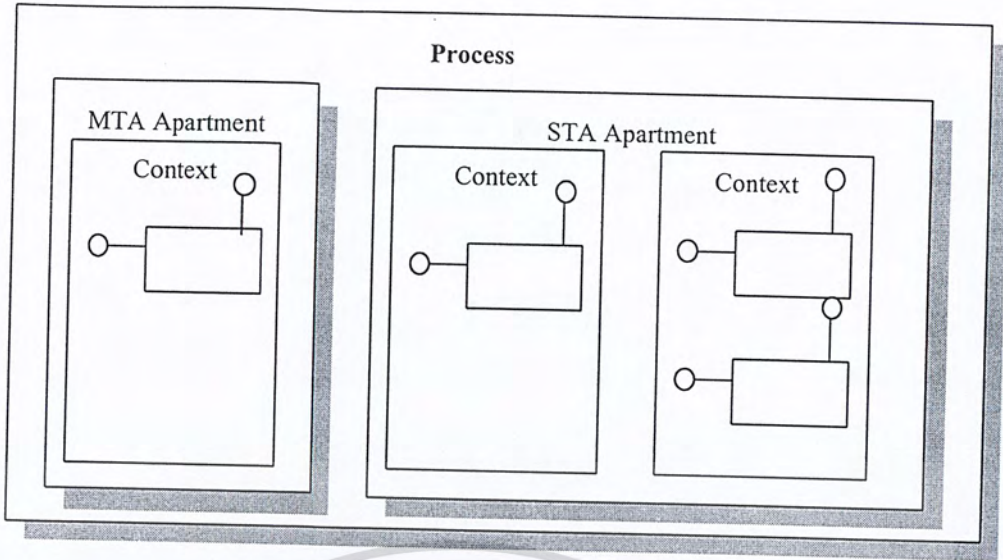
การใช้งาน Component Service Explorer สามารถเรียกใช้ได้โดย

1. กดปุ่ม Start
2. เลือก Programs \ Administrative Tools \ Component Services

##### 10.1.2 คอนเทก (Context)

COM+ คอนเทก คือ สภาพแวดล้อมรันไทม์ ซึ่ง COM+ ออบเจ็กต์ ที่มีความเข้ากันได้กับ โพรเซสที่เหมาะสมทำการประมวลผล (สำหรับความเข้ากันได้นั้นหมายถึง ออบเจ็กต์ที่ใช้ runtime requirement ร่วมกัน; หากมีการปรับแต่งค่า สำหรับ ออบเจ็กต์ตัวใดที่แตกต่างกัน ก็ต้องอยู่ต่าง คอนเทก กัน)

Context จะอยู่ภายใน อพาร์ทเมนต์ (Apartment) ซึ่งจะอยู่ภายใน โพรเซส อีกที เฉพาะ เซรต ที่อยู่ใน Apartment นั้นเท่านั้นที่จะสามารถเรียกใช้โดยผ่าน คอนเทก ได้โดยตรง (นอกจากจะเป็น thread-neutral apartment) ในแต่ละ อพาร์ทเมนต์ นั้นสามารถมีได้หลาย คอนเทก และ ภายในแต่ละ คอนเทก ก็มีได้หลายออบเจ็กต์ ดังรูปที่ 10-1



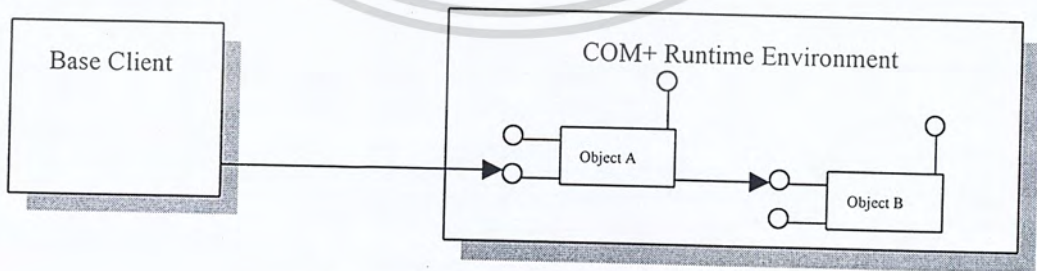
รูปที่ 10-1 แสดงโพรเซส อพาร์ทเมนต์ และ คอนเทก

10.1.3 COM+ ไคลเอนต์ กับ Context

COM+ ออบเจกต์สามารถมีไคลเอนต์ได้ 2 ประเภทคือ 1) COM+ ออบเจกต์ตัวอื่นแต่อยู่ในสภาพแวดล้อม COM+ runtime หรือ 2) ไคลเอนต์ซึ่งรันอยู่นอกสภาพแวดล้อม COM+ runtime ดังรูปที่ 10-2

รูปแบบแรกนั้นจะมีลักษณะเดียวกับ ออบเจกต์ A ในรูปที่ x ซึ่งเรียกใช้ออบเจกต์ B และอยู่ในสภาพแวดล้อม COM+ รันไทม์ ส่วนในรูปแบบหลังนั้นเรียกว่า Base Client ซึ่งเป็นส่วนของซอฟต์แวร์ที่เรียกใช้ COM+ ออบเจกต์แต่ตัวมันจะอยู่นอกสภาพแวดล้อม COM+ รันไทม์ โดยมากแล้วไคลเอนต์ประเภทนี้มักจะเป็นพวก User Interface แต่ก็ไม่ใช่เสมอไป

ส่วนใหญ่แล้วตัวไคลเอนต์กับตัว COM+ ออบเจกต์มักจะอยู่ต่าง Process หรือ อย่างน้อยก็จะอยู่ต่าง Context การติดต่อสื่อสารระหว่างส่วนต่างๆจึงจำเป็นต้องมีกระบวนการที่ใช้จัดการนี้ซึ่งเรียกว่า Interception (ซึ่งจะได้อธิบายต่อไป) อย่างไรก็ตามในบางกรณีเราจะออกแบบให้ตัวไคลเอนต์กับตัว COM+ ออบเจกต์อยู่ใน Context เดียวกันก็ได้

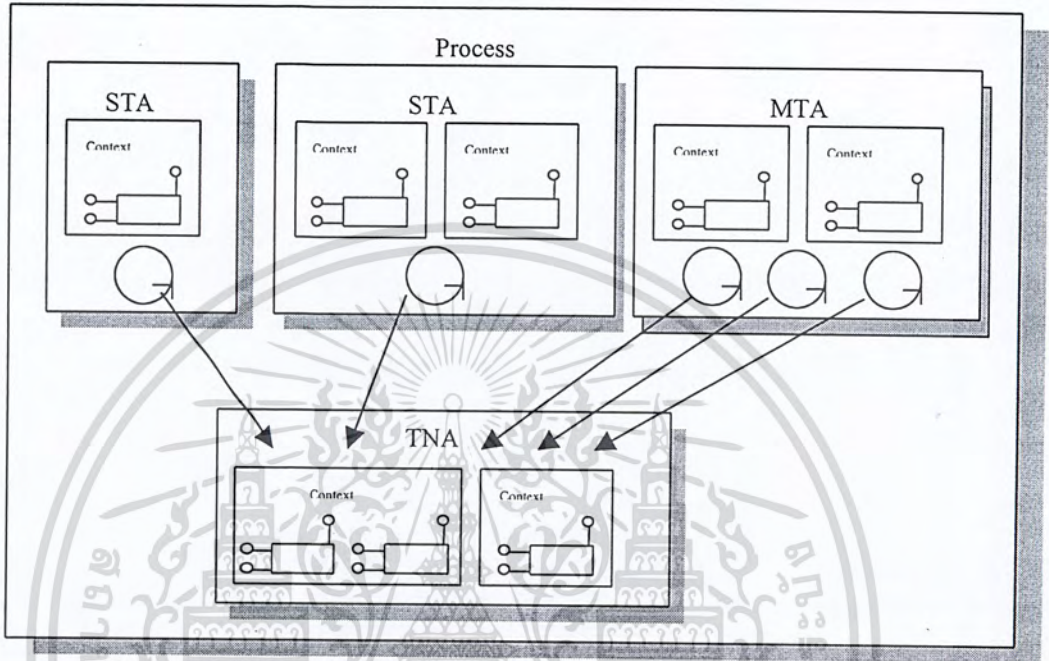


รูปที่ 10-2 แสดงCOM+ ไคลเอนต์ทั้งสองประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 10.1.4 คอนเทก และ อพาร์ตเมนต์

ในมุมมองของ COM+ อพาร์ตเมนต์ จะเป็นกลุ่มของ คอนเทก ซึ่งอยู่ใน โพรเซส ดังรูปที่ 10-3 Context ทำหน้าที่เป็นสภาพแวดล้อมที่ใช้ในการประมวลผลขั้นในสุดแทน อพาร์ตเมนต์ โดยที่ อพาร์ตเมนต์ จะทำหน้าที่พิจารณาว่า Thread ใดมีสิทธิที่จะเข้าไป ติดต่อกับออบเจกต์ได้โดยตรงใน คอนเทก ที่เหมาะสม



รูปที่ 10-3 แสดงอพาร์ตเมนต์ และ คอนเทก

คอนเทก สามารถควบคุม การทำเรดซิงโครไนซ์สำหรับ COM+ ออบเจกต์ซึ่งการให้บริการนี้เป็นของ สภาพแวดล้อม COM+ รันไทม์

วิธีการทำ การทำเรดซิงโครไนซ์ที่ COM+ ให้บริการโดย COM+ รันไทม์ ทำโดย เมื่อมี เรดใด เรด หนึ่ง อ้างถึงเมธอดที่อยู่ในออบเจกต์ ทุกๆ เรด ที่เหลือจะถูกบล็อกไว้จนกว่า เรด แรกจะออกจากเมธอดและยกเลิกการล็อกซึ่งวิธีการนี้ใช้ได้อย่างมีประสิทธิภาพกับ Object Pooling ซึ่งกำหนดให้ออบเจกต์นั้นสามารถถูกเรียกใช้จาก Thread ใดๆ ก็ได้ อย่างไรก็ตาม การทำเรดซิงโครไนซ์ นั้นไม่สามารถใช้งานได้ ในระบบงานที่เป็น Thread-affinity เนื่องจากระบบงานแบบนี้จำเป็นต้องจำเพาะเจาะจงไปเลยว่าต้องการ เรด ใน เรด หนึ่งอย่างเดียว

สำหรับ STA นั้นไม่จำเป็นต้องมีการล็อกเนื่องจากมีแค่ Thread เดียวในตัวมันแต่อย่างไรก็ตาม STA นั้นไม่เหมาะกับ Object Pooling สรุปก็คือ คอนเทก จะใช้ในการควบคุมว่า เรด ต่างๆ จะสามารถเรียกใช้ออบเจกต์ได้ เมื่อไร ส่วน Apartment นั้นจะใช้ในการควบคุมว่า เรด ไหน ที่จะถูกอนุญาตให้เข้ามาเรียกใช้ออบเจกต์ นอกจากนั้นยังมีข้อแนะนำใหม่อีก 2 ข้อคือ

- COM+ ออบเจกต์ควรจะใช้ TNA และใช้ การทำเรดซิงโครไนซ์ซึ่งให้บริการโดย COM+
- COM+ ออบเจกต์ที่เป็น UI หรือที่เป็น Thread-affinity ควรใช้ STA ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้สามารถเข้ากันได้กับระบบเก่า ๆ COM อพาร์ทเมนต์ ก็ยังรวมถึง คอนเทก ที่เป็น ดิฟอลต์ ซึ่งจะไม่มีบริการต่างของ COM+

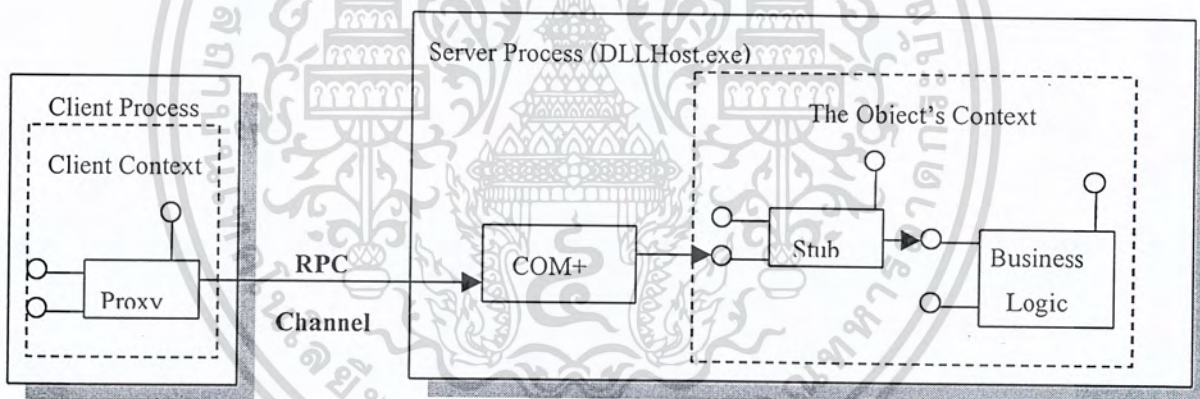
การทำเรดซิงโครไนซ์ เป็นทางเลือกหนึ่ง ซึ่งสามารถเลือกได้ว่าจะใช้บริการนี้หรือไม่

### 10.1.5 Interception

ในขณะที่ ไคลเอนต์เรียกใช้ COM ออบเจกต์ ถ้าตัวไคลเอนต์นั้นอยู่ใน คอนเทก เดียวกับ COM ออบเจกต์ ก็จะสามารถเรียกใช้ได้โดยตรงเนื่องจากตัวไคลเอนต์นั้นอยู่ในสภาพแวดล้อมแบบเดียวกับที่ตัว COM ออบเจกต์ต้องการ แต่ในการใช้งานจริงนั้นส่วนใหญ่แล้วตัวไคลเอนต์ และ COM ออบเจกต์มักจะ อยู่ต่าง คอนเทก กันดังนั้นจึงต้องมีตัวกลางในการจัดการ ซึ่งตัวกลางนี้เรียกว่า Interceptor ซึ่งมีลักษณะ เป็น พร็อกซี่ ที่มีขนาดเล็ก

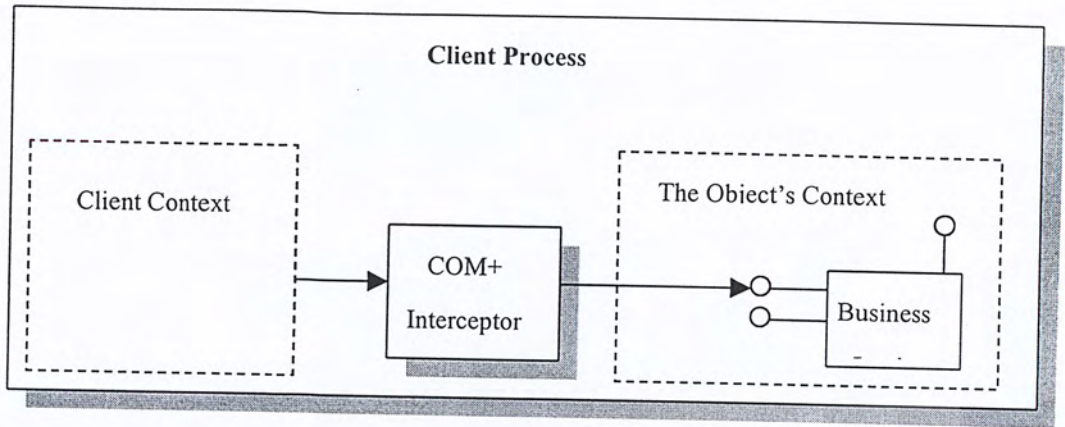
Interceptor มีหน้าที่ทำต่างๆ ซึ่งจำเป็นต่อการส่งคำสั่งจากไคลเอนต์ไปยังตัว COM ออบเจกต์ เช่นถ้าหากตัวออบเจกต์ ต้องการทราบเซชชันนั้นก็คือ ทราบเซชชัน ต้องถูกเริ่มต้น และจะบังคับให้มีการ ควบคุมการเข้าถึงแบบ fine-grained โดยการตรวจสอบว่าผู้ใช้นั้นมีสิทธิ์ที่จะใช้บริการนั้นๆ หรือไม่

หากตัวไคลเอนต์ และ COM ออบเจกต์นั้นอยู่คนละ โพรเซสตัว COM+ Interceptor ก็จะอยู่ ระหว่าง RPC Channel กับ Marshalling Stub ดังรูปที่ 10-4



รูปที่ 10-4 แสดง Interception ที่ใช้กับ COM+ ออบเจกต์ชนิด Out-of-Process

แต่ถ้าตัวไคลเอนต์ และตัว COM ออบเจกต์อยู่ใน โพรเซสเดียวกัน ตัว COM+ Interceptor ก็จะอยู่ ระหว่างตัวไคลเอนต์ และตัวออบเจกต์ ดังรูปที่ 10-5



รูปที่ 10-5 แสดง Interception ที่ใช้กับ COM+ ออบเจกต์ชนิด In-Process

การที่มีตัว Interceptor อยู่ระหว่างการเรียกใช้ของไคลเอนต์นั้นทำให้เกิดข้อดีที่เราสามารถที่จะควบคุมการทำงาน และการเรียกใช้งาน ระหว่างตัวไคลเอนต์และตัวออบเจกต์ได้โดยที่การทำงานนั้นจะเกิดขึ้นที่ตัว Interceptor ซึ่งวิธีการนี้จะเรียกว่า Interception และ COM+ จะนำไปใช้เพื่อ เริ่มต้นทรานแซกชัน, บังคับให้เกิด fine-grained security ฯลฯ ซึ่งเป็นการให้บริการต่าง ๆ ของ COM+ Runtime Environment โดยที่ COM ออบเจกต์นั้นไม่จำเป็นต้องถูกคอมไพล์ใหม่ หรือต้องเพิ่มโค้ด เพื่อให้สามารถใช้ได้ในระบบ COM+ แต่อย่างใด

## 10.2 การปรับแต่ง ออบเจกต์เพื่อใช้กับ สภาพแวดล้อม COM+ ใหม่

หลังจากที่ได้ศึกษาถึงลักษณะ และการให้บริการต่าง ๆ ที่ COM+ แล้วก็เป็นส่วนของการนำ COM ออบเจกต์ที่ได้พัฒนาไว้แล้วมาปรับแต่งเพื่อให้สามารถใช้กับ COM+ ได้อย่างมีประสิทธิภาพ ดังที่ได้กล่าวไว้แล้วว่า การปรับแต่งคุณสมบัติของ COM ออบเจกต์นั้นสามารถทำได้โดยโปรแกรม Component Service Explorer คุณสมบัติส่วนใหญ่จะแสดงผลในระดับคลาส แต่ในบางคุณสมบัติ ที่เป็น การแบ่งระดับในการรักษาความปลอดภัย (Fine - Grained Security ) ก็สามารถจัดการในระดับย่อยลงไป เป็นระดับ อินเตอร์เฟส หรือเมธอดได้

### ข้อจำกัด

- 1) COM ออบเจกต์ทุกตัวที่จะใช้ COM+ นั้นจะต้องเป็น In-Process Server
- 2) COM+ ออบเจกต์ควรใช้ TNA ออบเจกต์ที่ใช้ STA และ MTA สามารถทำงานได้แต่ก็มีข้อจำกัด เช่น ออบเจกต์ที่เป็น STA และ Primary STA นั้นจะไม่สามารถใช้ใน Object Pooling ได้
- 3) คอมโพเนนต์ทั้งหลายนั้นจะต้องใช้ การมาร์แชล แบบมาตรฐาน (ไม่สามารถใช้กับ การมาร์แชล แบบกำหนดเอง (Custom Marshaling) )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 10.3 COM+ แอปพลิเคชัน

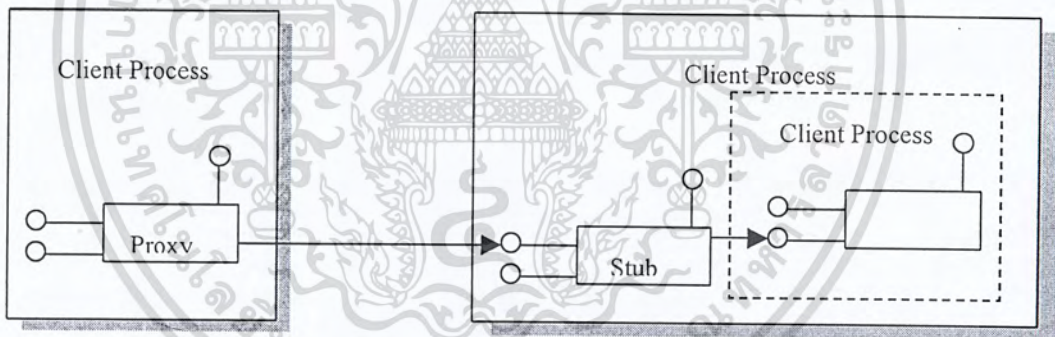
เป็นกลุ่มของ COM+ คอมโพเนนต์ซึ่งถูกจัดการเหมือนกับเป็น กลุ่มหนึ่งกลุ่ม และจำทำการรันในโพรเซส เดียวกันทั้งหมด แอปพลิเคชัน หนึ่ง ๆ นั้นสามารถจะมีได้หลายเซิร์ฟเวอร์ (ไฟล์ DLL) และในการปรับแต่ง COM+ คอมโพเนนต์นั้นจำเป็นต้องมีการสร้าง COM+ Application ขึ้นมาเสียก่อน

COM+ แอปพลิเคชัน นั้นมาแทนที่ MTS Package โดย COM+ แอปพลิเคชัน จะมีสองประเภทคือ เซิร์ฟเวอร์แอปพลิเคชัน และ ไลบรารีแอปพลิเคชัน

#### 10.3.1 เซิร์ฟเวอร์แอปพลิเคชัน ( Server Application )

แอปพลิเคชัน ประเภทนี้จะ รันอยู่ในโพรเซสของมัน ซึ่งในความเป็นจริงแล้วมันจะรันอยู่ใน Surrogate Process ซึ่ง COM+ เป็นผู้ให้บริการนี้ Surrogate Process จะเป็นอินสแตนซ์ของ DLLHost.Exe ซึ่งไฟล์นี้จะอยู่ที่ \WinNT\System32\

Surrogate Process คือวิธีการที่ทำให้ In-Process Server รันอยู่ในโพรเซสเฉพาะของมันซึ่งอยู่แยกกับโพรเซสของไคลเอนต์โดยการฝังตัว In-Process ที่ทำหน้าที่เป็น Proxy และ Stub ไว้ระหว่าง เซิร์ฟเวอร์และไคลเอนต์ ดังรูปที่ 10-6 ส่วนมากแล้วเรามักจะตั้งให้ แอปพลิเคชัน เป็น เซิร์ฟเวอร์แอปพลิเคชัน และเป็นค่าดีฟอลต์ของ COM+ ด้วย



รูปที่ 10-6 แสดง Surrogate Process

#### 10.3.2 ไลบรารีแอปพลิเคชัน ( Library Application )

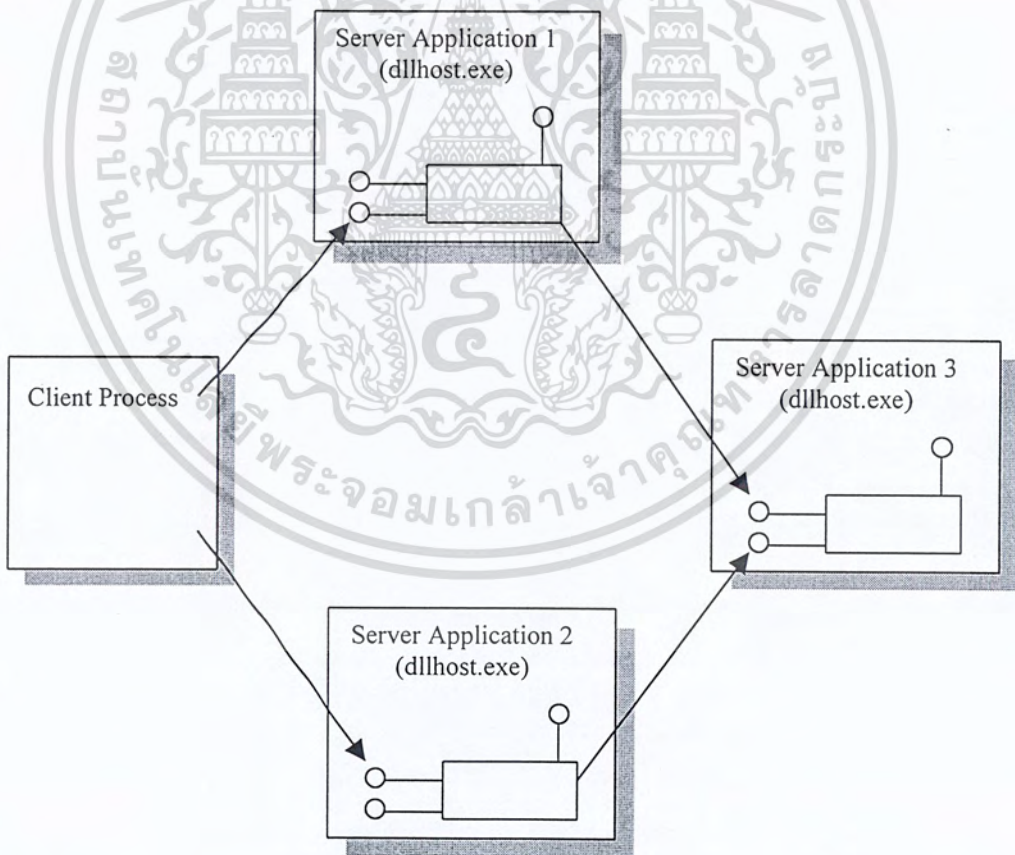
แอปพลิเคชัน ประเภทนี้จะอยู่ในโพรเซส เดียวกับไคลเอนต์ของมัน จุดประสงค์หลักในการเลือกใช้ แอปพลิเคชัน ประเภทนี้ก็คือสำหรับ COM+ ออบเจกต์ ซึ่งถูกเรียกใช้ จาก COM+ ออบเจกต์ ด้วยกันหลายๆตัว โดยที่ COM+ ออบเจกต์ที่เรียกใช้นั้นอยู่ต่างโพรเซสกันการแบ่งกันใช้ COM+ ออบเจกต์ดังรูปที่ 10-7 แต่ถ้าเรานำตัว COM+ ออบเจกต์ไปไว้ในโพรเซสเดียวกับ ออบเจกต์ที่เรียกใช้จะมีประสิทธิภาพสูงกว่า เนื่องจากรันอยู่ในโพรเซสเดียวกัน ซึ่งจะมีลักษณะดังรูปที่ 10-8

ในทางปฏิบัติเราอาจจะให้ Base Client ติดต่อกับ ไลบรารีแอปพลิเคชัน ได้ก็จริงแต่หากมองถึงประสิทธิภาพแล้วควรจะหลีกเลี่ยงวิธีนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

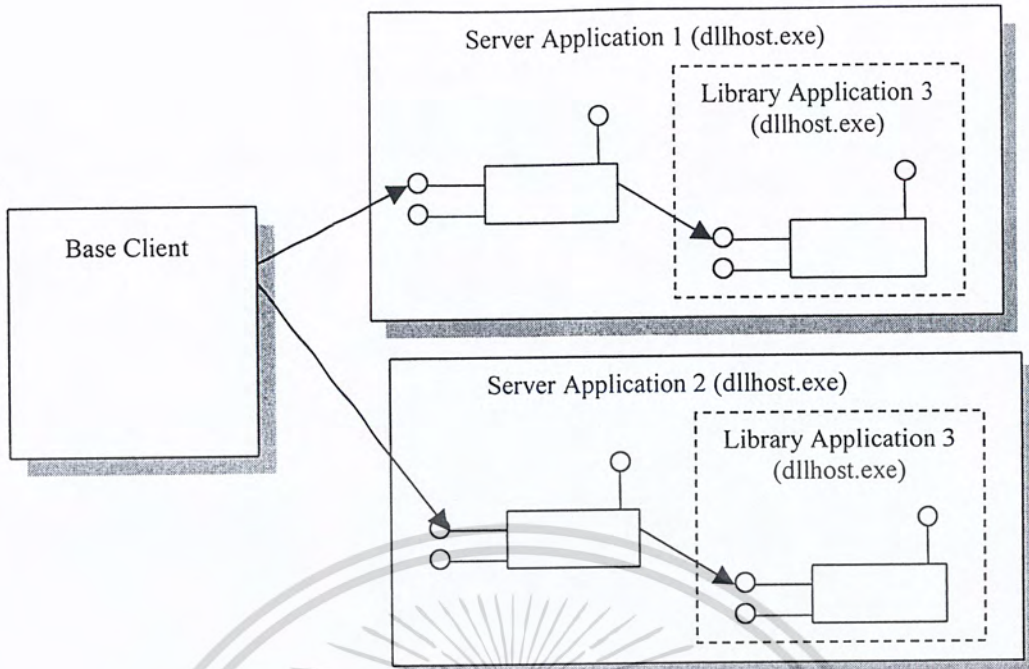
หลักการในการพิจารณาว่าควรจะจัดการกับการแบ่งกลุ่มออบเจกต์ต่าง ๆ ว่าจะอยู่ที่แอปพลิเคชัน ไหนอย่างไรนั้นมีรายละเอียดดังนี้

- ส่วนใหญ่แล้วควรมีแค่ แอปพลิเคชัน เดียวซึ่งรวมคอมโพเนนต์ที่มีทั้งไว้ภายใน
- คอมโพเนนต์ที่อยู่ต่าง แอปพลิเคชัน กันสามารถรันอยู่ในโพรเซสเดียวกันโดยใช้ COM+ ไลบรารีแอปพลิเคชันคอมโพเนนต์ที่อยู่ใน แอปพลิเคชัน กันจะรันที่โพรเซสเดียวกันตลอด
- คอมโพเนนต์ที่มีความสัมพันธ์กันมาก ควรจะอยู่ใน แอปพลิเคชัน เดียวกันเพื่อประสิทธิภาพที่ดีกว่า
- สำหรับ COM+ ไลบรารีแอปพลิเคชัน นั้นเหมาะกับงานที่ต้องการให้ แอปพลิเคชัน อยู่ในโพรเซสเดียวกับไคลเอนต์ของมันเพื่อประสิทธิภาพ
- คอมโพเนนต์ที่อยู่ใน แอปพลิเคชัน เดียวกันจะถูกจัดการด้วยกัน เราสามารถที่จะจัดระดับการรักษาความปลอดภัยของออบเจกต์ต่าง ๆ ให้อยู่ในระดับ กลาส หรือ เมรอด ก็ได้ แต่ผลของการจัดระดับนี้จะส่งผลกระทบต่อทุก ๆ ออบเจกต์ใน แอปพลิเคชัน นั้นดั่งนั้น หากมีความจำเป็นที่จะต้องจัดระดับต่างกันให้ออบเจกต์คนละตัว ควรแยกออบเจกต์ที่จัดระดับแตกต่างกันนั้นให้อยู่คนละ แอปพลิเคชัน กัน



รูปที่ 10-7 แสดง COM+ Server Application 2 ตัว ใช้ Server Application ตัวที่สามร่วมกัน

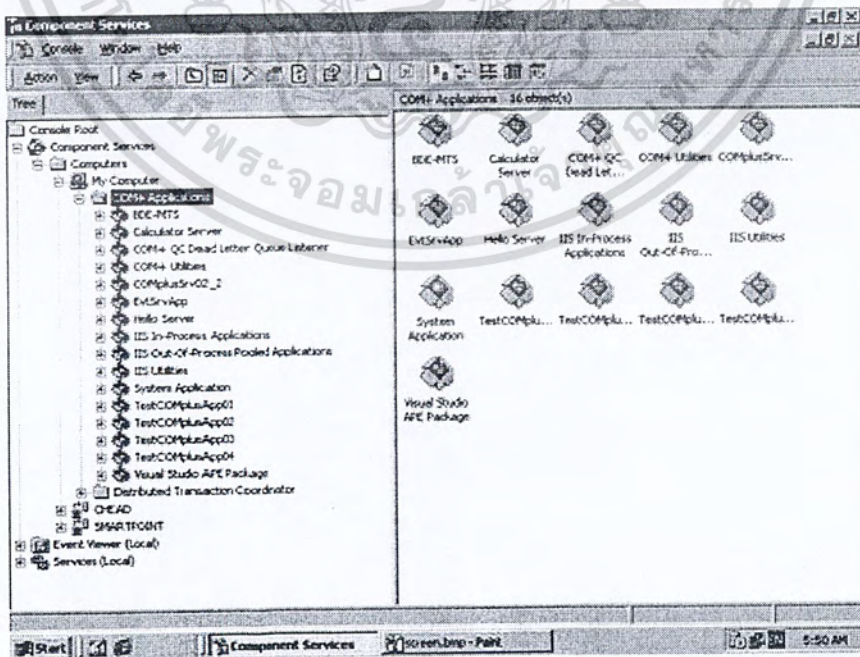
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10-8 แสดง COM+ Server Application 2 ตัวใช้ Library Application ร่วมกัน

#### 10.4 การสร้าง COM+ แอปพลิเคชัน

ในการสร้างและปรับแต่ง COM+ ออบเจกต์ และ COM+ แอปพลิเคชัน นั้นจะใช้ Component Service Explorer โดยที่สามารถเรียกใช้ได้จากปุ่ม Start \ Programs \ Administrative Tools \ Component Services \ ซึ่งจะมีลักษณะหน้าจอดังรูปที่ 10-9

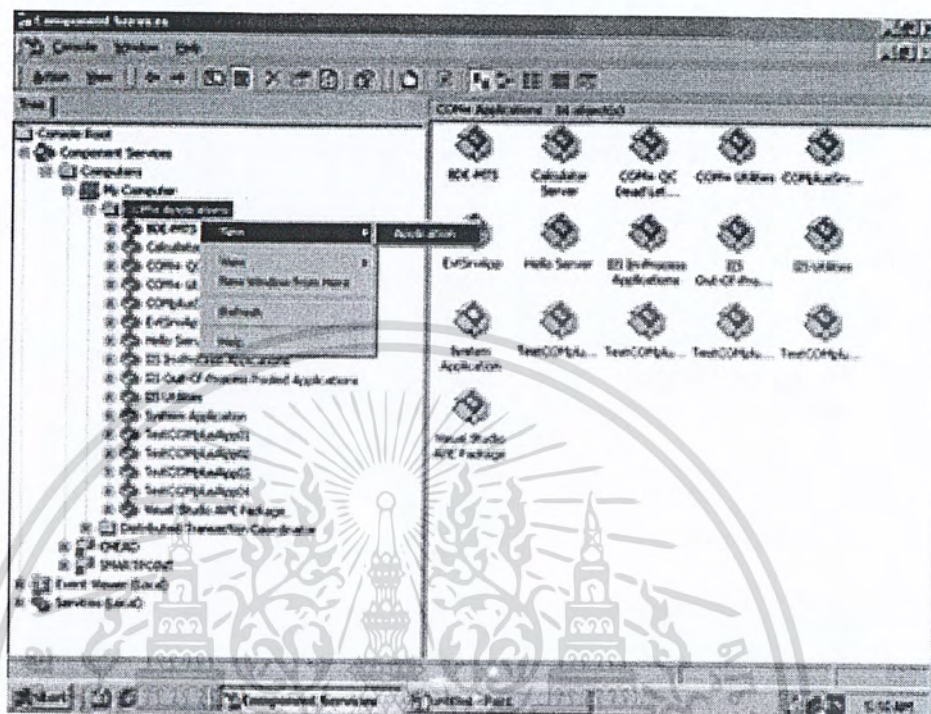


รูปที่ 10-9 แสดง Component Service Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการสร้าง COM+ Application ทำได้ดังนี้

- 1) คลิกขวาที่ Console Root \ Component Services \ Computers \ My Computer \ COM+ Application ดังรูปที่ 10-10

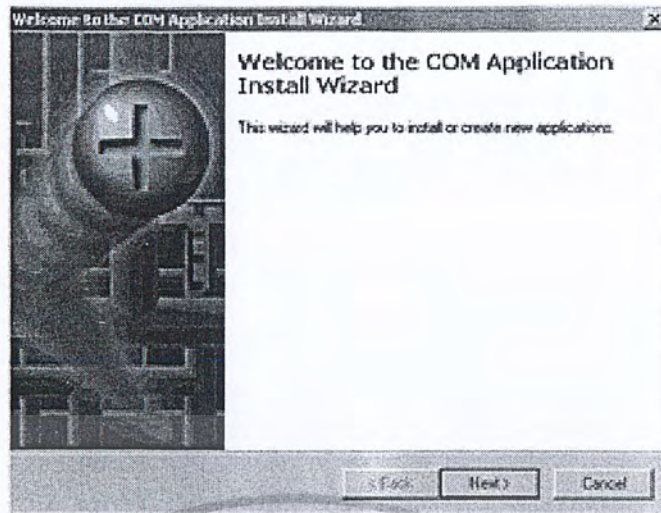


รูปที่ 10-10 แสดงขั้นตอนแรกในการสร้าง COM+ Application

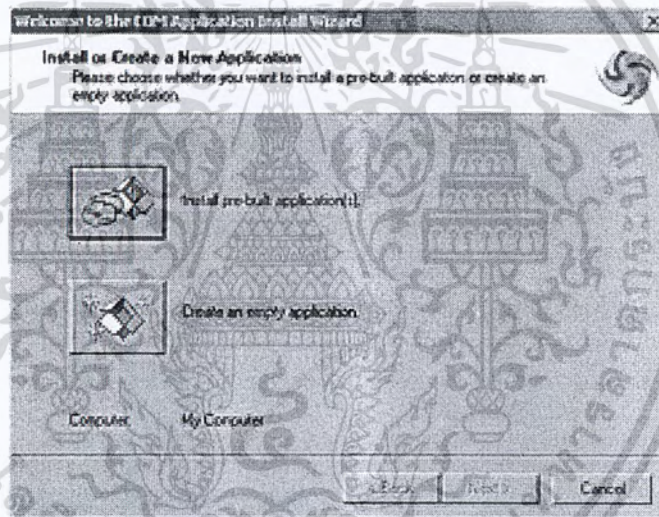
- 2) จากรูปที่ 7-10 ทำการเลือก New \ Application จะปรากฏหน้าจอ ดังรูปที่ 10-11
- 3) หลังจากกดปุ่ม Next จะปรากฏหน้าจอ ดังรูปที่ 10-12
- 4) กดปุ่ม Create an empty application แล้วจะเกิดหน้าจอ ดังรูปที่ 10-13
- 5) ใส่ชื่อ Application และเลือกชนิดการ Activation จากนั้นกดปุ่ม Next จะเกิดหน้าจอ ดังรูปที่ 10-14
- 6) เลือกลักษณะของ Identity
- 7) กดปุ่ม Next เพื่อสิ้นสุดการสร้าง COM+ Application

เมื่อทำการสร้าง Application ใหม่เสร็จแล้วก็จะมีไอคอนได้ COM+ Application ใน Component Service ดังรูปที่ 10-15

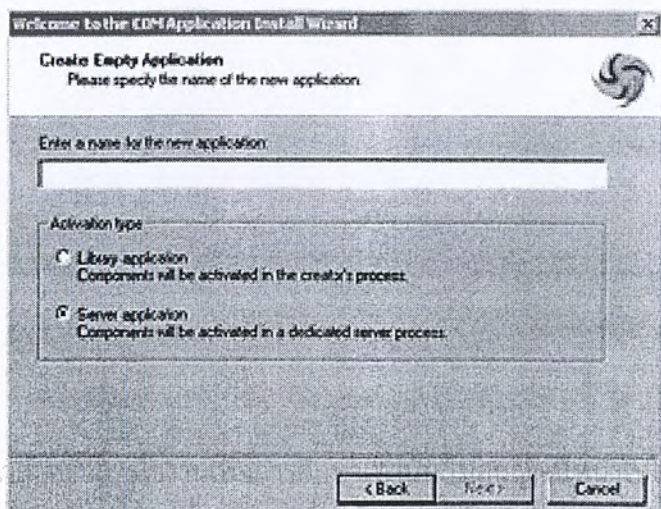
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10-11 แสดงหน้าจอแรกของ COM+ Application Install Wizard



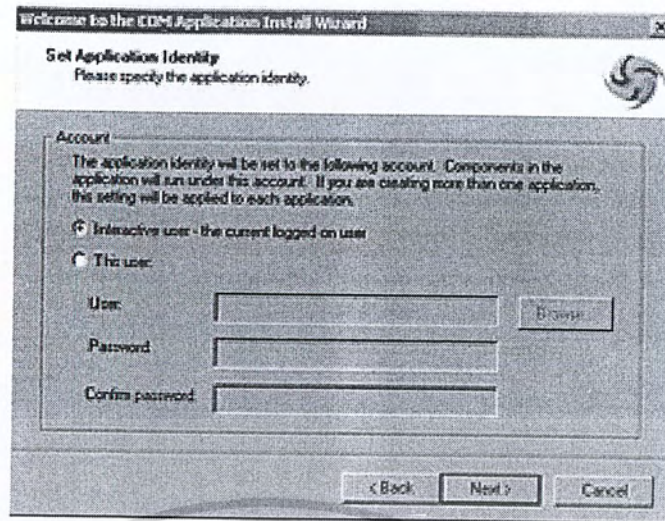
รูปที่ 10-12 แสดงหน้า Install Pre-Built COM+ Application หรือ สร้างใหม่



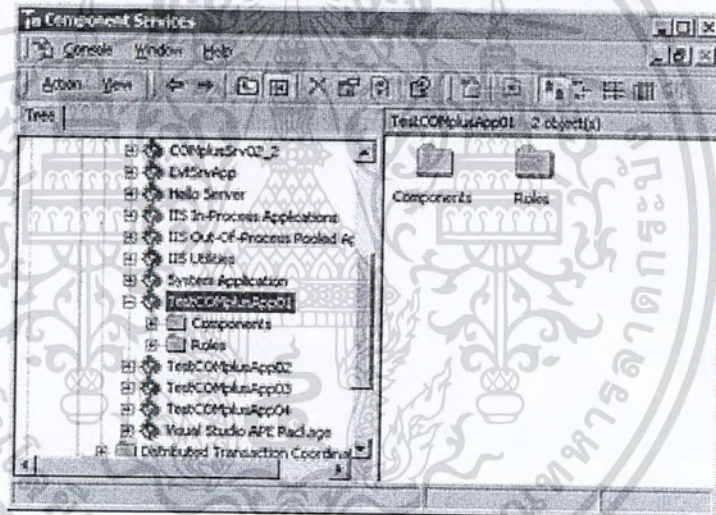
เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อใช้ภายในเท่านั้น ไม่ควรเผยแพร่สู่ภายนอกโดยไม่ได้รับอนุญาตจากฝ่ายที่เกี่ยวข้อง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลใดๆ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 10-13 แสดงการกำหนดชื่อของ Application และ ชนิดการ Activate



รูปที่ 10-14 แสดงการเลือก Identity ของ COM+ Application



รูปที่ 10-15 แสดง Component Service Explorer ที่แสดง Application ใหม่

เมื่อเราสร้าง COM+ Application เสร็จดังรูปที่ 10-15 จะมีไอคอนใหม่ที่เกิดขึ้นใน Application นั้นซึ่งได้แก่ Components และ Roles เราจะใส่คอมโพเนนต์ที่จะอยู่ใน Application ไว้ใน Components ส่วนการควบคุม Fine - Grained Security นั้นจะใช้ Role ซึ่งจะได้อธิบายต่อไป สำหรับขั้นตอนในการดูรายละเอียดในระดับ Application นั้นดูได้โดย

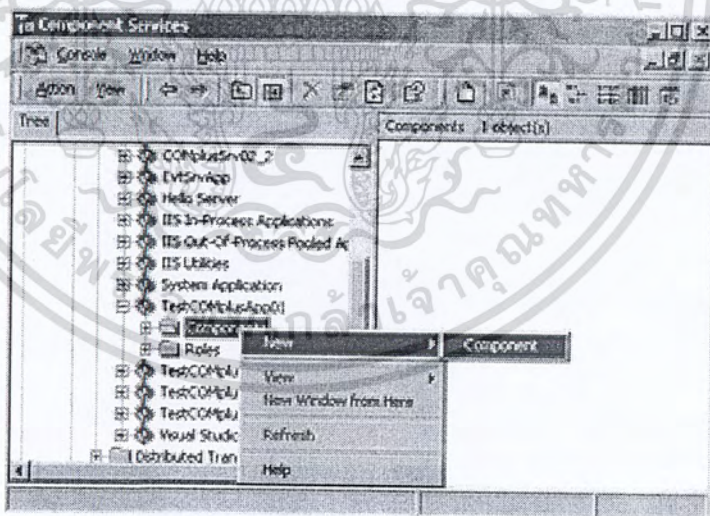
- 1) คลิกขวาที่ไอคอนของ COM+ Application ที่ต้องการจะดูรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เลือก Properties จาก เมนู Context (คุณสมบัติของ COM+ Application จะแสดงดังรูปที่ 10-16 ซึ่งจะใช้ในการปรับแต่งคุณสมบัติต่าง ๆ ของ COM+ ในระดับ Application ซึ่งจะได้อธิบายต่อไป



รูปที่ 10-16 แสดง General Tab ของ COM+ Application Properties Window



รูปที่ 10-17 แสดงคอมโพเนนต์โฟลเดอร์สำหรับ COM+ Application

ในการใส่คอมโพเนนต์ให้กับ COM+ Application จะใช้ COM ออบเจ็กต์ธรรมดา ซึ่งเป็น In-Process ตัวใด ๆ ก็ได้ใส่ลงใน Application ซึ่งมีขั้นตอนดังนี้

1) คลิกขวาที่โฟลเดอร์ Components ซึ่งอยู่ใต้ COM+ Application ดังรูปที่ 10-17

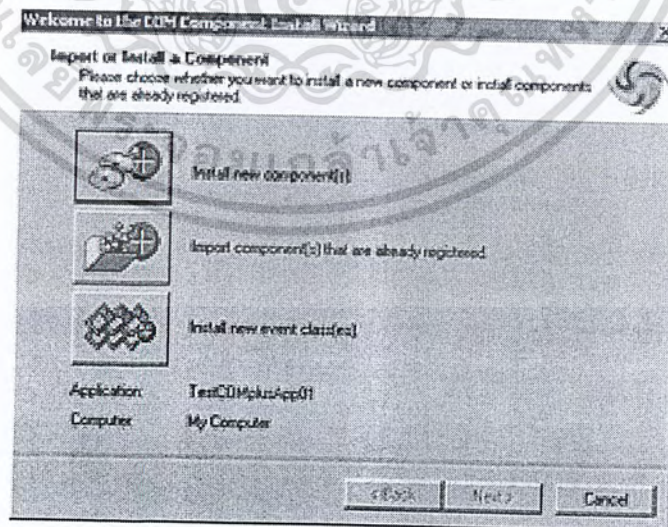
2) เลือก New \ Component จาก เมนู Context (จะเกิด Component Install Wizard ดังรูปที่ 10-18)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) กดปุ่ม Next (ขั้นตอนที่ 2 ของ Component Install Wizard จะแสดงดังรูปที่ 10-19)
- 4) กดปุ่ม Install New Components (ขั้นตอนที่ 3 ของ Component Install Wizard จะแสดงดังรูปที่ 10-20)
- 5) กดปุ่ม Add.. (จะปรากฏไดอะล็อก Select Files to Install)
- 6) เลือกไฟล์ DLL แล้วกดปุ่ม Open
- 7) กดปุ่ม Next และ Finished

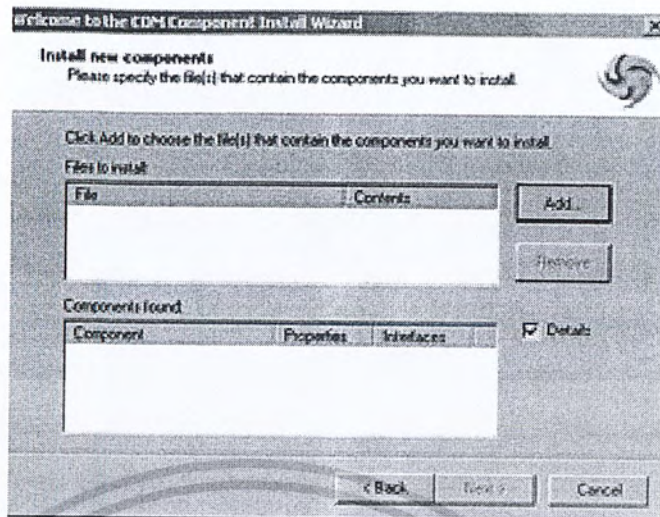


รูปที่ 10-18 แสดงหน้าจอแรกของ Component Install Wizard



รูปที่ 10-19 แสดงการเลือก Install ระหว่างคอมโพเนนต์ใหม่ กับคอมโพเนนต์ที่ลงทะเบียนเรียบร้อยแล้ว

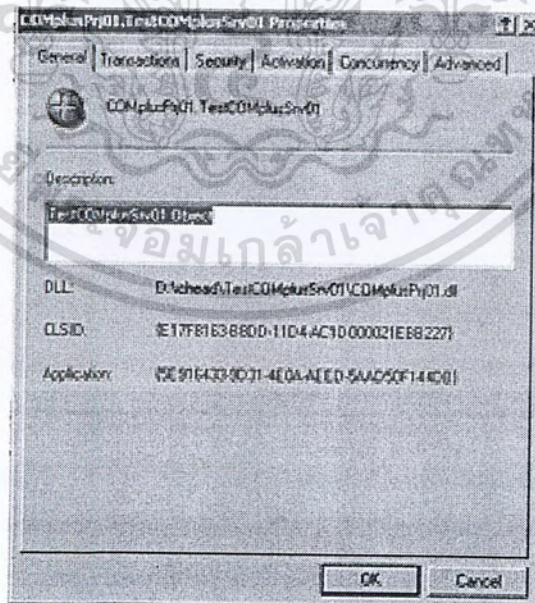
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10-20 แสดงการเลือก คอมโพเนนต์ที่ต้องการจะติดตั้ง

สำหรับการ ดูและปรับแต่งค่าคุณสมบัติต่างของ COM+ ออบเจ็กต์ในระดับคลาสสามารถทำได้ โดยมีขั้นตอนดังนี้

- 1) คลิกขวาที่คลาสใน Component tab
- 2) เลือก Properties จากเมนู Context (คุณสมบัติ COM+ คอมโพเนนต์จะแสดงดังรูปที่ 10-21)



รูปที่ 10-21 แสดงไดอะล็อก COM+ Component Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 11

### เรดโมเดลของ COM+

#### 11.1 โพรเซส ( Process )

อินสแตนซ์ของโปรแกรม ในแต่ละ โพรเซส จะถือครองทรัพยากรต่าง ๆ เช่น ไฟล์ และ หน่วยความจำ ทรัพยากรทั้งหมดนี้จะถูกทำลายเมื่อ โพรเซส นั้นถูกทำลาย อย่างไรก็ตาม ทรัพยากรบางอย่างก็ไม่สามารถทำงานอะไรได้ และแต่ละ โพรเซส จะต้องมียังน้อย 1 เรด

#### 11.2 เรด ( Thread )

ส่วนของการประมวลผลโดยผ่านโค้ดที่อยู่ใน โพรเซส ทุก ๆ Thread จะมี stack และ CPU-state เป็นของตัวเอง และเนื่องจากมี Stack เป็นของตัวเองดังนั้นตัวแปรของแต่ละThread จึงมีความเป็นส่วนตัวจากเรด อื่นๆ

#### 11.3 มัลติเรด ( Multi-Thread )

ทุก ๆ Thread จะต้องมียังน้อย 1 เรด ซึ่งเรียกว่า ไพมารีเรด (Primary Thread) ในการเขียนโปรแกรมให้สามารถใช้ได้เพียงแค่ เรด เดียวนั้นจะง่าย เนื่องจากไม่ต้องกังวลในเรื่องการที่มีหลาย เรด อ่านค่าเดียวกันพร้อมกัน แต่ข้อดีของการที่สามารถทำงาน ได้หลาย ๆ เรด พร้อมกันนั้นคือสามารถทำงานได้อย่างมีประสิทธิภาพกว่า เนื่องจาก เรด ที่ช้าเนื่องจากต้องติดต่อกับอุปกรณ์ภายนอก เช่น แผ่นดิสก์, เครื่องพิมพ์ เป็นต้น ก็จะไม่ขวางการทำงานของ เรด อื่น เนื่องจาก เรด อื่นสามารถทำงานได้

ในระบบปฏิบัติการสมัยใหม่ทั้งหลาย เช่น Windows 98 และ Windows 2000 นั้นสามารถประมวลผล เรด ได้หลาย ๆ ตัวได้อย่างต่อเนื่อง ระบบปฏิบัติการจะจัดการกับงานเหล่านี้โดยการตั้งช่วงเวลาในการประมวลผลให้กับแต่ละ เรด เมื่อแต่ละ เรด ทำงานจนหมดช่วงเวลาของตัวเองแล้ว ระบบปฏิบัติการก็จะทำการบันทึกสถานะของ ซีพียู และ เรด นั้นไว้แล้วทำการเปลี่ยนให้ เรด อื่นมาทำงานแทน จนกว่าจะถึงเวลาที่มันจะทำการประมวลผลใหม่ก็จะทำการ โหลดข้อมูลเหล่านั้นมาเพื่อทำงานต่อไป

#### 11.4 Thread-Safe

ซอร์สโค้ดที่สามารถทำงานในแบบ มัลติเรด (Multi-Thread ) ได้อย่างปลอดภัย ซึ่งมักจะใช้วิธี thread-synchronization primitive เช่น Critical Section ในขณะที่จะประมวลผลซึ่งไม่ปลอดภัย

#### 11.5 Multi-Threading และ COM

จุดประสงค์หลักจุดประสงค์หนึ่งของ COM คือการสร้างออบเจกต์ให้ง่ายต่อการนำกลับมาใช้ใหม่ โดยไม่ขึ้นกับภาษาที่ใช้พัฒนา และตำแหน่งของตัวออบเจกต์ อย่างไรก็ตาม ปัญหาของระบบซึ่งเป็นแบบ มัลติเรด นั้นจะต้องพิจารณาใหม่ เนื่องจากเราไม่สามารถนำระบบที่ไม่เป็นแบบThread-Safeมาทำงานในระบบงานที่เป็นแบบ Multi-Thread ได้

ดังนั้นนอกจากที่ต้องการให้ คอมโปเนนต์นั้นสามารถทำงานได้โดยไม่ขึ้นกับภาษาที่ใช้พัฒนา และ ตำแหน่งของตัวคอมโปเนนต์แล้ว เรายังต้องการให้มีความเป็นอิสระต่อลักษณะที่แตกต่างกันของ เรด ด้วย

ในระบบที่เป็น COM นั้นทำได้โดยการพิจารณาว่าออบเจกต์ที่จะเรียกใช้งานนั้นเป็นแบบ Thread-Safe หรือไม่ค่าที่เป็นตัวบ่งชี้ว่าเป็นหรือไม่นั้นสามารถดูได้จากวิธีทริกของออบเจกต์นั้นเมื่อระบบงานที่เป็นแบบMulti-Thread จะพยายามเรียกใช้ออบเจกต์ซึ่งมีลักษณะไม่เป็น Thread-Safe COM+รันไทม์ จะทำการ Serialize การเรียกใช้ต่างนั้นให้โดยอัตโนมัติ

จากหลักการที่ได้กล่าวมาทำให้เกิดปัญหาที่ตามมาหลายประการดังนี้

- จะทำให้ COM รันไทม์ รู้ได้อย่างไรว่าแอปพลิเคชันนั้นเป็น มัลติเธรด
- จะทำให้ COM ออบเจกต์บ่งชี้ว่าตัวเองนั้นเป็น Thread-Safe
- COM+ รันไทม์ จะทำการ Serialize การเรียกเมธอดได้อย่างไรเมื่อมันทำการเรียกใช้ COM ออบเจกต์ที่ไม่เป็น Thread-Safe จากแอปพลิเคชันที่เป็น มัลติเธรด

ก่อนที่จะศึกษาถึงวิธีการที่จะแก้ไขปัญหเหล่านี้ เราต้องเข้าใจถึงหลักการพื้นฐานที่อยู่ภายในแต่ละ โพรเซส เสียก่อน ซึ่งหลักการพื้นฐานนี้คือ อพาร์ตเมนต์ นอกจากนั้นปัญหาในของวิธีการ Serialize ของ COM รันไทม์ นั้นก็จะใช้อพาร์ตเมนต์จัดการด้วย

#### 11.6 อพาร์ตเมนต์

เป็น สถานที่ทางจินตภาพ (logical construct) ที่มีลักษณะคล้ายกับสภาพแวดล้อมของ เธรด ซึ่ง COM ออบเจกต์ หนึ่งหรือมากกว่า ออบเจกต์สามารถประมวลผลได้ภายใน และ COM ออบเจกต์ทุกตัวที่อยู่ใน โพรเซส จะต้องอยู่ใน อพาร์ตเมนต์ใด อพาร์ตเมนต์หนึ่ง เนื่องจากในแต่ละ โพรเซส ก็สามารถมีได้หลาย อพาร์ตเมนต์

เฉพาะ เธรด ที่เข้าถึง อพาร์ตเมนต์ ของออบเจกต์นั้นๆ ถึงจะสามารถเรียกใช้เมธอดของออบเจกต์นั้นได้โดยตรง สำหรับ เธรด อื่นนั้นจะสามารถเรียกใช้ได้โดยผ่านทาง การส่งแอสเสจไปยัง เธรด ใด ๆ ที่อยู่ใน อพาร์ตเมนต์ นั้นแทน

อพาร์ตเมนต์ มีประเภทต่าง ๆ ดังนี้

- 1) แบบที่สามารถมีได้เพียง Thread เดียวซึ่งเรียกว่า STA (Single - Threaded Apartment) ซึ่งทุก ๆ ออบเจกต์ใน อพาร์ตเมนต์ นั้นจะถูกเข้าถึง ได้จากเธรดเพียงเธรดเดียว ซึ่งจะมีปัญหาเกี่ยวกับการใช้ทรัพยากรเมื่อต้องทำงานในแบบที่เป็น มัลติเธรด
- 2) แบบที่สามารถมีได้หลายThread หรือ MTA (Multi-Threaded Apartment) ซึ่งสามารถให้บริการ เมธอดต่าง ๆ ที่อยู่ใน อพาร์ตเมนต์ ให้กับ Thread ต่าง ๆ ได้อย่างต่อเนื่อง แต่ ออบเจกต์ที่อยู่ใน MTA ได้นั้นต้องเป็น Thread-Safe

แต่ละ โพรเซส นั้นสามารถมีได้หลาย STA แต่จะมีเพียงแค่ MTA เดียวเท่านั้น

ออบเจกต์ทุกตัวจะระบุว่าตัวมันนั้นต้องการ สภาพแวดล้อมที่มันต้องการ โดยระบุไว้ที่ โมเดล

ของเธรด ซึ่งสามารถดูได้จาก วิธีทริกของระบบที่ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HKEY\_CLASSES\_ROOT\CLSID\CLSID\_FOR\_A\_COM\_CLASS\ThreadingModel  
เช่น

HKEY\_CLASSES\_ROOT\CLSID\{06637330-314D-11D3-998A-E0EC08C10000}\

ThreadingModel = Apartment

ค่าที่เป็นไปได้ของ โมเดลเรด มีทั้งหมด 5 ค่าได้แก่

1) Single มีชนิดของ อพาร์ทเมนต์ เป็น STA ออบเจ็กต์ประเภทนี้จะไม่เป็น Thread-Safe ไม่ใช่แค่ตัวออบเจ็กต์เท่านั้นแต่ยังรวมถึง Class Factory สำหรับคลาส และฟังก์ชัน DllGetClassObject และ DllCanUnloadNow ก็เช่นกัน Single-Thread สามารถทำการเรียกใช้ทุก ๆ เมธอดที่อยู่ในเซิร์ฟเวอร์

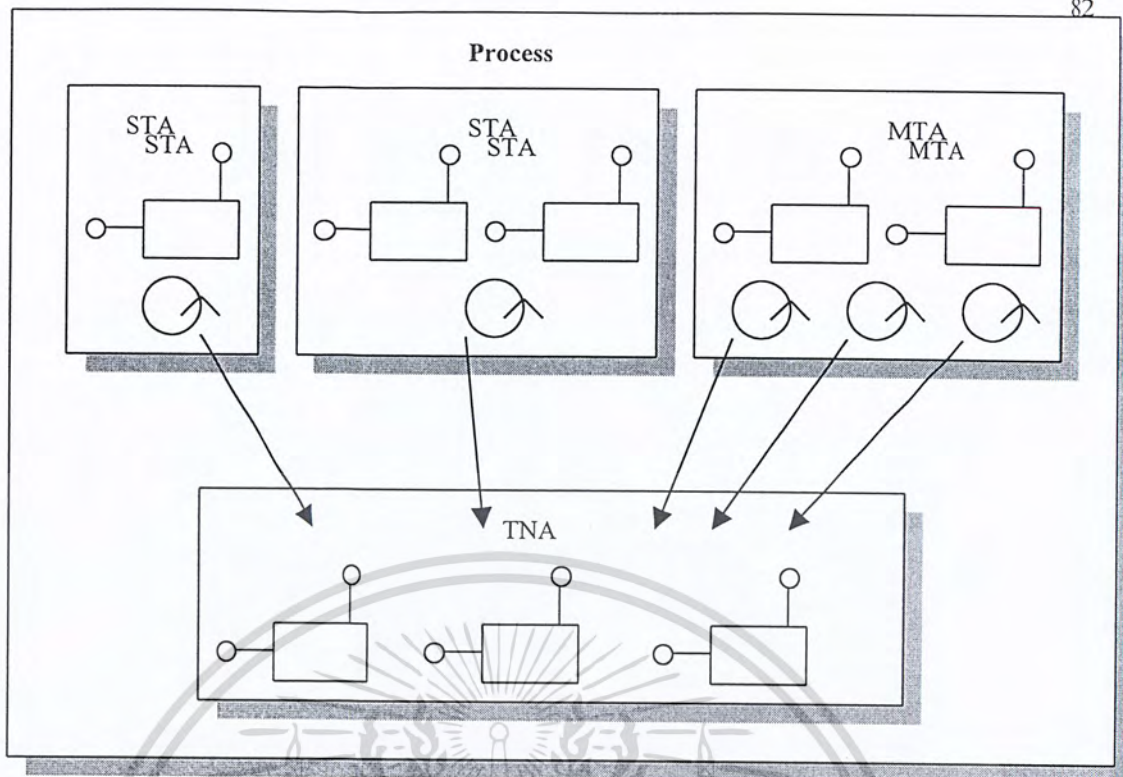
2) Apartment มีชนิดของอพาร์ทเมนต์เป็น STA ออบเจ็กต์ประเภทนี้จะสามารถป้องกันปัญหาของระบบ Multi-Thread ได้เฉพาะในระบบดับคลาสเท่านั้น เนื่องจาก Class Factory และ ฟังก์ชัน DllGetClassObject และ DllCanUnloadNow ในคลาสของมันเป็น Thread-Safe แต่ในระดับ อินสแตนซ์จะถูกจำกัด เนื่องจากการเข้าถึงแต่ละอินสแตนซ์จะถูกกำหนดให้เป็น Single-Thread เท่านั้น

3) Free มีชนิดของ อพาร์ทเมนต์ เป็น MTA ออบเจ็กต์ประเภทนี้สามารถป้องกันปัญหาของระบบ มัลติเรด ทั้งสถานะต่ออินสแตนซ์ และสถานะต่อคลาสโดยใช้ Thread Synchronization Primitives เช่น Critical Section หรือ Mutex ออบเจ็กต์เหล่านี้สามารถทำงานในสถานะที่เป็น มัลติเรด ได้อย่างปลอดภัย

4) Both สามารถมีชนิดของอพาร์ทเมนต์ ได้ทั้ง STA และ MTA ออบเจ็กต์ประเภทนี้จะสร้างตัวเองอยู่ในอพาร์ทเมนต์เดียวกับ ไคลเอนต์ของมันซึ่งทำให้มีประสิทธิภาพสูงสุด และทำงานในเรดเดียวกับผู้ที่เรียกมัน และเนื่องจากมันสามารถทำงานได้ในระบบที่เป็นมัลติเรดมันจึงป้องกันปัญหาที่จะเกิดในระดับคลาส และ ระดับอินสแตนซ์

5) NeuTral มีชนิดของอพาร์ทเมนต์เป็น TNA (Thread Neutral Apartment) ออบเจ็กต์ประเภทนี้จะทำงานในเรดของผู้เรียกแต่อยู่ในอพาร์ทเมนต์ของมันเองแทนที่จะเป็นอพาร์ทเมนต์ ของมันเนื่องจากเรดที่อยู่ในอพาร์ทเมนต์ ประเภทอื่นใน โพรเซสเดียวกันสามารถเข้าถึง TNA ได้ตลอดเวลา และออกจากอพาร์ทเมนต์เมื่อมันทำงานเมื่อการทำงานของเมธอดนั้นเสร็จ เนื่องจากอพาร์ทเมนต์ประเภทนี้สามารถทำการ Thread-Synchronization primitives ได้ภายในตัวของมันเอง หรือสามารถ Thread-Synchronize ได้ด้วยบริการของ COM+ ซึ่งจะได้อธิบายโดยละเอียดในภายหลัง

ความสัมพันธ์ระหว่าง โพรเซส, อพาร์ทเมนต์, เรด และ COM ออบเจ็กต์อธิบายได้ดังรูปที่ 11-1



รูปที่ 11-1 แสดง Process, Apartment, Thread และ COM ออบเจกต์

### 11.7 วิธีที่ COM+ runtime จัดการกับ User Apartment

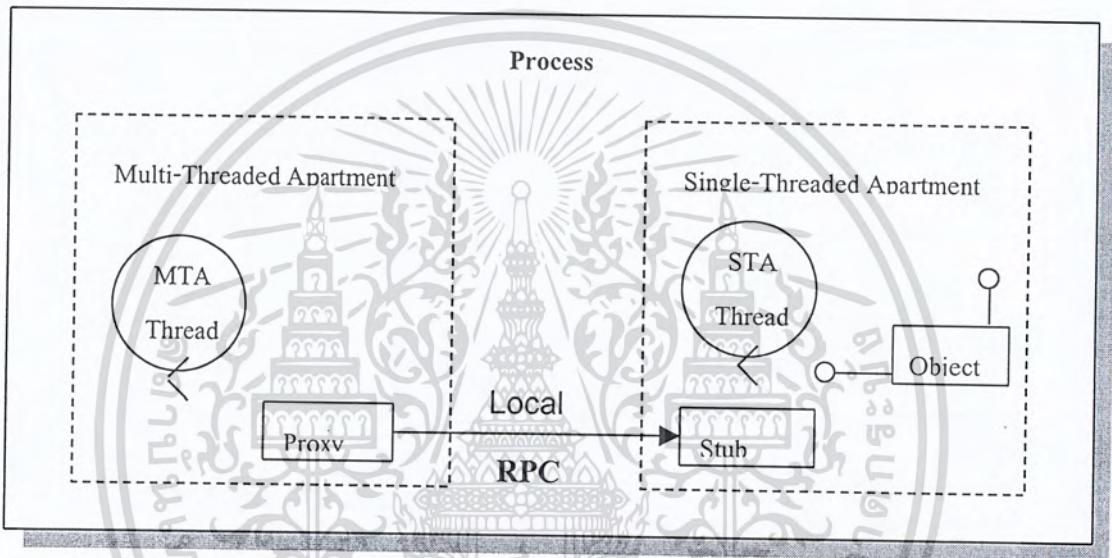
ถ้า COM ออบเจกต์ระบุว่าตัวมันนั้นเป็น Thread-Safe (ThreadingModel = Free) ตัว COM + รันไทม์ จะมองตัวออบเจกต์นั้นรันอยู่ใน MTA หากออบเจกต์นั้นถูกสร้างใน MTA Thread (ตัวโคลเอนต์เป็น มัลติเธรด) แล้ว COM+ เธรด ก็ไม่จำเป็นต้องทำอะไรเพิ่มเติม แต่หากตัวออบเจกต์นั้นถูกสร้างใน STA เธรด (ตัวโคลเอนต์เป็น Single-Thread) ทาง COM+ รันไทม์ จะทำการสร้าง MTA เธรด ขึ้นใหม่การสร้าง STA เธรด จะได้รับ Marshaled Interface Pointer (ซึ่งจะได้อธิบายต่อไป) หลาย ๆ Thread สามารถเข้าถึง MTA และทุก ๆ เธรด ที่เข้าถึง MTA สามารถเรียกใช้ทุก ๆ เมธอดจากทุก ๆ ออบเจกต์ที่อยู่ใน MTA การที่สามารถทำได้ดังที่กล่าวมาแล้ว ก็เนื่องเธรด โมเดล แบบ Free นั้นมีส่วนจัดการ การ Synchronization ให้โดยอัตโนมัติจึงไม่ต้องกังวลเกี่ยวกับปัญหาที่เกิดขึ้นในระบบที่เป็น Single-Thread หรือ Multi-Thread

แต่ถ้า COM ออบเจกต์นั้นไม่เป็น Thread-Safe (เธรดโมเดล เป็น Apartment หรือ Single) COM+ รันไทม์ จะมองว่าออบเจกต์นั้นรันอยู่ที่ เธรด STA ถ้าเธรดโมเดล ของออบเจกต์เป็น อพาร์ตเมนต์ และออบเจกต์ถูกสร้างจากเธรด STA ออบเจกต์นั้นจะถูกสร้างอยู่ในอพาร์ตเมนต์เดียวกับเธรด ที่สร้างมัน ถ้าออบเจกต์นั้นถูกสร้างจาก MTA Thread เมื่อ COM+ รันไทม์ สร้างเธรด และ STA ของออบเจกต์ ให้มัน COM+ รันไทม์ จะส่ง Marshaled Interface กลับมาให้กับเธรด MTA ด้วย และถ้าเธรดโมเดล ของออบเจกต์เป็น Single ออบเจกต์นั้นจะอยู่ใน Primary STA ของ Process (Primary STA เป็น STA แรกที่ถูกสร้างในโพรเซส) นอกจากนี้ COM+ รันไทม์ จะสร้าง ออบเจกต์ไว้ใน primary STA และส่ง Marshaled Interface คืนไปยัง อพาร์ตเมนต์ ที่เรียกมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 11.8 การเรียกใช้เมธอดแบบ Inter-Apartment

ถึงแม้ว่าเราจะอนุญาตให้เทรดเดิยวสามารถใช้งานเมธอดที่อยู่ใน STA ได้โดยตรง แต่ก็ไม่ได้หมายความว่า จะไม่อนุญาตให้เทรดอื่นๆเข้าไปใช้งานเมธอดนั้นแต่จะต้องเรียกใช้เมธอดนั้นโดยอ้อม (Indirect Method Call) หากมีเทรดที่อยู่ใน อพาร์ทเมนต์หนึ่งเรียกใช้เมธอดที่อยู่ในออบเจกต์ที่อยู่ใน อพาร์ทเมนต์อื่น COM+ รันไทม์จะตรวจสอบว่าการเรียกใช้เมธอดถูกต้องหรือไม่โดยดูจากว่าการเรียกใช้เมธอดนั้นจะเกิดจากเทรด ที่อยู่ในอพาร์ทเมนต์ของออบเจกต์ที่ถูกเรียกใช้ สำหรับลักษณะในการติดต่อกัน จะเหมือนกับมาเซตลิ่งซึ่งใช้ในการเรียกใช้เมธอดผ่าน โพรเซสและผ่านเครื่องซึ่งเรียกวิธีนี้ว่า Intra-Process Marshaling หรือ Thread Switch ดังรูปที่ 11-2



รูปที่ 11-2 แสดง Intra-Process Marshaling

อย่างไรก็ดีข้อวิธีการ Intra-Process Marshaling ก็มีจุดด้อยที่ค่อนข้างซ้ำมากเมื่อเทียบกับการเรียกใช้ เมธอดภายใน Apartment เดียวกัน

### 11.9 Marshalling Interface Pointer ระหว่าง อพาร์ทเมนต์

เมื่อ เทรด ใดๆ ซึ่งรันอยู่ใน อพาร์ทเมนต์ หนึ่ง เรียกใช้เมธอดซึ่งอยู่ในออบเจกต์ที่อยู่ในอีก อพาร์ทเมนต์ หนึ่งจะทำให้ Interface Pointer จัดการติดต่อดีสารระหว่าง อพาร์ทเมนต์ โดยที่จะติดตั้งตัว ฟร็อกซ์อยู่ที่ฝั่งไคลเอนต์ และติดตั้งตัวฟร็อกซ์ ที่ฝั่งของตัวออบเจกต์

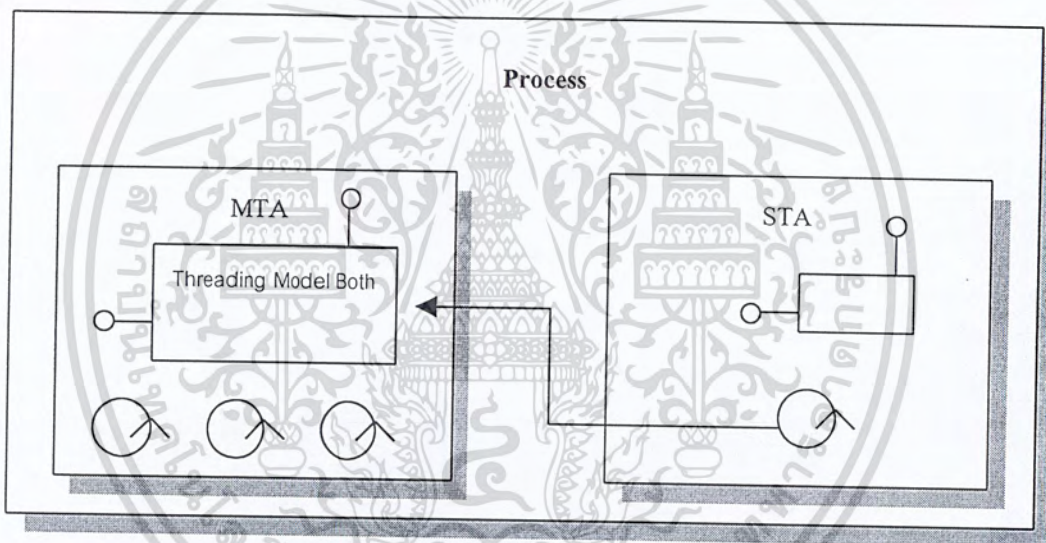
### 11.10 Threading Model แบบ Both และ Neutral

รายละเอียดของเทรด โมเดลทั้งสองนี้มีจุดที่เหมือนและแตกต่างกันอยู่พอสมควร จุดที่เหมือนกัน ก็คือเทรด โมเดลทั้งสองถูกออกแบบมาเพื่อหลีกเลี่ยงการใช้ Thread-Switch จากการเรียกใช้เมธอดซึ่งอยู่ ต่างอพาร์ทเมนต์โดยการให้ออบเจกต์ซึ่งมีเทรด โมเดลแบบนี้ทำงานในเทรดของไคลเอนต์ ส่วนความแตกต่างระหว่างเทรด โมเดล ทั้งสองนั้นสามารถอธิบายได้จากรายละเอียดของมันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

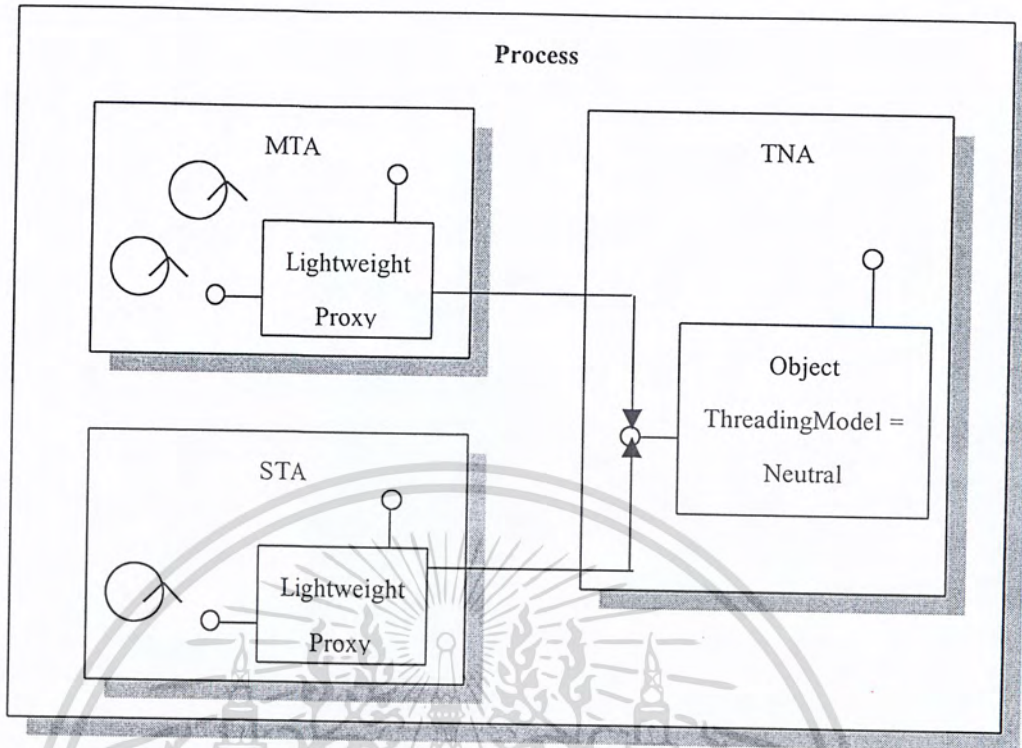
- Both เป็นชื่อที่ตั้งได้ไม่ค่อยดีนัก เนื่องจากในสมัยที่ออกแบบ เทรด โมเดล ชนิดนี้ยังมีอพาร์ตเมนต์ แต่สองแบบคือ STA และ MTA แต่ในความเป็นจริงแล้วมันก็สามารถทำงานให้กับ ออบเจ็กต์ที่มีเทรด โมเดลเป็นแบบ TNA ได้เช่นเดียวกัน และเนื่องจากออบเจ็กต์ที่สามารถใช้เทรด โมเดล ชนิดนี้ได้ นั่นอาจเป็น MTA หรือ TNA ก็ได้ ดังนั้นออบเจ็กต์ทุกตัวที่จะมี เทรด โมเดล ชนิดนี้ต้องมีความเป็น Thread-Safe ด้วย

จุดด้อยของเทรด โมเดลชนิดนี้ก็คือ ไม่สามารถที่จะจำกัดการเรียกใช้ Thread-Switch ได้ เช่น ในกรณีที่มี MTA Thread เรียกใช้ ออบเจ็กต์ตัวหนึ่งที่มี เทรด โมเดลเป็น Both ตัว ออบเจ็กต์นั้นก็จะถูกสร้างไว้ใน อพาร์ตเมนต์ MTA นั้น หากมี STA เทรด อีก เทรด หนึ่ง ต้องการเรียกใช้ออบเจ็กต์ตัวนั้น ดังรูปที่ 11-3 ในกรณีนี้ก็ยังจำเป็นต้องเรียกใช้ Thread-Switch หรือ หากจะหลีกเลี่ยงจริงๆ ก็ต้องใช้ FTM (Free-Threaded Marshaler) ร่วมกับ GIT ซึ่งค่อนข้างยุ่งยากเมื่อเทียบกับการใช้ Neutral



รูปที่ 11-3 แสดงเทรด ที่เป็น MTA และ STA และออบเจ็กต์ที่มี เทรด โมเดลชนิด Both

- Neutral เป็น Threading Model ที่ถูกเพิ่มเข้ามาใน Windows 2000 ซึ่งถูกออกแบบมาในแนวทางเดียวกันกับ Both โดยปราศจากความซับซ้อนในการใช้ FTM และ GIT ความแตกต่างระหว่างทั้งสองแบบนี้ก็คือ ทุก ๆ Thread นั้นไม่มีขอบเขตที่แบ่งแยกกันอย่างชัดเจน Thread ใน STA และ MTA สามารถเรียกใช้ออบเจ็กต์ที่อยู่ใน TNA โดยผ่านทาง Lightweight Proxy ดังรูปที่ 11-4



รูปที่ 11-4 แสดงอพาร์ทเมนต์ชนิด STA และ MTA เรียกใช้ออบเจกต์ที่รันอยู่ใน TNA

Lightweight Proxy นั้นจะไม่ทำการ Thread Switch แต่ถูกขอร้องอื่นของอพาร์ทเมนต์ก็ยังใช้ได้และเช่นเดียวกับเทรดโมเดลแบบ Both ทุกๆ ออบเจกต์จะเป็น Thread-Safe การใช้งานเทรดโมเดลชนิดนี้ไม่จำเป็นต้องใช้โปรแกรมลงไปเองเนื่องจากเป็นส่วนบริการของ สภาพแวดล้อม COM+ รันใหม่อยู่แล้วมันจะทำการ Thread-Synchronization โดยไม่จำเป็นต้องเขียนโค้ดลงไปเอง สำหรับรายละเอียดของการทำ Thread-Synchronization จะได้อธิบายภายหลัง

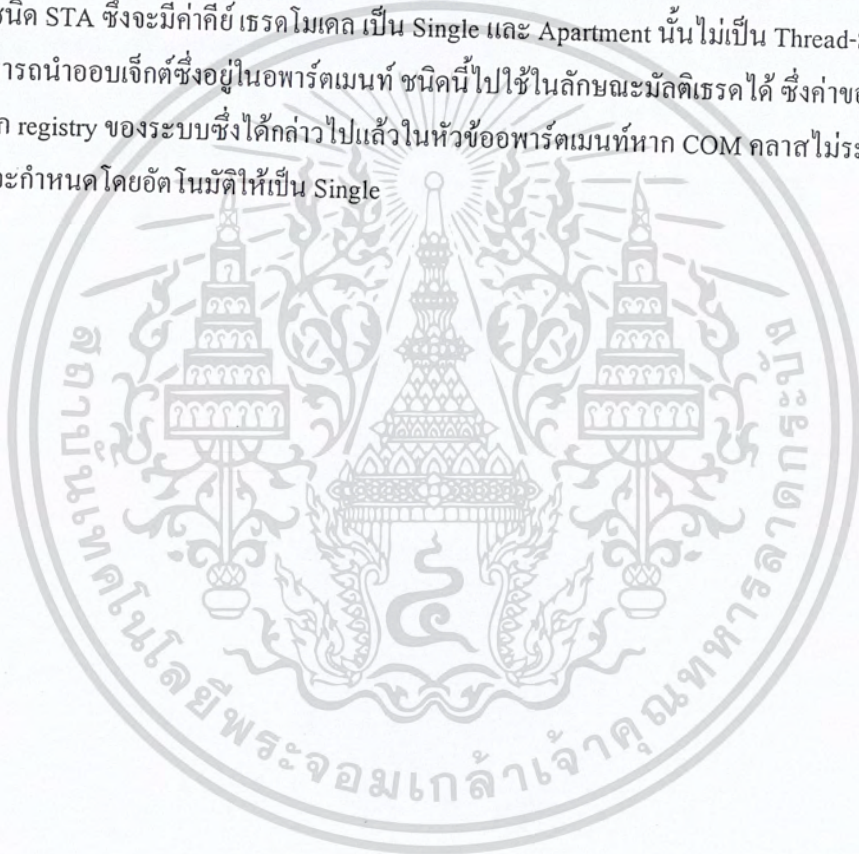
อย่างไรก็ตามเทรดโมเดลชนิด TNA นั้นก็ไม่สามารถที่ใช้ได้ คอมโพเนนต์ที่ถูกออกแบบมาเพื่อให้มีเพียงเทรดที่เหมาะสมเทรดเดียวเท่านั้นที่สามารถเรียกใช้มันได้ (เราเรียกเทรด ประเภทนี้ว่า Thread affinity) ซึ่งได้คอมโพเนนต์ ประเภทที่ต้องแสดงผล (Visual Component) เป็นต้น สำหรับออบเจกต์พวกนี้จำเป็นต้องใช้เทรดโมเดลที่เป็นอพาร์ทเมนต์เป็นชนิด STA อย่างเดียว เท่านั้น

### 11.11 วิธีพิจารณาว่าแอปพลิเคชันเป็นมัลติเธรด

จากที่ได้กล่าวถึงเรื่องอาร์ทเมนต์คำถามนี้จึงเป็นคำถามที่ผิด คำถามที่ถูกต้องควรเป็นว่า COM+ รันไทม์ จะพิจารณาว่าเธรดที่เข้ามาเรียกใช้ ออบเจกต์นั้นเป็นเธรดที่อยู่ในส่วนของสภาพแวดล้อมที่มี เธรดเดียวหรือ อยู่ในสภาพแวดล้อมที่มีหลายเธรด สำหรับคำตอบก็คือแต่ละเธรดจะต้องประกาศว่าตัวมันเป็น Single – Thread หรือ Multi-Thread โดยดูจากว่า เธรดเหล่านั้นทำงานอยู่ในอาร์ทเมนต์ที่เป็น STA หรือ MTA นั่นเอง

### 11.12 วิธีการที่ออบเจกต์จะบอกว่าเป็น Thread-Safe

ในเรื่อง Apartment ที่ได้พูดกล่าว ชนิดของ Apartment และ ค่าต่าง ๆ ของคีย์ เธรดโมเดล ซึ่งจะพบว่าชนิด STA ซึ่งจะมีคีย์ เธรดโมเดล เป็น Single และ Apartment นั้นไม่เป็น Thread-Safe ดังนั้นจึงไม่สามารถนำออบเจกต์ซึ่งอยู่ในอาร์ทเมนต์ ชนิดนี้ไปใช้ในลักษณะมัลติเธรดได้ ซึ่งค่าของคีย์นี้ สามารถดูได้จาก registry ของระบบซึ่งได้กล่าวไปแล้วในหัวข้ออาร์ทเมนต์หาก COM คลาสไม่ระบุค่า เธรดโมเดลจะกำหนดโดยอัตโนมัติให้เป็น Single

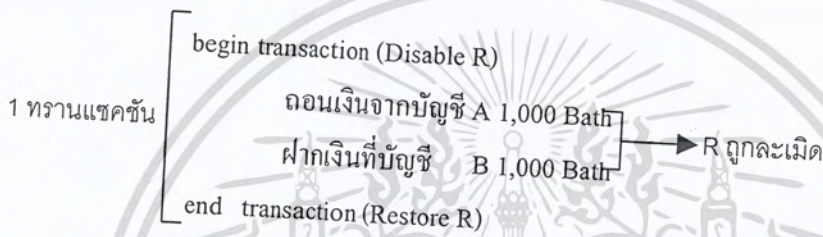


### บทที่ 12

### ทรานแซกชัน

ทรานแซกชัน คือ หน่วยของงาน หรือกลุ่มคำสั่งระดับลอจิกอล เช่น SQL โดยในกลุ่มของคำสั่งเหล่านี้อนุญาตให้มีการละเมิด Integrity Constraints (Integrity Rule) ได้ ตัวอย่างเช่น การโอนเงินประกอบด้วยงานสองอย่าง คือ การถอนเงินออกจากบัญชีหนึ่งและนำไปฝากที่อีกบัญชีหนึ่ง Integrity Rule คือ “R: ใน การโอนเงินจะไม่มียอดเงินหายไปจากระบบ” ถ้าต้องการโอนเงินจากบัญชี A ไปบัญชี B เป็นเงิน 1,000 บาท ทำได้ดังนี้

A  $\xrightarrow{1,000 \text{ Baht}}$  B



จะเห็นว่าในขณะที่ทำการถอนหรือฝาก Integrity Rule จะถูกละเมิด จนกว่าจะทำงานทั้งสองอย่างเสร็จสิ้น กลุ่มของงานดังกล่าวจึงเป็น 1 ทรานแซกชัน

#### 12.1 คุณสมบัติของทรานแซกชัน

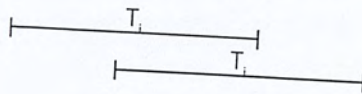
เป็นที่รู้จักกันในชื่อคุณสมบัติ ACID ประกอบด้วย

A (Atomic) - ทรานแซกชันต้องถูกมองเป็นหน่วยเดียว แบ่งย่อยไม่ได้ (ถ้าระบบทำงานล้มเหลว ต้องยกเลิกทั้งทรานแซกชัน ถ้าสำเร็จต้องสำเร็จหมดทั้งทรานแซกชัน)

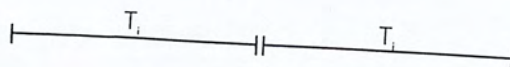
C (Correctness หรือ Consistency) - ทรานแซกชันต้องรักษาความถูกต้องตาม Integrity Rule เมื่อเสร็จทรานแซกชันแล้ว

I (Isolation) - ถ้ามีทรานแซกชันอื่นปฏิบัติพร้อมๆกับทรานแซกชันของเราในช่วงเวลาเดียวกัน ทรานแซกชันของเราต้องไม่ได้รับผลกระทบจากทรานแซกชันอื่นๆเหล่านั้น

เช่น



ต้องทำงานเหมือนกับ



หรือ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**D (Durability)** - เมื่อมีทรานแซกชันสำเร็จ (commit) ทรานแซกชันนั้นต้องอยู่ถาวรเช่น เมื่อทรานแซกชันเสร็จแล้ว system fail เมื่อ recovery กลับมาต้องเห็นทรานแซกชันอยู่

ระบบจัดการฐานข้อมูล (Database Management System, DBMS) จะมีโครงสร้างภายในที่สนับสนุนการเริ่มต้น, คอมมิต และ โรลแบ็กทรานแซกชัน ขั้นตอนพื้นฐานในการทำทรานแซกชันแสดงได้ดังนี้

```
try {
    Transaction.Begin
    ถอนเงิน 1000 บาท จากบัญชี A
    ฝากเงิน 1000 บาท จากบัญชี B
    Transaction.Commit
}
catch (Exception) {
    // ถ้าเกิดเหตุการณ์ผิดพลาด ให้โรลแบ็กการทำงานทั้งหมด
    Transaction.Rollback
}
```

เริ่มต้นทรานแซกชันโดยใช้ฟังก์ชัน Begin เมื่อทำงานเสร็จสมบูรณ์จึงใช้ฟังก์ชัน Commit เพื่อคอมมิต ทรานแซกชัน แต่ถ้าเกิดผิดพลาดระหว่างการทำงานเราสามารถใส่ฟังก์ชัน Rollback เพื่อยกเลิกการทำงานทั้งหมดนับตั้งแต่เรียกใช้ฟังก์ชัน Begin จนถึงตำแหน่งที่เกิดความผิดพลาด

## 12.2 ทรานแซกชันแบบกระจาย (Distribute Transaction)

ในกรณีที่ข้อมูลถูกจัดเก็บอยู่ในดาตาเบสตัวเดียว เราสามารถจัดการทรานแซกชันโดยใช้ฟังก์ชันของระบบจัดการฐานข้อมูลได้ แต่โดยทั่วไปแล้วข้อมูลต่าง ๆ ถูกจัดเก็บไว้ที่ดาตาเบสซึ่งอยู่ในหลาย ๆ เซิร์ฟเวอร์ หากการทำทรานแซกชันเสร็จสมบูรณ์ รีซอร์สเมเนเจอร์ทั้งหมดจะต้องคอมมิตส่วนของทรานแซกชันที่ตัวเองทำอยู่ หากไม่สามารถคอมมิตได้ก็จะต้องโรลแบ็กทั้งหมดเช่นกัน

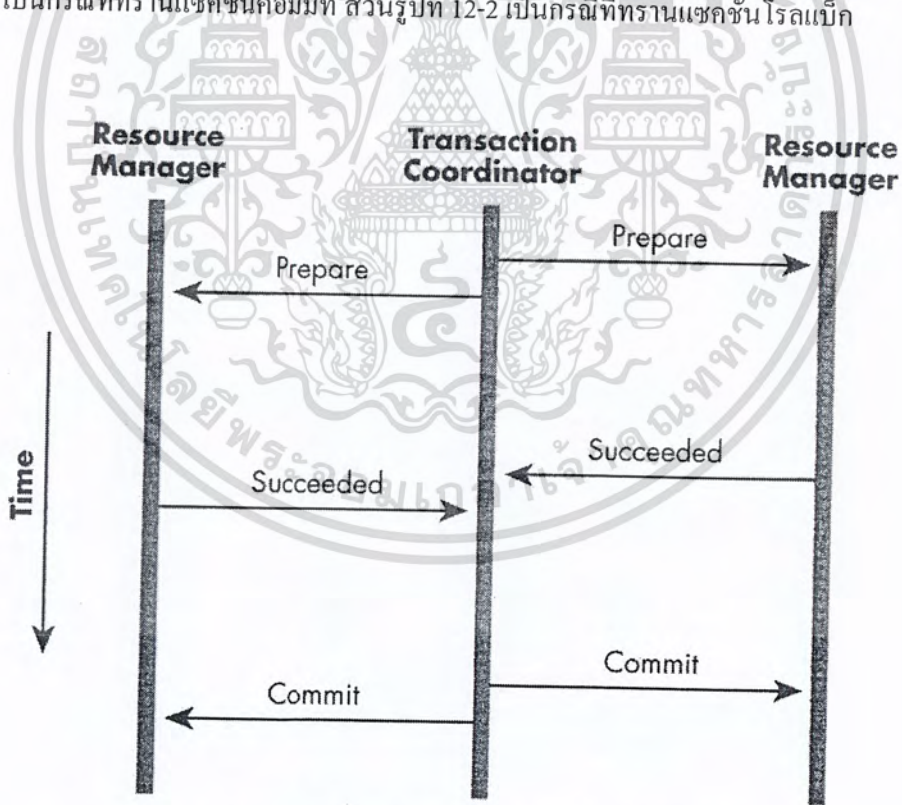
รีซอร์สเมเนเจอร์เป็นชื่อในเทอมของกระบวนการทรานแซกชัน หมายถึงระบบที่ทำงานกับออบเจกต์ใด ๆ ที่มันถือครองอยู่โดยรักษาคุณสมบัติ ACID ตัวอย่างรีซอร์สเมเนเจอร์ที่เห็นได้ชัดคือ RDBMS เช่น SQL Server หรือ Oracle

## 12.3 โพรโตคอลแบบทู-เฟสคอมมิต (2-Phase Commit Protocol)

สิ่งสำคัญในการทำทรานแซกชันแบบกระจายคือ โพรโตคอลแบบทู-เฟสคอมมิต ในโพรโตคอลนี้กิจกรรมต่างๆของรีซอร์สเมเนเจอร์จะถูกควบคุมโดยส่วนของซอฟต์แวร์ซึ่งเรียกว่า ทรานแซกชันเมเนเจอร์ (Transaction Manager) หรือ ทรานแซกชันโคออร์ดิเนเตอร์ (Transaction Coordinator) ขั้นตอนของโพรโตคอลมีดังนี้

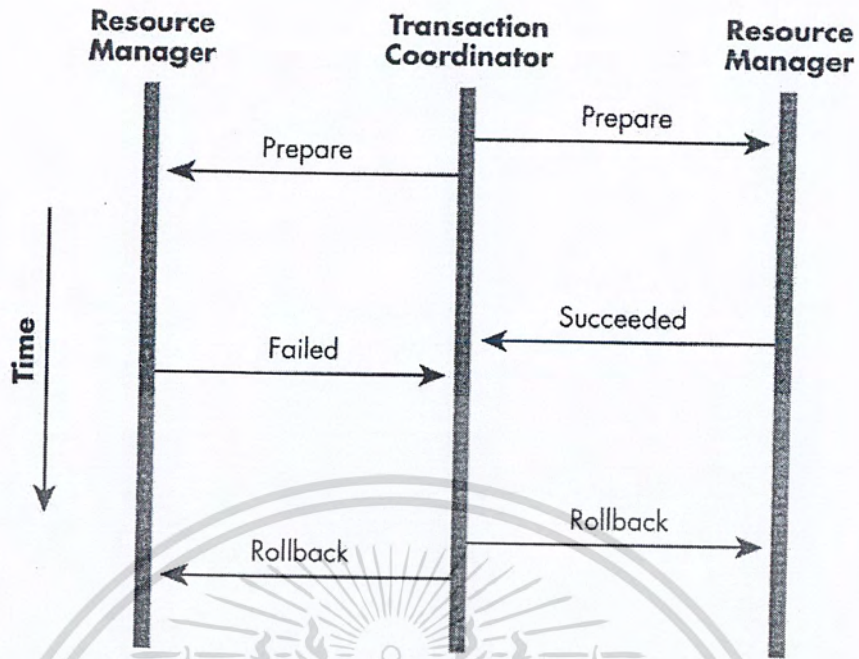
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แอปพลิเคชันเรียกใช้มธคคอมมิทในทรานแซกชันโคออร์ดิเนเตอร์
2. ทรานแซกชันโคออร์ดิเนเตอร์ติดต่อกับริชอร์สมเนเจอร์แต่ละตัวเพื่อให้จัดการทรานแซกชันและบอกให้เตรียมคอมมิททรานแซกชัน (เป็นจุดเริ่มต้นของเฟสที่หนึ่ง)
3. ริชอร์สมเนเจอร์แต่ละตัวจะวางตัวอยู่ในสถานะที่จะการันตีกับทรานแซกชันโคออร์ดิเนเตอร์ว่าจะคอมมิทหรือโรลแบ็กทรานแซกชัน โดยริชอร์สมเนเจอร์ทั้งหมดจะส่งไฟล์ที่ระบุสิ่งที่มันจะปฏิบัติในกรณีที่พร้อมจะคอมมิท แต่ถ้าไม่สามารถคอมมิทได้มันจะตอบสนองกับทรานแซกชันโคออร์ดิเนเตอร์ว่าไม่สามารถคอมมิทได้
4. ทรานแซกชันโคออร์ดิเนเตอร์ รวบรวมการตอบสนองทั้งหมดของริชอร์สมเนเจอร์
5. เฟสที่สอง ทรานแซกชันโคออร์ดิเนเตอร์แจ้งผลลัพธ์ไปยังริชอร์สมเนเจอร์แต่ละตัวเกี่ยวกับทรานแซกชันนั้น โดยถ้าในเฟสที่หนึ่งหากมีริชอร์สมเนเจอร์ตัวใดตัวหนึ่งส่งคำสั่งตอบสนองว่าไม่สามารถคอมมิทได้ ทรานแซกชันโคออร์ดิเนเตอร์ก็จะส่งคำสั่งโรลแบ็กไปยังริชอร์สมเนเจอร์ทั้งหมดให้โรลแบ็กทรานแซกชันนั้น แต่ถ้าริชอร์สมเนเจอร์ทั้งหมดตอบสนองในเฟสที่หนึ่งว่าสามารถคอมมิทได้ ทรานแซกชันโคออร์ดิเนเตอร์ก็จะส่งคำสั่งคอมมิทไปยังริชอร์สมเนเจอร์ทุกตัว เพื่อให้คอมมิททรานแซกชัน
6. โพรโตคอลแบบทู-เฟสคอมมิท แสดงเป็นแผนภาพลำดับชั้นได้ดังรูปที่ 12-1 และ 12-2 โดยรูปที่ 12-1 เป็นกรณีที่ทรานแซกชันคอมมิท ส่วนรูปที่ 12-2 เป็นกรณีที่ทรานแซกชันโรลแบ็ก



รูปที่ 12-1 การคอมมิททรานแซกชันโดยใช้โพรโตคอลทู-เฟสคอมมิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12-2 การโรลแบ็กทรานแซกชันโดยใช้โปรโตคอลทู-เฟสคอมมิต

#### 12.4 ทรานแซกชันกับ COM+

COM+ สนับสนุนการทำทรานแซกชันแบบกระจายสำหรับ COM คอมโพเนนต์ โดยเราไม่ต้องเขียนโค้ดที่ซับซ้อน กระบวนการต่างๆ เป็นดังนี้

1. ปรับแต่งออบเจกต์ที่ทำงานภายใต้ COM+
2. ตั้งค่าแอตทริบิวต์ของออบเจกต์ว่าจะให้เป็นแบบ Requires a Transaction, Requires a New Transaction หรือ Supports Transactions
3. ใช้เมธอด SetComplete ของออบเจกต์คอนเท็กซ์ (Object Context) เมื่อพร้อมที่จะคอมมิตทรานแซกชัน
4. ใช้เมธอด SetAbort ของออบเจกต์คอนเท็กซ์ (Object Context) เมื่อพร้อมที่จะโรลแบ็กทรานแซกชัน

ตารางที่ 12-1 สรุปความหมายของค่าต่างๆ ของทรานแซกชันแอตทริบิวต์ทั้งหมดของ COM+ ออบเจกต์ดังนี้

ทรานแซกชันแอตทริบิวต์	คำอธิบาย
Requires a Transaction	ออบเจกต์นี้จะต้องทำงานภายใต้ทรานแซกชัน COM+ จะเริ่มต้นทรานแซกชันใหม่สำหรับออบเจกต์นี้ก็ต่อเมื่อ ไม่มีออบเจกต์อื่น ๆ ที่อยู่ในคอนเท็กซ์เดียวกัน เริ่มต้นทรานแซกชันเลย
Requires a New Transaction	ออบเจกต์นี้จะต้องทำงานภายใต้ทรานแซกชัน COM+ จะเริ่มต้นทรานแซกชันใหม่ให้เมื่อออบเจกต์นี้ถูกสร้างขึ้น
Supports Transaction	ออบเจกต์นี้จะทำงานภายใต้ทรานแซกชันก็ต่อเมื่อ ผู้สร้าง (Creator) ได้สร้างออบเจกต์ที่เริ่มต้นทรานแซกชันไว้ก่อนแล้ว ไม่เช่นนั้นออบเจกต์จะทำงานโดยปราศจากทรานแซกชัน
Does Not Support Transactions	ออบเจกต์นี้ไม่ต้องทำงานแบบทรานแซกชัน และ COM+ จะไม่รับออบเจกต์นี้ภายใต้ทรานแซกชันใหม่ (เป็นค่าดีฟอลต์)
Disabled	ออบเจกต์ไม่สนใจที่จะเลือกค่าใดๆ ในทรานแซกชันแอตทริบิวต์ ตัวเลือกนี้ทำให้คอมโพเนนต์มีลักษณะเป็นคอมโพเนนต์ที่ยังไม่ได้ตั้งค่าอะไรเลย

ตารางที่ 12-1 ค่าทรานแซกชันแอตทริบิวต์

### การเริ่มต้นทรานแซกชัน

หลังจากที่ตั้งค่าทรานแซกชันให้กับคอมโพเนนต์ COM+ รันไทม์จะอ่านค่าทรานแซกชันแอตทริบิวต์เมื่อเริ่มสร้างออบเจกต์ หลังจากนั้นจะติดต่อกับทรานแซกชันโคออร์ดิเนเตอร์ที่เรียกว่า Distribute Transaction Coordinator (DTC) โดยจะบอกให้ DTC เริ่มต้นทรานแซกชันสำหรับออบเจกต์นี้ DTC จะเริ่มต้นทรานแซกชันให้กับออบเจกต์แล้วส่งค่า GUID ของทรานแซกชันกลับมา (DTC จะกำหนด GUID ให้กับทุกทรานแซกชัน) GUID นี้จะเก็บอยู่ในคอนเท็กซ์ของออบเจกต์นั้น เราสามารถใช้เมธอด IsInTransaction และ GetTransactionId ของอินเทอร์เฟซ IObjectContextInfo เพื่อดูว่าออบเจกต์นี้ทำงานอยู่ภายใต้ทรานแซกชันหรือไม่ และมี GUID ทรานแซกชันอะไร

ออบเจกต์ที่เป็นตัวเริ่มต้นทรานแซกชันจะเรียกว่าออบเจกต์ราก (Root Object) ของทรานแซกชัน ออบเจกต์รากมีความสำคัญมากเนื่องจากว่า COM+ รันไทม์จะไม่คอมมิททรานแซกชัน จนกว่าออบเจกต์รากจะถูกดีแอกติเวท (Deactivated) ออบเจกต์อื่น ๆ ที่อยู่ภายใต้ทรานแซกชันเดียวกันจะเป็นเพียงตัวลงความเห็นว่าจะคอมมิททรานแซกชันหรือไม่ แต่ออบเจกต์รากเป็นตัวกำหนดว่าเมื่อไหร่ที่จะคอมมิททรานแซกชัน

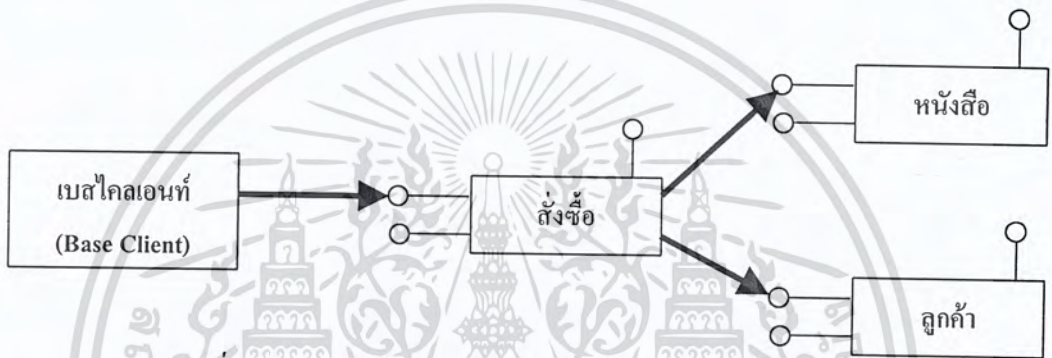
เมื่อ COM+ ออบเจกต์ที่อยู่ภายใต้ทรานแซกชันทำการติดต่อกับรีซอร์สเมเนเจอร์ รีซอร์สเมเนเจอร์นั้นก็จะอยู่ในทรานแซกชันด้วย รีซอร์สเมเนเจอร์จะติดต่อกับ DTC เพื่อบอกว่าขณะนี้มันเป็นส่วนหนึ่งของทรานแซกชัน ถ้าออบเจกต์ติดต่อกับรีซอร์สเมเนเจอร์หลายตัว รีซอร์สเมเนเจอร์เหล่านั้นทั้งหมดก็จะอยู่ในทรานแซกชันเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด SetComplete เป็นการ โหวตเพื่อคอมมิททรานแซกชัน ออบเจ็กต์รากเป็นตัวบ่งบอกว่าขณะนี้พร้อมคอมมิทแล้ว DTC จะทำงานโดยใช้โปรโตคอลทวู-เฟสคอมมิท เพื่อคอมมิทหรือโรลแบ็กทรานแซกชันตามขั้นตอนที่ได้กล่าวมาแล้วข้างต้น

### 12.5 ทรานแซกชันที่เกี่ยวข้องกับออบเจ็กต์มากกว่า 1 ตัว

ก่อนหน้านี้เราได้กล่าวถึง COM+ ออบเจ็กต์ตัวเดียวที่ทำงานในลักษณะที่เป็นทรานแซกชันและมีการติดต่อกับรีเซอร์ทเมเนเจอร์ตั้งแต่สองตัวขึ้นไป แต่ในหลาย ๆ กรณีทรานแซกชันมักจะเกี่ยวข้องหรือใช้ออบเจ็กต์หลาย ๆ ตัว ตัวอย่างเช่นการสั่งหนังสือออนไลน์ สมมติว่าให้ออบเจ็กต์การสั่งซื้อเป็นออบเจ็กต์รากของทรานแซกชัน ออบเจ็กต์การสั่งซื้อมีการสร้างและเรียกใช้ออบเจ็กต์ลูกค้ำกับออบเจ็กต์หนังสือดังรูปที่ 12-3



รูปที่ 12-3 ออบเจ็กต์หลาย ๆ ตัวทำงานอยู่ในทรานแซกชันเดียวกัน

เมื่อ COM+ ออบเจ็กต์ถูกสร้างโดย COM+ ออบเจ็กต์ด้วยกัน ออบเจ็กต์ที่ถูกสร้างขึ้นใหม่จะมีลักษณะการทำงานแบ่งได้เป็น 3 ทางคือ (1) อยู่ในทรานแซกชันเดียวกับออบเจ็กต์ที่สร้างมัน (2) สร้างทรานแซกชันของตนเองขึ้นมาใหม่ (3) ไม่ทำงานเป็นทรานแซกชันเลย กรณีทั้งสามจะเกิดขึ้นอยู่กับการตั้งค่าทรานแซกชันแอตทริบิวต์ให้กับออบเจ็กต์ต่าง ๆ

ถ้าออบเจ็กต์สั่งซื้อและออบเจ็กต์ลูกค้ำต่างมีทรานแซกชันเป็นของตัวเอง ออบเจ็กต์สั่งซื้ออาจคอมมิทงานของตน โดยที่ออบเจ็กต์ลูกค้ำอาจทำงานแล้วเกิดปัญหาทำให้ต้องโรลแบ็กงานของตน จะเห็นได้ว่าออบเจ็กต์ทั้งสองอาจทำงานขัดแย้งกันได้ซึ่งเป็นเหตุการณ์ที่เราไม่ต้องการ หากออบเจ็กต์ทั้งคู่ทำงานอยู่ในทรานแซกชันเดียวกัน การคอมมิทหรือโรลแบ็กทรานแซกชันจะขึ้นอยู่กับออบเจ็กต์ทั้งหมดที่อยู่ในทรานแซกชันนั้น ออบเจ็กต์แต่ละตัวจะเสนอการคอมมิทโดยเรียกใช้เมธอด SetComplete ของอินเทอร์เฟซ IObjectContext หรืออาจเสนอให้โรลแบ็กโดยใช้เมธอด SetAbort ของอินเทอร์เฟซ IObjectContext เช่นเดียวกัน ถ้าออบเจ็กต์ทุกตัวเสนอให้คอมมิท ทรานแซกชันจึงจะคอมมิท หากมีตัวใดตัวหนึ่งเสนอให้โรลแบ็ก ทรานแซกชันจะต้องโรลแบ็ก สิ่งสำคัญประการหนึ่งคือ การเรียกใช้เมธอด SetComplete เป็นเพียงการเสนอว่าออบเจ็กต์นั้นยอมรับและพร้อมคอมมิท ทรานแซกชัน ออบเจ็กต์จะยังไม่ทราบว่าทรานแซกชันคอมมิทแล้วจริง ๆ หรือไม่ ดังนั้น COM+ ออบเจ็กต์ที่ทำงานในลักษณะที่เป็นทรานแซกชันไม่ควรสรุปว่าสถานะของตนหลังจากใช้เมธอด SetComplete ไปแล้วจะเป็นสถานะสุดท้ายที่ถูกต้องเสมอไปเมื่อทรานแซกชันคอมมิท ตัวอย่างเช่น ถ้าออบเจ็กต์ลูกค้ำทำงานโดยหักเงินของบัญชีลูกค้ำในฐานข้อมูลด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจกต์ลูกค้ายังไม่ควรเปลี่ยนแปลงสถานะยอดเงินจนกว่าทรานแซคชันจะคอมมิต เพราะหากเปลี่ยนแปลงแล้วออบเจกต์อื่น ๆ ที่ทำงานในทรานแซคชันเดียวเสนอให้โรลแบ็กทรานแซคชัน จะทำให้ออบเจกต์ลูกค้ายู่ในสถานะที่ไม่ถูกต้องโดยเงินในบัญชีลูกค้ายถูกหักออกไปโดยที่การสั่งหนังสือทำงานไม่สำเร็จ ในการป้องกันปัญหาดังกล่าว COM+ จะดีแอกติเวทออบเจกต์ทั้งหมดที่เกี่ยวข้องกับทรานแซคชันเมื่อทรานแซคชันคอมมิต การดีแอกติเวทออบเจกต์เป็นการเพิ่ม scalability ของระบบ แต่เหตุผลของในเรื่องของ ทรานแซคชันคือเพื่อความความถูกต้องของงานที่ปฏิบัติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 13

### Scalability

Scalability ซึ่งในที่นี้หมายถึงความสามารถที่แอปพลิเคชันสามารถจัดการกับงานที่มีจำนวนมากขึ้นเนื่องจากในระบบงานที่ต้องมีการติดต่อกับไคลเอนต์เป็นจำนวนมาก ทุก ๆ ไคลเอนต์ก็จะมีออบเจกต์ของมันเอง และจะถือครองออบเจกต์เหล่านั้นตลอดเวลา เมื่อมีไคลเอนต์เป็นจำนวนมากๆ จึงต้องเสียทรัพยากรของเซิร์ฟเวอร์เป็นจำนวนมากสำหรับออบเจกต์เหล่านั้นตลอดเวลา และจะทำให้ถึงขีดจำกัดของเซิร์ฟเวอร์ได้ง่ายซึ่งทำให้ไม่สามารถสร้างระบบงานที่มีการติดต่อกับไคลเอนต์ที่มีจำนวนมากขึ้นได้ นอกจากนั้นการที่ครองออบเจกต์ตลอดเวลา นั้นไม่คุ้มค่าเนื่องจากไคลเอนต์นั้นจะเรียกใช้ออบเจกต์เพียงช่วงเวลาสั้น

ในการแก้ปัญหาที่เกิดขึ้นนั้น COM+ ได้เตรียมบริการ ไว้สองตัวคือ JIT (Just-In Time) Activation และ Object Pooling ซึ่งจะร่วมกันในการเพิ่มความสามารถในการขยายขนาด (Scalability)

#### 13.1 JIT Activation

จากที่ได้กล่าวมาแล้วว่าในระบบงานที่มีการติดต่อกับไคลเอนต์นั้น ตัวไคลเอนต์จะทำการสร้างออบเจกต์ไว้ตั้งแต่เริ่มโปรแกรม และจะครองไว้จนกว่าจะจบการทำงาน สำหรับในเหตุผลก็คือ ในการสร้างออบเจกต์บ่อยถือเป็นการทำงานที่เปลืองทรัพยากรโดยไม่จำเป็นเมื่อนึกถึง ไคลเอนต์ที่อยู่ต่างเครื่องยังต้องเสียเวลาติดต่อผ่านเน็ตเวิร์คและการทำงานอย่างอื่นอีกมากกว่าที่จะสามารถสร้างออบเจกต์ไว้ที่ฝั่งเซิร์ฟเวอร์ได้ อย่างไรก็ตามสิ่งที่ได้กล่าวมาแล้วว่าการถือครองออบเจกต์ไว้ตลอดเวลา ก็ไม่เป็นผลดีเช่นกัน ดังนั้นการแก้ปัญหา ก็คือให้หลักการที่จะให้ไคลเอนต์นั้นถือครองอินสแตนซ์ของออบเจกต์เมื่อเวลาที่มันต้องการเรียกใช้งานเท่านั้นและ รีบคืนออบเจกต์นั้นให้กับระบบโดยเร็วที่สุดหลังจากที่ทรานแซกชันนั้นคอมมิต (Commit) หรือ ไม่ก็โรลแบ็ค (Roll back)

จากหลักการที่ได้กล่าวมาแล้วนั้นเป็นวิธีที่อยากที่จะให้ผู้พัฒนา โปรแกรมเขียน โค้ดเพื่อทำงานนี้ให้ใช้งานง่ายดังนั้น COM+ จึงได้จัดเตรียมฟังก์ชันเหล่านี้ไว้ให้โดยเรียกว่า JIT (Just-In Time) Activation โดยที่สามารถควบคุมได้โดยเป็นลักษณะแอททริบิวต์เช่นเดียวกับบริการอื่นๆ ของ COM+ Runtime โดยมีขั้นตอนการทำงานคือ

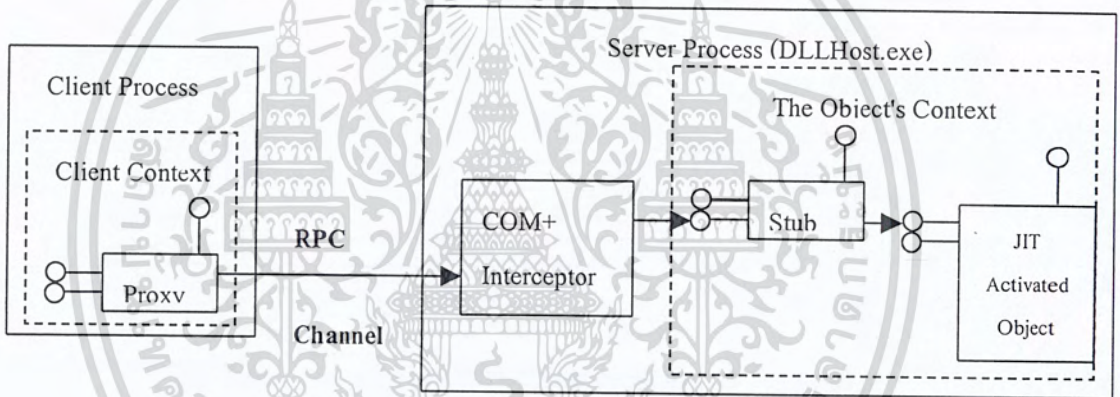
เมื่อมีการสร้างออบเจกต์ขึ้นใหม่ COM+ Runtime จะสร้าง Proxy และ Stub มาตรฐานพร้อมด้วย RPC Channel ระหว่าง ออบเจกต์และไคลเอนต์ของมันดังรูปที่ 13-1

เมื่อออบเจกต์เซตค่าบิต Deactivate-on-return ของมัน COM+ Runtime จะทำการ Deactivate ตัวออบเจกต์นั้น ซึ่งออบเจกต์นั้นจะถูกทำลาย หรือ ไม่ก็จะถูกนำกลับไปไว้ที่ Object Pool (ขึ้นกับค่าที่ถูกตั้งไว้ของ แอททริบิวต์ Object Pooling) แต่ตัวไคลเอนต์ก็ยังคงถือครอง Proxy ไปยังตัวออบเจกต์อยู่ RPC และ Stub ก็ยังคงอยู่ ดังรูปที่ 13-2 ในขณะที่ตัวไคลเอนต์จะยังคงรับรู้ว่าตัวออบเจกต์ยังอยู่ จนกระทั่งเมื่อไคลเอนต์จะเรียกใช้เมธอดที่ของออบเจกต์อีกครั้งหนึ่ง ตัว Interceptor ที่อยู่ระหว่าง RPC Channel และ

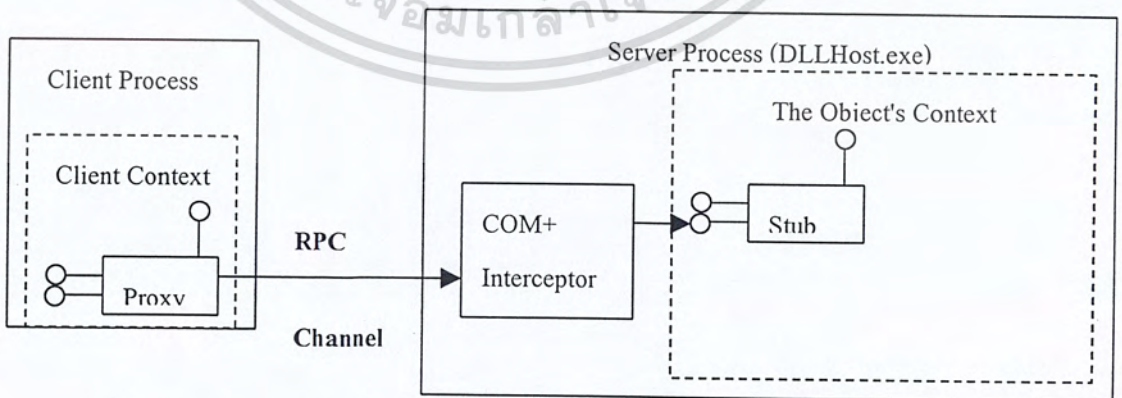
Sub จะทำการสร้าง อินสแตนซ์ใหม่อื่นของออบเจ็กต์เดิม (หรือ คึงอินสแตนซ์นั้นมาจาก Object Pool) และนำมาเชื่อมเข้ากับ Stub และใช้งานอย่างปกติ

นอกจากจะช่วยลดจำนวนออบเจ็กต์ที่จะต้อง Activate ในแต่ละเวลาแล้ว สำหรับออบเจ็กต์ทุกๆ ตัวที่มีลักษณะเป็นทรานแซคชันของ COM+ นั้นต้องใช้ JIT Activation ซึ่งจะช่วยบังคับให้ทุก ๆ ทรานแซคชัน นั้นมีความถูกต้อง เราสามารถเซต JIT Activation ได้ในระดับคลาส โดยใช้ Component Services Explorer โดยมีขั้นตอนดังนี้

1. คลิกขวาที่คอมโพเนนต์ใน โพลเดอร์ Components ของแอปพลิเคชัน
2. เลือก Properties... จาก เมนู Context (ไดอะล็อก Component Properties จะแสดงดังรูปที่ 13-3)
3. เลือกหน้า Activation
4. เซตเช็คบ็อกซ์ Enable Just In Time Activation
5. กด OK



รูปที่ 13-1 แสดงออบเจ็กต์ที่สนับสนุน JIT Activation ในสถานะ Activated



รูปที่ 13-2 แสดงออบเจ็กต์ที่สนับสนุน JIT Activation ในสถานะ Deactivated

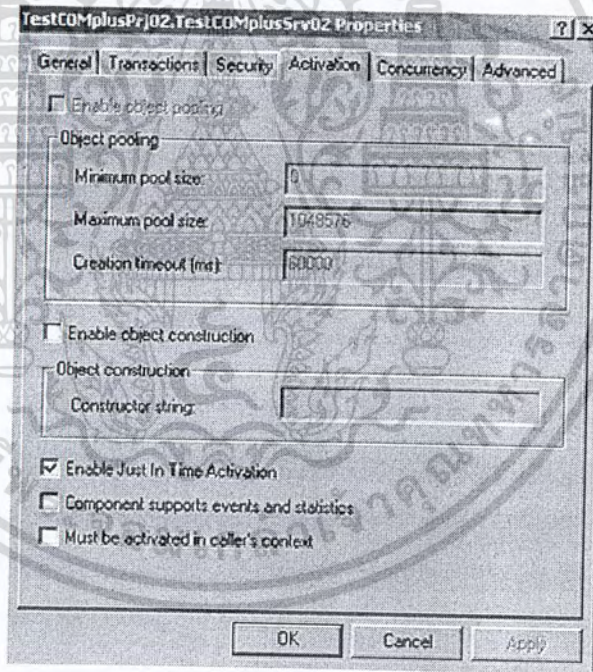
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 13.2 Object Pooling

สำหรับแอปพลิเคชันที่ใช้ JIT Activation ในการ Activate และ Deactivate ออบเจกต์เป็นจำนวนมาก ๆ นั้นค่อนข้างจะเป็นวิธีที่สิ้นเปลืองที่จะต้องสร้างออบเจกต์ใหม่ ทุก ๆ ครั้งที่มีการเริ่มต้น ทรานแซกชัน Object Pool เป็นวิธีการที่ถูกนำมาใช้ในการแก้ปัญหาดังกล่าวโดยเมื่อไคลเอนต์จะทำการ Activate ออบเจกต์นั้น COM+ Runtime จะส่งอินสแตนซ์จาก Object Pool มาให้ถ้าหากมีออบเจกต์ชนิดนั้นอยู่ใน Object Pool แต่หากไม่มีก็จะทำการสร้างอินสแตนซ์ใหม่ขึ้นมา ไคลเอนต์สามารถใช้ออบเจกต์ได้นานเท่าที่มันต้องการและเมื่อไคลเอนต์ปล่อยออบเจกต์นั้น ออบเจกต์ก็จะไม่ถูกทำลายแต่จะถูกนำกลับไปไว้ที่ Object Pool เพื่อรอการเรียกใช้ครั้งต่อไป ซึ่งจะมีลักษณะดังรูปที่ 13-4

สำหรับการที่จะทำให้ออบเจกต์มีความสามารถที่เป็นออบเจกต์ที่สามารถใช้ใน Object Pool ได้นั้นก็สามารถจะเซตแอททริบิวต์ได้เหมือนกับบริการอื่น ๆ ของ COM+ Runtime โดยมีขั้นตอนดังนี้

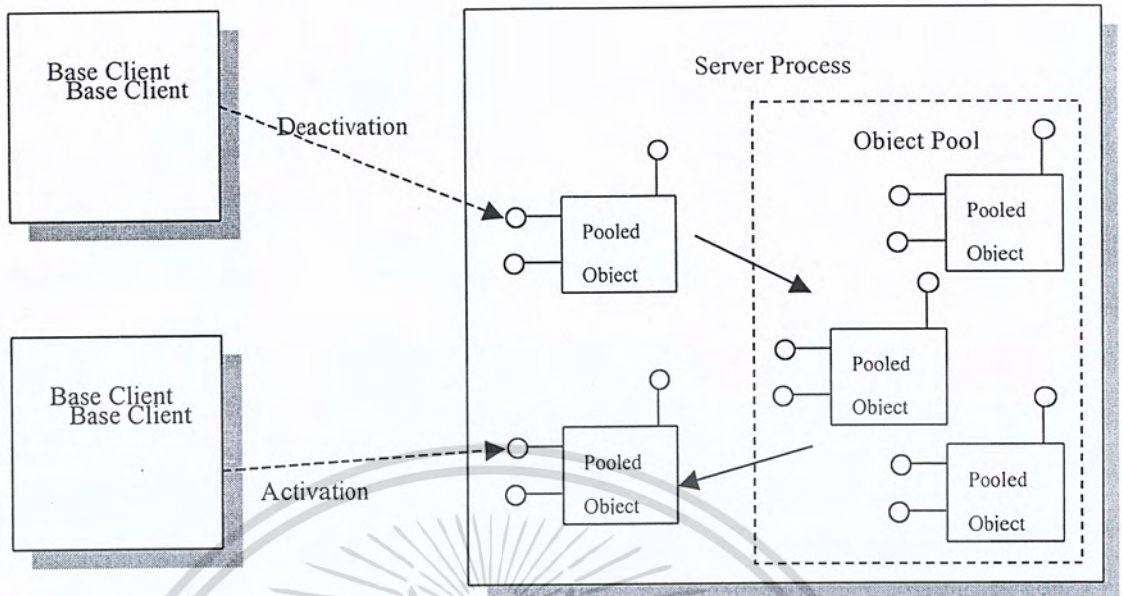
1. คลิกขวาที่คอมโพเนนต์ในโฟลเดอร์ Components ของแอปพลิเคชัน
2. เลือก Properties... จากเมนู Context ( ในไดอะล็อก Component Properties จะปรากฏดังรูปที่ 13-3)



รูปที่ 13-3 ไดอะล็อก Component Properties

3. เลือกหน้า Activation
4. เซตเชคบ็อกซ์ Enable object pooling
5. กด OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13-4 แสดง Object Pooling

จากวิธีที่ได้กล่าวมานั้นเป็นวิธีที่จะเซตให้ออบเจ็กต์สามารถนำมาใช้ร่วมกัน (Pool) ได้แต่ก่อนที่เซตค่านั้นได้ตัวออบเจ็กต์จะต้องมีลักษณะดังนี้ด้วยคือ

- ต้องอยู่ในอาร์ทเมนต์ชนิด Thread Neutral หรือ Free-Thread
- ต้องไม่มีสถานะ
- ต้องไม่มี Thread-affinity
- ต้องสามารถ รวมกันได้ (aggregateable)

นอกจากจะเซตให้ออบเจ็กต์นั้นสามารถใช้งานร่วมกันได้แล้วยังสามารถกำหนดจำนวนอินสแตนซ์ของออบเจ็กต์ที่ต้องมีอย่างน้อยที่สุด และมากที่สุดเท่าที่จะมีได้ โดยหากออบเจ็กต์ที่มีอยู่ใน Object Pool ยังน้อยกว่าจำนวนที่ต้องมีอย่างน้อยที่สุด เมื่อมีการเรียกใช้จากไคลเอนต์ตัวใหม่ COM+ Runtime ก็ จะสร้างอินสแตนซ์ใหม่ให้ทันที แต่หากมีอยู่เกินกว่าจำนวนที่น้อยที่สุดแล้วแต่ถูกใช้อยู่ทุกตัว และยังมีจำนวนที่น้อยกว่าจำนวนที่มากที่สุดเท่าที่จะมีได้แล้ว ก็จะสร้างอินสแตนซ์ใหม่ให้กับไคลเอนต์นั้น แต่ถ้าเกินขอบเขตที่มากที่สุดแล้ว ไคลเอนต์จะต้องรอนจนกว่าจะมีออบเจ็กต์ว่าง แล้วจึงนำเอาออบเจ็กต์นั้นไปใช้งานได้

สำหรับ COM+ ออบเจ็กต์ที่ต้องการแสดงสถานะว่าสามารถที่จะถูกนำกลับไป Pool ได้ หรือให้ทำการ Activate และ Deactivate เองต้องทำการอิมพลีเมนต์อินเตอร์เฟซ IObjectControl โดยมีเมธอด Activate และ Deactivate ที่ใช้ในการ Activate และ Deactivate ออบเจ็กต์ตามลำดับ และ เมธอด CanBePooled ที่ใช้ในการบอก COM+ Runtime ว่าสามารถนำกลับไปใช้ได้หรือไม่ในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 13.3 Synchronization

ในหัวข้อ COM+ Thread ได้กล่าวถึงคุณสมบัติ และข้อเด่นของ เทรดโมเดล ชนิด STA และ MTA โดยที่ MTA นั้นจะทำงานได้อย่างมีประสิทธิภาพในสภาพแวดล้อมที่เป็น Multi-Thread แต่ตัวออบเจกต์นั้นจะต้องมีลักษณะเป็น Thread Safe ถ้าหากออบเจกต์นั้นไม่เป็น Thread-Safe แล้วจะต้องทำงานภายใน STA

นอกจากนั้น STA ยังไม่เหมาะกับ Object Pool เนื่องจากมันจะจزمว่ามีเพียงเทรด เทรดเดียวเท่านั้นที่จะใช้งานมันได้ดังนั้นถึงแม้ว่าจะนำออบเจกต์นี้ไปไว้ใน Object Pool ก็ไม่สามารถถูกเรียกใช้จากออบเจกต์ตัวอื่น ๆ ได้และจะต้องสร้างอินสแตนซ์ใหม่ให้กับแต่ละไคลเอนต์ทุกครั้งอยู่ที่

TNA นั้นเป็นวิธีที่ถูกนำมาใช้ในการแก้ปัญหาสำหรับทั้ง MTA และ STA เพื่อให้ COM+ Runtime สามารถทำงานได้อย่างมีประสิทธิภาพสูงสุดเมื่อทำงานกับ Object Pool เนื่องจากอนุญาตให้เทรดที่อยู่ในเทรดโมเดลชนิดต่าง ๆ เข้าทำงานภายในตัวมันได้ เช่นเดียวกับบริการอื่น ๆ ของ COM+ Runtime เราสามารถให้ COM+ ออบเจกต์ของเรานั้นเป็น Thread-Safe ได้โดยไม่ต้องใส่โค้ดเพิ่มเติม จากการใช้บริการ Activity-Based Synchronization Support

สำหรับ Thread-Safe สำหรับ COM+ ออบเจกต์นั้นซับซ้อนกว่าแค่การ ล็อกออบเจกต์หนึ่ง ๆ ไว้แล้วอนุญาตให้เทรดเพียงเทรดเดียวทำงานกับ เมธอดของเทรดได้เท่านั้น เนื่องจากออบเจกต์เหล่านั้นอาจจะเรียกเมธอดจากออบเจกต์ตัวอื่นซึ่งอยู่ต่างโพรเซส หรือต่างเครื่อง แล้วทำการแก้ไขหลังจากนั้นก็การยกเลิก ซึ่งอาจทำให้เกิดปัญหาในเรื่องทรานแซกชันต่าง ๆ ได้ ในการใช้ Activity-Based Synchronization Support นั้นจะช่วยป้องกันปัญหาของการทำทรานแซกชันหลาย ๆ ทรานแซกชันพร้อมกันได้ โดยใช้หลักการของ Activity

### 13.4 Activity

คือเทรดในทางจินตภาพซึ่งจะทำงานขึ้นกับยูสเซอร์เพียงยูสเซอร์เดียว และสามารถทำงานข้ามเครื่องได้ ยูสเซอร์แต่ละยูสเซอร์นั้นจะมี Activity ที่เป็นอิสระจากกัน ภายใน Activity นั้นก็สามารถมีได้หลาย ออบเจกต์ สภาพแวดล้อม COM+ รุ่นใหม่จะจัดการรายละเอียดเหล่านี้โดยการสร้าง Activity ใหม่ขึ้นมาทุก ครั้งเมื่อยูสเซอร์ทำการสร้าง COM+ ออบเจกต์ขึ้นมาใหม่

เมื่อมีการสร้าง Activity ใหม่ขึ้นมา COM+ รุ่นใหม่จะทำการสร้าง STA ใหม่ขึ้นมาด้วย แต่หากจำนวน STA ที่สร้างขึ้นมาได้ถึงขีดจำกัดของระบบแล้ว COM+ รุ่นใหม่ก็จะทำการสร้าง Activity ขึ้นมาแบ่งกันใช้งานภายใน STA เดียวกัน เมื่อมีเหตุการณ์นี้ COM+ รุ่นใหม่จะคอยจัดการ Multiplexing STA (เทรดจริง) และ การเข้าออกของ Activity (เทรดในทางจินตภาพ)

ในทางปฏิบัติเมื่อมี Activity มากกว่าหนึ่ง Activity ใช้ STA ร่วมกัน การทำงานของ ยูสเซอร์หนึ่ง ก็จะต้องบล็อกการทำงานของยูสเซอร์อื่นอีกเช่นกัน แต่เราไม่จำเป็นต้องสนใจในเรื่องนั้น เนื่องจากสิ่งที่สำคัญของวิธีการนี้คือการที่ในการทำงาน ในไม่มีปัญหาในเรื่องของการ Synchronization และมีรูปแบบ Threading Model ที่เข้าใจได้ง่าย ๆ ว่าเป็นเทรดทางจินตภาพที่ COM+ รุ่นใหม่คอยจัดการให้

อีกสิ่งที่สำคัญของการทำงานใน COM+ รันไทม์ซึ่งจัดการ Activity ให้นั้นคือการสร้างออบเจกต์ใหม่ เนื่องจาก หากมีการสร้าง COM+ ออบเจกต์ใหม่นั้นก็จะทำให้เกิดการสร้าง Activity ใหม่ซึ่งเมื่องานของยูสเซอร์ต้องเรียกใช้ออบเจกต์ผ่าน Activity ก็จะทำให้เปลืองทรัพยากรของระบบ และทำให้ความเร็วลดลง เนื่องจากต้องติดต่อบริเวณ Activity 2 Activity

ในทางปฏิบัตินั้นเราจะใช้คำสั่ง *CreateInstance* เพื่อสร้างออบเจกต์ใหม่ให้อยู่ใน Activity เดิม โดยคำสั่งนี้จะมีลักษณะการเรียกใช้งาน เป็นซอร์สโค้ดของ วิชาลเบสิก ได้ดังนี้

```
Dim ObjCtx As ObjectContext
```

```
Set ObjCtx = GetObjectContext()
```

```
Dim Object2 As CMyClass2
```

```
Set Object2 = ObjCtx.CreateInstance("MyServer.CMyClass2")
```

แต่ในกรณีที่ต้องการสร้างออบเจกต์ที่ไม่ใช่ COM+ ออบเจกต์ (ไม่ได้ติดตั้งลงบน COM+ รันไทม์ เช่น ออบเจกต์ ADO) ซึ่งออบเจกต์เหล่านั้นไม่จำเป็นต้องมีการจัดการสภาพแวดล้อมจาก COM+ รันไทม์ ซึ่งทำให้ไม่มีการสร้าง Activity ใหม่ดังนั้นจึงไม่จำเป็นต้องเรียกใช้ผ่านทางเมธอด *CreateInstance*

เราสามารถกำหนดลักษณะการสนับสนุนการ ซิงโครไนซ์ ของออบเจกต์ได้โดยใช้ Component Service Explorer โดยมีขั้นตอนดังนี้

1. คลิกขวาที่คอมโพเนนต์ในโฟลเดอร์ Components ของแอปพลิเคชัน
2. เลือก Properties... จากเมนู Context ( โดะลอก Component Properties จะปรากฏดังรูปที่ 13-3)
3. เลือกหน้า Concurrency
4. เลือกปุ่มเรดิโอสำหรับค่าที่ต้องการ
5. กด OK

- อย่างไรก็ตามออบเจกต์บางรูปแบบก็มีลักษณะการ ซิงโครไนซ์ ที่เหมาะสมและเป็นข้อบังคับดังนี้
- COM+ คอมโพเนนต์ที่ทำงานในรูปแบบทรานแซกชันต้องอยู่ใน Activity ซึ่งค่าที่จะเลือกได้จะเป็น ได้แก่ Required หรือ Required New เท่านั้น
  - STA คอมโพเนนต์ต้องอยู่ใน Activity เสมอไม่ว่ามันจะทำงานในรูปแบบทรานแซกชันหรือไม่ก็ตาม ค่าที่เลือกได้ก็คือ Required หรือ Required New
  - ออบเจกต์ที่สนับสนุน JIT Activation ไม่ว่ามันจะทำงานในรูปแบบทรานแซกชันหรือไม่ก็ตาม ต้องอยู่ใน Activity เช่นกัน และค่าที่จะเลือกได้ก็จะมีแค่ Required หรือ Required New สำหรับรายละเอียดของค่าต่าง ๆ ที่จะเซตได้จะแสดงในตารางที่ 13-1

ค่าการ ซิงโครไนซ์	รายละเอียด
Not Support	ออบเจกต์จะไม่รันใน Activity
Supported	ออบเจกต์จะรันใน Activity เดียวกับผู้สร้าง ถ้าผู้สร้างมันรันใน Activity
Required	ออบเจกต์จะรันใน Activity เดียวกับผู้สร้าง ถ้าผู้สร้างมันรันใน Activity และจะสร้าง Activity ของมันเองหากผู้สร้างไม่ได้รันใน Activity
Required New	จะทำการสร้าง Activity ของมันเองเสมอ
Disable	ไม่สนใจในการทำ ซิงโครไนซ์ ทั้งหมด

ตารางที่ 13-1 ค่าที่เป็นไปได้ทั้งหมดของการ ซิงโครไนซ์

ออบเจกต์ที่รันอยู่ใน Activity ก็จะมี Activity ID ซึ่งสามารถดูได้โดยเมธอด getActivityId ซึ่งอยู่ใน IObjectContextInfo ซึ่งสามารถอธิบายการทำงานของ Activity ได้ดังนี้ (สมมติให้ออบเจกต์มีค่าการซิงโครไนซ์ เป็น Required และผู้สร้างยังไม่ได้รันใน Activity) เมื่อออบเจกต์ถูกสร้างมันก็จะสร้าง Activity ของมันเองซึ่งแต่ละ Activity จะมี ล็อกซิงโครไนซ์ระหว่างโพรเซส (Process-Wide Synchronization Lock) เมื่อมีเรดมาขอเรียกใช้ออบเจกต์นี้ เรดนั้นก็จะนำล็อกตัวนี้ไปด้วย และจะไม่คืนจนกว่าจะเรดจะออกจากเมธอดนั้น เมื่อมีเรดอื่นมาขอเรียกใช้ออบเจกต์นี้ก็จะยังไม่สามารถใช้งานได้ หากเรดแรกนั้นยังไม่ออกจากเมธอดของออบเจกต์ เนื่องจากไม่มีล็อก

หากเรดที่อยู่ในเมธอดของออบเจกต์พยายามสร้างออบเจกต์ใหม่ออบเจกต์ใหม่จะมีลักษณะที่จะอยู่กับ Activity ในรูปแบบใดก็จะขึ้นกับคุณสมบัติในการซิงโครไนซ์ของมันเองซึ่งเป็นตามตารางที่ 13-2

ค่าการ ซิงโครไนซ์	สถานะ Activity
Not Supported	ไม่รันใน Activity
Support	รันใน Activity ของผู้สร้าง
Required	รันใน Activity ของผู้สร้าง
Required New	รันใน Activity ของมันเอง

ตารางที่ 13-2 สถานะของ Activity ของออบเจกต์ซึ่งผู้สร้างอยู่ใน Activity

หากออบเจกต์ตัวใหม่มีค่าในการซิงโครไนซ์เป็น Required มันจะทำการใช้ Activity ร่วมกันออบเจกต์แรก และเมื่อมีเรดใหม่เรียกใช้ออบเจกต์ใหม่ในขณะที่ออบเจกต์แรกทำงานอยู่ นั้นจะไม่สามารถทำได้เนื่องจาก เรดใหม่นี้ก็ต้องกรล็อกซิงโครไนซ์ระหว่างโพรเซส ซึ่งไม่มีเนื่องจากเรดที่ทำงานกับออบเจกต์แรกนั้น ใช้อยู่ ดังนั้นทั้งสองออบเจกต์ก็จะไม่สามารถทำงานได้อย่างต่อเนื่อง แต่ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจกต์มีค่าในการซิงโครไนซ์เป็น Not Support หรือ Required New ก็จะทำให้ออบเจกต์ใหม่นั้นสามารถทำงานพร้อมกับออบเจกต์แรกได้

เมื่อออบเจกต์ที่อยู่ภายใน Activity เดียวกัน แต่อยู่ต่าง โพรเซส หรือต่างเครื่องกัน มันจะไม่สามารถใช้เธรดร่วมกันได้แต่ COM+ Runtime จะคอยจัดการตัวล็อกซิงโครไนซ์ระหว่างโพรเซสนี้ให้

หากมีออบเจกต์หนึ่งเรียกเมธอดที่อยู่ในออบเจกต์หนึ่ง และ ออบเจกต์ตัวที่สองก็เรียกใช้เมธอดในออบเจกต์แรกเช่นกันออบเจกต์แรกก็จะล็อกและรอนกว่าออบเจกต์ตัวที่สองจะว่าง เช่นเดียวกับออบเจกต์ตัวที่สองที่จะล็อก และรอนกว่าออบเจกต์จะว่าง(ซึ่งจะทำให้เกิด Dead Lock) ปัญหานี้จะแก้ไขโดยใช้หลักการ Causality (ซึ่งเป็นหน้าที่ของ DCOM) Causality เป็นชื่อที่แทนกลุ่มองการเรียกใช้ฟังก์ชันที่ซ้อนกัน Causality ID (ซึ่งเป็น GUID) จะถูกส่งผ่านทาง DCOM ทุกครั้งที่มีการเรียกใช้เมธอด ออบเจกต์แรกจะตรวจสอบว่า Causality ID และพิจารณาว่าตรงกับ Causality ID ตรงกับที่มีนรออยู่หรือไม่ ออบเจกต์แรกจะทำการป้องกันไม่ให้เกิด Dead Lock โดยไม่อนุญาตให้ออบเจกต์ตัวที่สองเรียกใช้เมธอด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 14

### ความปลอดภัยบนอินเทอร์เน็ต (Internet security)

การติดต่อสื่อสารบนอินเทอร์เน็ตนั้นจะใช้ Transmission Control Protocol/Internet Protocol (TCP/IP) โดย TCP/IP นั้นอนุญาตให้สามารถส่งข้อมูลระหว่าง คอมพิวเตอร์ ได้ ทำให้เป็นไปได้ว่าจะมีมือที่สามเข้ามาก่อความวุ่นวายที่สื่อสารกันอยู่ก็เป็นได้ ลักษณะที่จะถูกมือที่สามเข้ามาก่อความวุ่นอยู่ 3 ลักษณะ ดังนี้

1. การลอบฟัง (Eavesdropping) ข้อมูลอาจถูกผู้อื่นเข้ามาลอบฟังได้ระหว่างทางที่มีการขนส่งข้อมูล เช่น มีการลอบฟังเอาหมายเลขบัตรเครดิตของเราขณะที่ทำการซื้อสินค้าผ่านอินเทอร์เน็ต
2. การลักลอบเปลี่ยนแปลงข้อมูล (Tampering) การเข้ามาเปลี่ยนแปลงข้อมูลขณะที่ทำการส่งข้อมูลทำให้ข้อมูลบิดเบือนไปทำให้การรับข้อมูลเป็นไปอย่างผิดๆ เช่น ข้อมูลการส่งสินค้าถูกเปลี่ยนแปลงจำนวนไปทำให้เกิดความเสียหายได้
3. การปลอมตัวเข้ามา (Impersonation) การปลอมตัวเข้ามาหลอกเราว่าเป็นคนๆ นั้น หรือ เว็บไซต์นั้น ทำให้เราหลงเชื่อและทำการติดต่อด้วย อาจเกิดความเสียหายได้

ดังที่ได้กล่าวมาแล้วถึงปัญหาความปลอดภัยบนอินเทอร์เน็ต จึงมีการคิดวิธีการแก้ปัญหาข้างต้น ซึ่งจะกล่าวดังต่อไปนี้

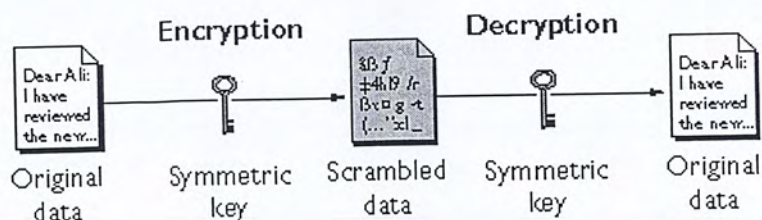
14.1 การแก้ปัญหาการลอบฟัง จะใช้การเข้ารหัส และ การถอดรหัส ข้อมูล มีรายละเอียดดังนี้

#### 14.1.1 การเข้ารหัสข้อมูลและถอดรหัสข้อมูล (Encryption and Decryption)

การเข้ารหัสข้อมูลเป็นวิธีการที่ทำให้ข้อมูลที่ถูกเข้ารหัสไม่สามารถอ่านเข้าใจได้ และการถอดรหัสข้อมูลเป็นการทำให้ข้อมูลที่ถูกเข้ารหัสกลับมาเป็นข้อมูลที่สามารถอ่านเข้าใจได้ กระบวนการเข้ารหัส (Cryptographic Algorithm) มักเรียกอีกอย่างว่า Cipher เป็นกระบวนการทางคณิตศาสตร์ที่ใช้ทำการเข้ารหัสและถอดรหัส

ในที่นี้จะกล่าวถึงการเข้ารหัสและถอดรหัส โดยใช้ Symmetric-Key Encryption และ Public-Key Encryption

14.1.2 การเข้ารหัสแบบสมมาตร (Symmetric-Key Encryption) คือ ใช้คีย์ตัวเดียวกันทั้งการเข้ารหัสและถอดรหัส

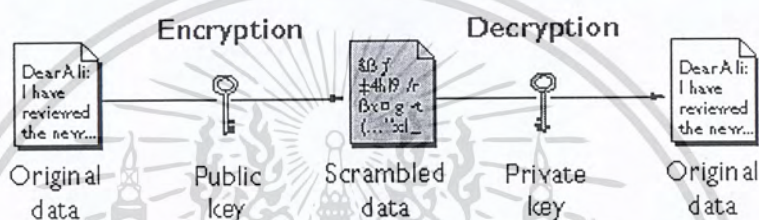


รูปที่ 14-1 แสดงการเข้ารหัสแบบสมมาตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้ารหัสแบบสมมาตรมีประสิทธิภาพสูง มีความเร็วในการเข้ารหัสและถอดรหัสสูง แต่ คีย์ที่ใช้ในการเข้ารหัสและถอดรหัสต้องถูกเก็บไว้เป็นความลับ ถ้าถูกมือที่สามเอาไปได้จะทำให้ข้อมูลถูกถอดรหัสได้ การเข้ารหัสแบบสมมาตรมีบทบาทสำคัญสำหรับ SSL Protocol ที่จะกล่าวถึงในบทนี้

**14.1.3 การเข้ารหัสโดยพับลิกคีย์ (Public-Key Encryption) หรือ เรียกอีกอย่างว่า การเข้ารหัสแบบอสมมาตร (Asymmetric Encryption) เป็นวิธีที่ที่ใช้นิยมมาก โดยการเข้ารหัสและถอดรหัสจะใช้คีย์ สองตัว คือ พับลิกคีย์ (Public key) และ ไพรวเทคคีย์ (Private key) โดย พับลิกคีย์เป็นคีย์ที่สามารถให้ผู้อื่นรู้ได้ ส่วน ไพรวเทคคีย์เป็นคีย์ที่ต้องเก็บไว้เป็นความลับห้ามให้ผู้อื่นรู้ โดยการเข้ารหัสถ้าเข้าโดยพับลิกคีย์จะต้องถอดรหัสด้วยไพรวเทคคีย์ และในทางตรงกันข้าม ถ้าเข้ารหัสด้วยไพรวเทคคีย์จะต้องถอดรหัสด้วยพับลิกคีย์**



รูปที่ 14-2 การเข้ารหัสแบบอสมมาตร

โดยการส่งข้อมูลถ้าเราจะส่งข้อมูลให้กับใครนั้น จะต้องทำการเข้ารหัสข้อมูลด้วยพับลิกคีย์ของคนๆ นั้น แล้วทำการส่งข้อมูลไปให้ แล้วจากนั้นคนๆ นั้นจะทำการถอดรหัสด้วยไพรวเทคคีย์ของตนเอง บางทีมีการใช้การเข้ารหัสแบบพับลิกคีย์ในการส่ง คีย์แบบสมมาตร ซึ่งเป็นวิธีการที่ใช้ใน SSL Protocol

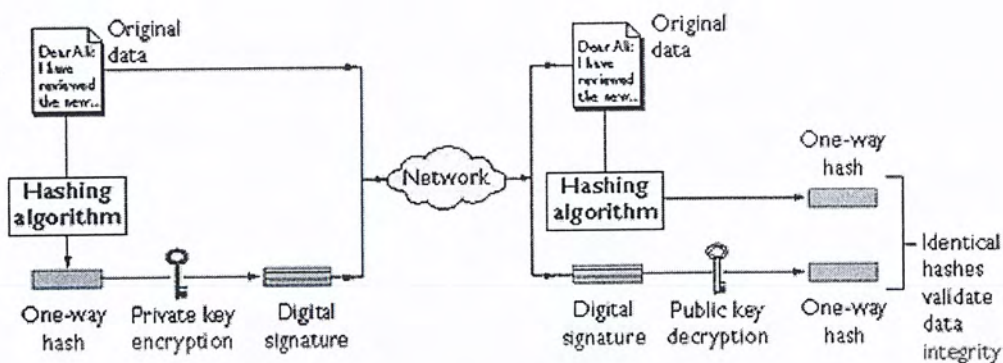
การเข้ารหัสโดยใช้ ไพรวเทคคีย์นั้นไม่ควรใช้กับข้อมูลที่ต้องการความปลอดภัยสูง แต่ก็มีประโยชน์มากในการใช้เป็นลายเซ็นดิจิทัล (Digital signature)

**14.2 การแก้ปัญหาการลักลอบเปลี่ยนข้อมูล ทำได้โดยการใช้ ลายเซ็นอิเล็กทรอนิกส์ มีรายละเอียดดังนี้**

#### 14.2.1 ลายเซ็นดิจิทัล (Digital signature)

ใช้วิธีการเข้ารหัสแบบพับลิกคีย์เข้ามาช่วย โดยการตรวจสอบการเปลี่ยนแปลงข้อมูลทำได้โดยใช้กระบวนการทางคณิตศาสตร์ เรียกว่า One-way hash หรือ Message digest ซึ่งมีคุณสมบัติสองอย่างคือ 1. ข้อมูลที่ต่างกันเพียงอักขระตัวเดียวก็จะได้ ผลของการ Hash ไม่เหมือนกัน 2. การ Hash เมื่อทำกับข้อมูลแล้วไม่สามารถทำให้กลับมาเหมือนเดิมได้

ดังที่ได้กล่าวมาแล้วว่า เราสามารถใช้ ไพรวเทคคีย์ในการเข้ารหัส และถอดรหัสด้วย พับลิกคีย์ได้ แม้วิธีนี้จะไม่ดีต่อการส่งข้อมูลที่เป็นความลับ แต่กลับเป็นประโยชน์ในการทำลายเซ็นดิจิทัล



รูปที่ 14-3 แสดงการลงลายเซ็นดิจิทัล

ขั้นตอนการลงลายเซ็นนั้นจะมีการส่งข้อมูลไป สองชุด คือ ชุดที่ 1 เป็นข้อมูลต้นฉบับ ที่ไม่ผ่านกระบวนการใดๆ ทั้งสิ้น ชุดที่ 2 เป็นข้อมูลต้นฉบับที่ผ่านการทำ Hash แล้วทำการ ลงลายเซ็นด้วยไพรเวทคีย์ เสร็จแล้วทำการส่งข้อมูลทั้งสองไปพร้อมกัน เมื่อส่งมาถึงทางด้านผู้รับแล้วผู้รับจะทำการ ตรวจสอบโดยการ ทำสองขั้นตอนดังนี้ ขั้นที่ 1 นำข้อมูลต้นฉบับที่ได้มาทำการ Hash ขั้นที่ 2 นำลายเซ็นอิเล็กทรอนิกส์มาถอดรหัสด้วย พับลิคคีย์ เมื่อทำทั้งสองขั้นตอนแล้วนำข้อมูลที่ ได้มาเปรียบเทียบกันถ้าเหมือนกันหมายความว่า ข้อมูลไม่ถูกแก้ไขระหว่างทาง แต่ถ้าเปรียบเทียบแล้วไม่เหมือนกันหมายความว่าข้อมูลถูกแก้ไขระหว่างทาง

### 14.3 การแก้ปัญหาการปลอมตัวเข้ามา ทำโดยการใช้ Certificate และ Authentication มีรายละเอียดดังนี้

14.3.1 Certificate เป็นเหมือนใบรับรองที่ใช้เฉพาะบุคคล เซิร์ฟเวอร์ หรือ บริษัทต่างๆ เป็นเหมือนใบอนุญาต พาสปอร์ต หรือบัตรประจำตัวประชาชน บอกถึงลักษณะของตัวบุคคล ช่วยแก้ปัญหาการปลอมตัวเข้ามาได้

การทำงานของ Certificate ใช้ในการบ่งชี้ถึงบุคคล โดย Certificate Authorities (CAs) จะออกให้ โดยบุคคลที่สามที่เชื่อถือได้ หรือ องค์กรที่มี Certificate เซิร์ฟเวอร์ ตัว Certificate ที่ออกให้มานั้นจะใช้ดูได้โดย พับลิคคีย์ที่เข้ากับไพรเวทคีย์ที่เข้ารหัส Certificate เท่านั้น เพื่อเป็นการยืนยันว่าผู้ที่ถือ Certificate เป็นตัวจริง

ตัวอย่างของ Certificate

- - - - - B E G I N      C E R T I F I C A T E - - - - -  
 MIICKzCCA ZSgAwIBAgIBAzANBgkqhkiG9w0BAQQFADA3MQswCQYDVQQGEwJVUzER  
 MA8GA1UEChMITmV0c2NhcGUxFTATBgNVBAsTDFNlcHJpeWEncyBDQTAeFw05NzEw  
 MTgwMTM2MjVaFw05OTEwMTgwMTM2MjVaMEGxCzAJBgNVBAYTAiVTMREwDwYDVQQK  
 EwhOZRZrZy2FwZTENMA sGA1UECxMEUHVicZEXMBUGA1UEAxMOU3Vwcm15YSBtGV0  
 dHkwZ8wDQYJKoZIhvcNAQEFBQADgY0AMIGJAoGBAMr6eZiPGfjX3uRjGjEjmKiqG  
 7SdATYazBcABu1AVyd7chRkiQ31FbXFOGD3wNktbf6hRo6EAmM5/R1AskzZ8AW7L  
 iQZBcrXpc0k4du+2Q6xJu2MPm/8WKuMOnTuvzpo+SGXeImHVChEqooCwfdiZywyZ  
 NMmrJgaoMa2MS6pUkfQVAgMBAAGjNjA0MBEGCWCGSAGG+EIBAQQEAwIAgDAfBgNV  
 HSMEGDAWgBTy8gZZkBhHUFWJM1oxeuZc+zYmyTANBgkqhkiG9w0BAQQFAAQBt  
 I6/z07Z635DfzX4XbAFpjIRI/AYwQzTSYx8GfcNAqCqCwaSDKvsuj/vwbf91o3j3  
 UkdGYpcd2cYRCgKi4MwqdWyLtpuHAH18hHZ5uvi00mJYw8W2wUOsY0RC/a/IDy84  
 hW3WWehBUqVK5SY4/zJ4oTjx7dwNMdGwbWfpRqjd1A==  
 -----END CERTIFICATE-----

**14.3.2 Authentication** เป็นการยอมรับบุคคลที่เข้ามาให้เข้ามาได้หรือไม่ มีทั้งการ Authentication ที่เซิร์ฟเวอร์ และการ Authentication ที่ไคลเอนต์ซึ่งการ Authentication ทั้งสองแบบ สามารถแบ่งย่อยได้อีกตามนี้คือ การ Authentication ที่ไคลเอนต์ มีทั้ง Password Based Authentication ซึ่งเป็นการ ใช้ชื่อและรหัสผ่าน ในการบังคับตัวไคลเอนต์ และ Certificate Based Authentication เป็นการ ใช้ Certificate ในการ Authentication ซึ่งจะอยู่ในส่วนของ SSL Protocol ในการลงลายเซ็นดิจิทัล

**14.4 SSL Protocol**

Secure Sockets Layer (SSL) protocol เป็นวิธีการในการทำ Authentication ที่เซิร์ฟเวอร์ และ การ ทำ Authentication ที่ไคลเอนต์ รวมไปถึงการเข้ารหัสระหว่างการส่งข้อมูลของเซิร์ฟเวอร์และไคลเอนต์ SSL ใช้กันอย่างกว้างขวางในอินเทอร์เน็ตโดยเฉพาะอย่างยิ่งการส่ง หมายเลข เครดิตการ์ด

SSL ต้องการ Certificate เป็นอย่างน้อยในการทำ Handshake ขณะเริ่มต้นการส่งข้อมูล โดย เซิร์ฟเวอร์จะส่ง Certificate ให้ไคลเอนต์ในการทำ Authentication ซึ่งในกระบวนการนั้นมีทั้งการใช้ การ เข้ารหัสแบบพับลิคคีย์ การใช้ลายเซ็นดิจิทัล ในการยอมรับเซิร์ฟเวอร์ หลังจากการ Authentiction เสร็จ แล้ว ไคลเอนต์และเซิร์ฟเวอร์จะใช้ การเข้ารหัสแบบสมมาตร ซึ่งมีความเร็ว ในการคุยกันทั้งสองฝ่าย

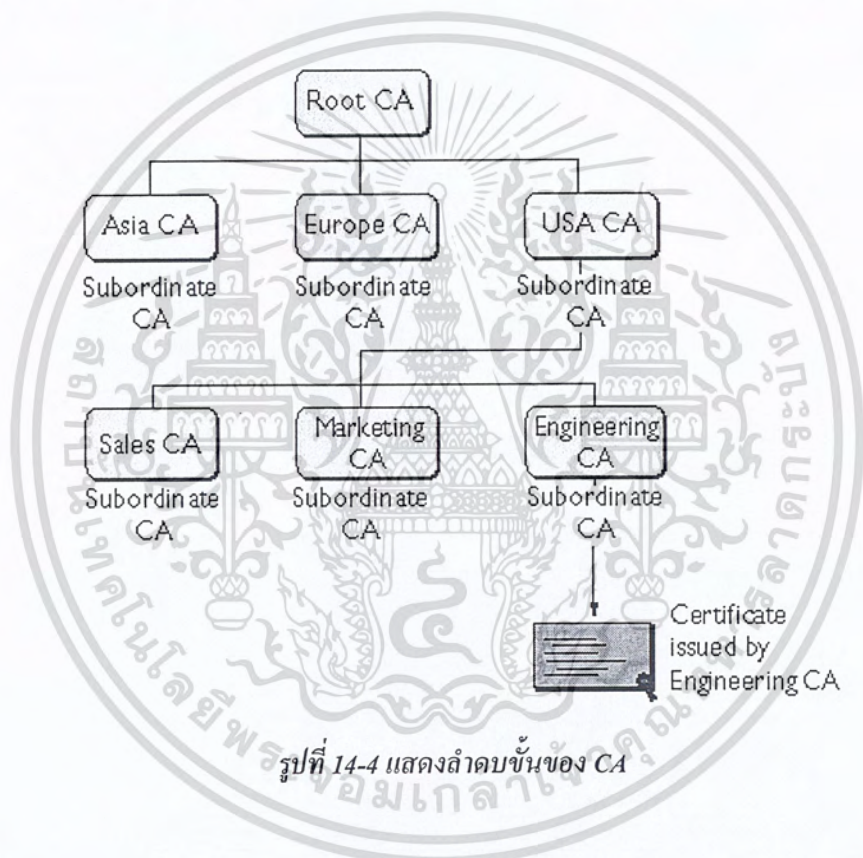
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางฝั่งเซิร์ฟเวอร์สามารถเลือกได้ว่าจะให้มีการ Authentication ที่ฝั่งไคลเอนต์ได้ด้วยหลังจาก การที่ทำการ Authentication เซิร์ฟเวอร์เสร็จแล้ว จึงจะทำการ Authentication ไคลเอนต์ ก่อนที่จะเกิด SSL Connection

Certificate hierarchy และ Certificate chain เป็นการบอกถึงลำดับการเชื่อถือ (Trust) ของ Certificate

#### 14.5 Certificate hierarchy

ในองค์กรขนาดใหญ่อาจแบ่งหน้าที่ในการออก Certificate เป็นหลายๆ ส่วน เช่น มีคนจำนวนมากที่ต้องการ Certificate เกินที่ CA เดียวจะรับได้ เป็นไปได้ว่า ตัว CA จะมอบหน้าที่ให้ลูกของมันทำแทนได้

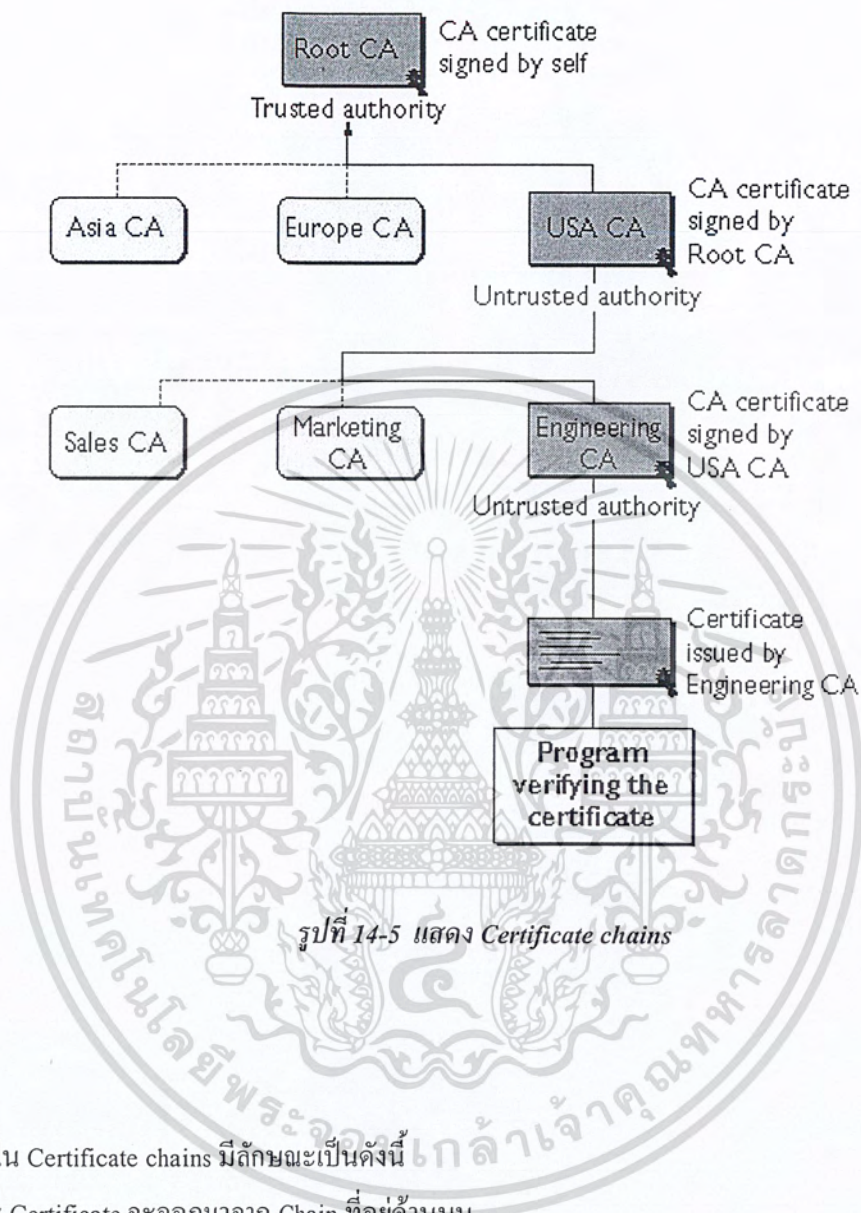


CA ที่อยู่ด้านล่างจะถูก Sign ด้วย CA ที่อยู่ด้านบน ทำให้การกระจายการออก CA เป็นไปได้ทั่วถึง โดย ตัวที่อยู่บนสุด Root CA จะเป็น Self-signed certificate ทำ การ Sign ลูกที่อยู่ต่ำลงมา

#### 14.6 Certificate chains

ตัว CA hierarchy มีผลต่อมายัง Certificate chains ซึ่งเป็นชุดของการออก Certificate โดย CA ให้กับลูกที่อยู่ต่ำกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



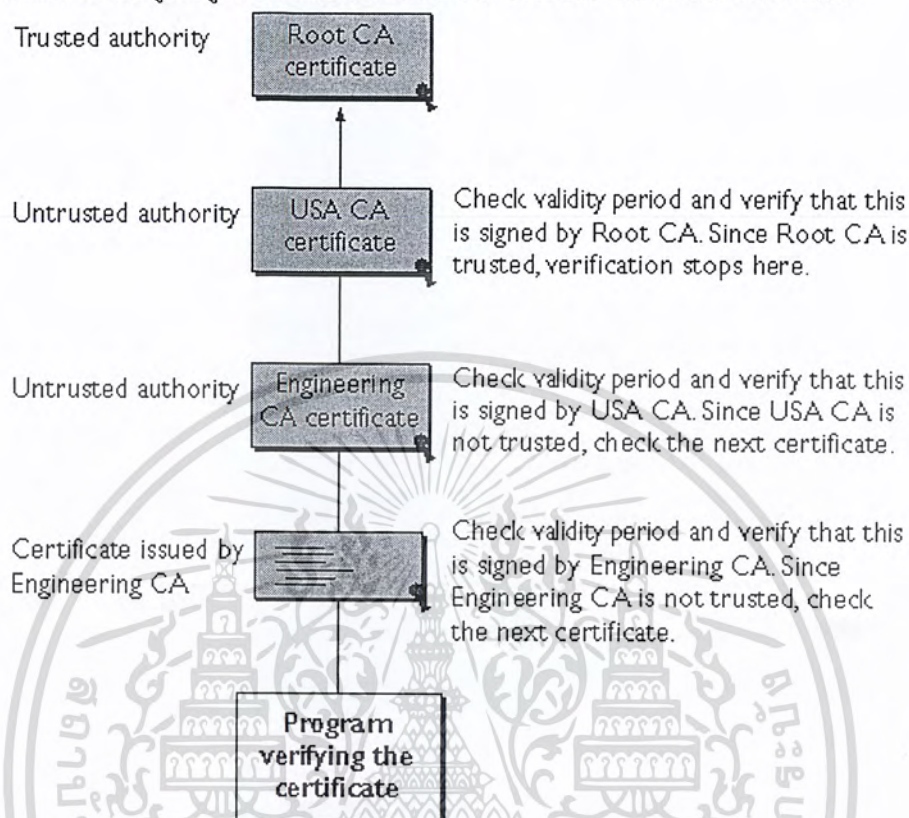
ใน Certificate chains มีลักษณะเป็นดังนี้

- แต่ละ Certificate จะออกมาจาก Chain ที่อยู่ด้านบน
- แต่ละ Certificate จะเก็บข้อมูล ของผู้ที่ออกให้ ตามรูป Engineering CA จะเก็บข้อมูล (DN) ของ USA CA เอาไว้ และ USA CA ก็จะเก็บข้อมูลของ CA ที่อยู่ถัดขึ้นไป
- แต่ละ Certificate จะถูก เซ็น ด้วย ไพรเวตคีย์ของผู้ที่ออกให้ และเป็นอย่างนี้ขึ้นไปเรื่อยๆ ตามรูป USA CA จะเซ็น Certificate ให้ Engineering CA

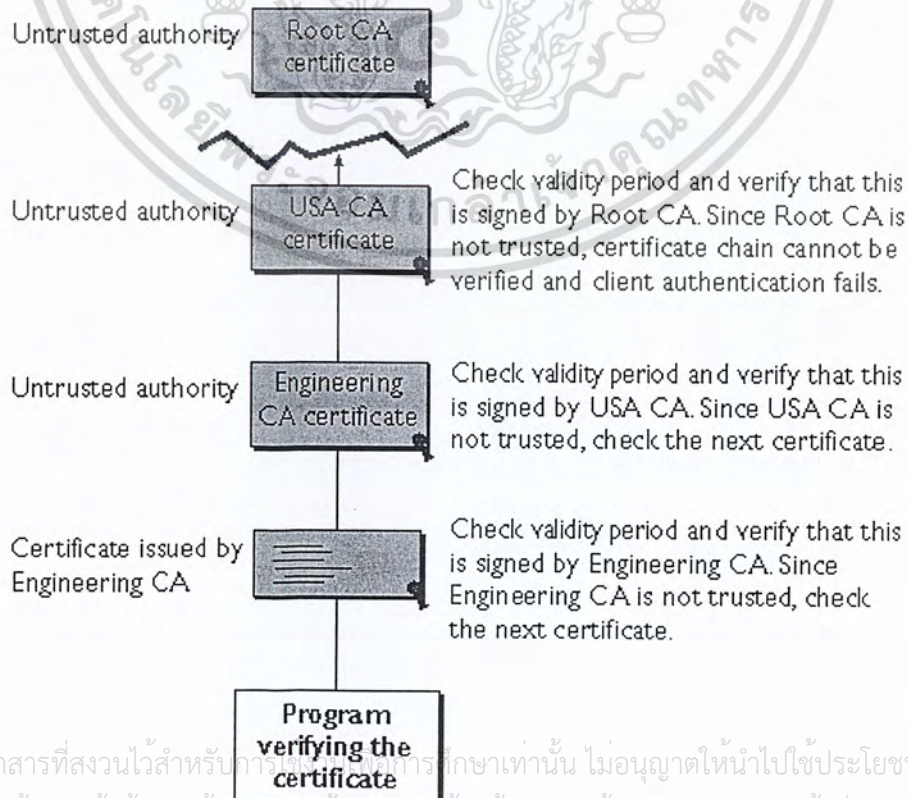
#### 14.7 การตรวจสอบ Certificate chains

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามี Certificate Chains ต่อขึ้นไปเรื่อยๆ หลายชั้น การตรวจสอบจะตรวจสอบกับ CA ที่ออก Certificate ให้ความเชื่อได้หรือไม่ ถ้าไม่ก็จะ ตรวจสอบขึ้นไปเรื่อยๆ จนกว่าจะมี CA ที่ เชื่อถือได้ ถ้าหากไม่มี ที่เชื่อถือได้เลย ก็จะขึ้นอยู่กับ ผู้ใช้ว่าจะเชื่อหรือไม่ แต่ถ้ามี CA ที่เชื่อได้ก็จะทำการเชื่อให้เลย



รูปที่ 14-6 แสดงการ Trust ของ Certificate chains

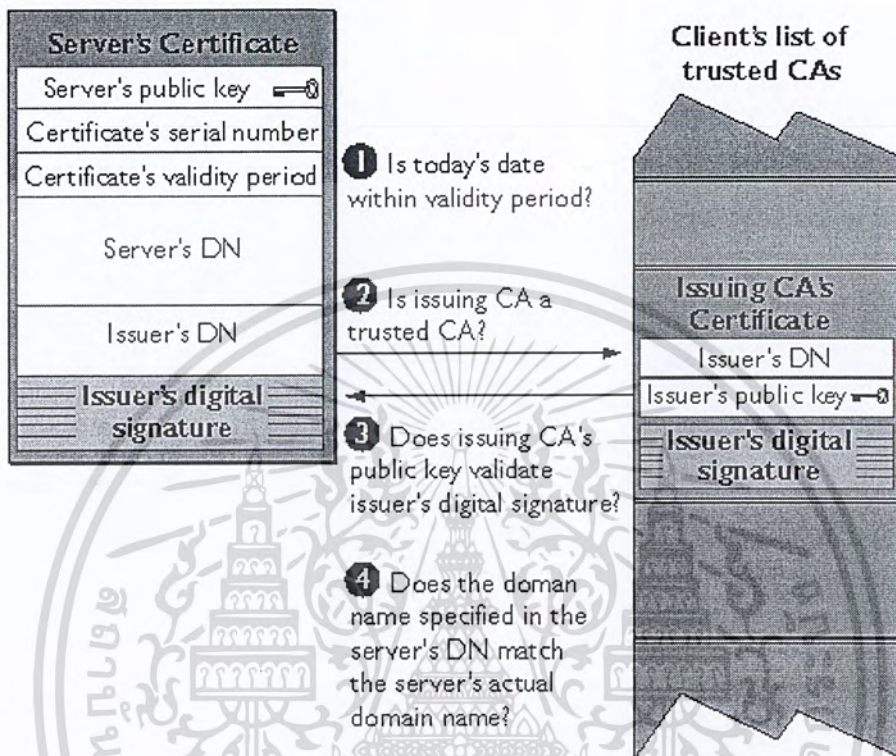


รูปที่ 14-7 แสดงการไม่ Trust ของ Certificate chains

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ทางการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 14.8 SSL Server authentication

เป็นการพิสูจน์ตนของ เซิร์ฟเวอร์ โดย SSL สามารถให้ไคลเอนต์ใช้การเข้ารหัสแบบพับลิคคีย์มาตรฐาน ในการตรวจสอบ Certificate ของ เซิร์ฟเวอร์ ว่าเชื่อถือได้ตามที่ CA ได้ออกให้มาหรือไม่ กระบวนการนี้สำคัญมาก เช่น การส่งหมายเลขเครดิตการ์ด



รูปที่ 14-8 แสดงการ Authentication ของเซิร์ฟเวอร์

ขั้นตอนการ ระหว่างการ Authentication เป็นดังนี้

1. ไคลเอนต์ตรวจสอบว่า Certificate ของเซิร์ฟเวอร์นั้นเทียบกับเวลาปัจจุบันว่าหมดอายุหรือยัง ถ้าหมดอายุแล้วการ Authentication จะหยุดทันที แต่ถ้ายังไม่หมดอายุจะไปทำข้อสอง
2. ไคลเอนต์ทำการตรวจสอบว่า CA ที่ออก Certificate นั้นอยู่ในบัญชีที่เชื่อถือได้หรือไม่ถ้าอยู่ในบัญชีที่เชื่อถือได้ก็จะผ่านไปทำขั้นที่สามต่อไป แต่ถ้าไม่อยู่ในบัญชีที่เชื่อถือได้ก็จะทำการตรวจขึ้นไปตาม Certificate chains ตามที่ได้กล่าวไปแล้วข้างต้น
3. ไคลเอนต์ใช้พับลิคคีย์ที่มากับ Certificate ทำการตรวจสอบลายเซ็นดิจิทัลว่าตรงกันหรือไม่ ถ้าข้อมูลของ Certificate ถูกเปลี่ยนแปลงจะทำให้ ข้อมูลที่ถูกถอดรหัสโดย พับลิคคีย์จะไม่ตรงกับข้อมูลใน Certificate จะหยุดทำการ Authentication แต่ถ้าตรงกันจะไปทำขั้นต่อไป
4. ตรวจสอบว่า โดเมนเนมใน Certificate ตรงกับ โดเมนเนมของเซิร์ฟเวอร์หรือไม่ ถ้าตรงกัน จึงไปทำขั้นต่อไป

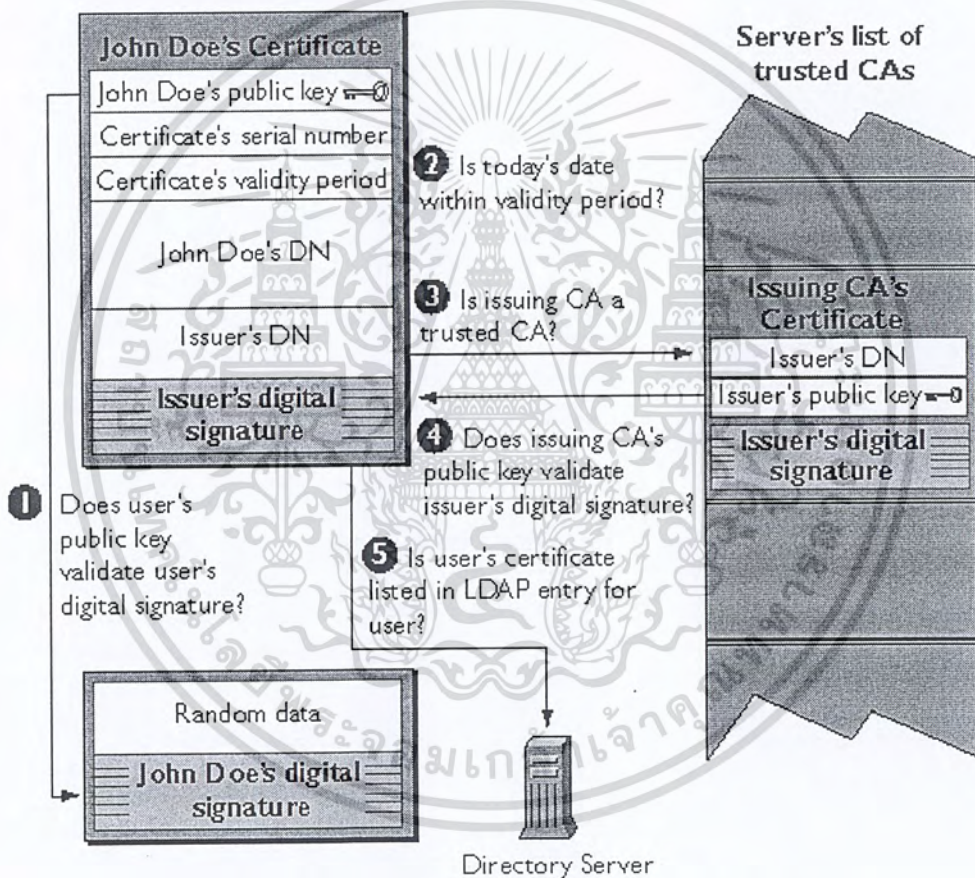
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เซิร์ฟเวอร์ผ่านการ Authentication จะใช้ Protocol SSL Handshake ซึ่งจะกล่าวต่อไปทำการสร้างการติดต่อระหว่างเซิร์ฟเวอร์ กับไคลเอนต์

#### 14.9 Client Authentication

SSI สามารถให้มี Authentication ของไคลเอนต์ได้ โดยไคลเอนต์จะส่ง Certificate และ ลายเซ็นดิจิทัล ไปให้เซิร์ฟเวอร์ เพื่อทำการ Authentication แล้วเซิร์ฟเวอร์จะใช้ลายเซ็นดิจิทัลในการ ตรวจสอบ Certificate

SSL Protocol ให้ไคลเอนต์ทำการสร้างลายเซ็นดิจิทัลโดยใช้ One way hash ในการ hash ข้อมูลที่ส่งขึ้นมาระหว่างการ handshake โดยจะรู้กันเพียงสองฝ่าย คือ ไคลเอนต์กับเซิร์ฟเวอร์เท่านั้น โดยข้อมูลที่ถูกลายเซ็น จะถูกเข้ารหัสด้วย โพรโทคอลของไคลเอนต์ ที่เข้ากันได้กับพับลิคคีย์ที่ส่งไปให้เซิร์ฟเวอร์



รูปที่ 14-9 แสดงการ Authentication ของไคลเอนต์

ขั้นตอนการ ระหว่างการ Authentication เป็นดังนี้

1. ตรวจสอบลายเซ็นดิจิทัล ของไคลเอนต์ ว่าถูกต้องหรือไม่
2. เซิร์ฟเวอร์ตรวจสอบ ว่า Certificate ของไคลเอนต์นั้นเทียบกับเวลาปัจจุบันว่าหมดอายุหรือ

ยัง ถ้าหมดอายุแล้วการ Authentication จะหยุดทันที แต่ถ้ายังไม่หมดอายุจะไปทำข้อสาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เซิร์ฟเวอร์ทำการตรวจสอบว่า CA ที่ออก Certificate นั้นอยู่ในบัญชีที่เชื่อถือได้หรือไม่ถ้าอยู่ในบัญชีที่เชื่อถือได้ก็จะผ่านไปทำขั้นที่สี่ต่อไป แต่ถ้าไม่อยู่ในบัญชีที่เชื่อถือได้ก็จะทำการตรวจสอบขึ้นไปตาม Certificate chains ตามที่ได้กล่าวไปแล้วข้างต้น
4. เซิร์ฟเวอร์ใช้ฟังก์ชันที่มากับ Certificate ทำการตรวจสอบลายเซ็นดิจิทัลของ Certificate ว่าตรงกันหรือไม่ ถ้าข้อมูลของ Certificate ถูกเปลี่ยนแปลงจะทำให้ ข้อมูลที่ถูกถอดรหัสโดย ฟังก์ชันจะไม่ตรงกับข้อมูลใน Certificate จะหยุดทำการ Authentication แต่ถ้าตรงกันจะไปทำขั้นต่อไป
5. ถ้ามีการใช้ไคลเอนต์เซอรัวส์ก็จะทำการตรวจสอบว่าผู้ใช้นั้นอยู่ในบัญชีหรือไม่ ถ้าไม่มีก็จะเข้าใช้เซิร์ฟเวอร์ไม่ได้
6. เซิร์ฟเวอร์ตรวจสอบว่าไคลเอนต์จะใช้ทรัพยากรใด โดยดูว่ามีสิทธิใช้ได้หรือไม่ ถ้าไม่มีสิทธิก็จะเข้าใช้ไม่ได้

#### 14.10 SSL Handshake

ใน SSL Protocol ใช้ทั้ง การเข้ารหัสทั้งสองแบบคือ การเข้ารหัสแบบฟังก์ชัน และการเข้ารหัสแบบสมมาตรโดยการเข้ารหัสแบบสมมาตรจะทำให้เร็วกว่าการเข้ารหัสแบบฟังก์ชัน แต่การเข้ารหัสแบบฟังก์ชันทำมาเพื่อการ Authentication ตัว SSL session จะเริ่มการติดต่อกันด้วยการทำ SSL Handshake ในการ handshake นั้นทำการ Authenticate ตัวเซิร์ฟเวอร์ด้วยวิธีฟังก์ชัน และก็ให้ไคลเอนต์ติดต่อกับเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบสมมาตร ขั้นตอนการทำ SSL Handshake มีดังนี้

1. ไคลเอนต์ส่ง เวอร์ชันของ SSL, Cipher setting, ข้อมูลที่สุ่มขึ้นมา และข้อมูลที่เซิร์ฟเวอร์ต้องการให้เซิร์ฟเวอร์
2. เซิร์ฟเวอร์ส่ง เวอร์ชันของ SSL, Cipher setting, ข้อมูลที่สุ่มขึ้นมา และข้อมูลที่ไคลเอนต์ต้องการให้ไคลเอนต์ โดยส่วนมากเซิร์ฟเวอร์จะส่ง Certificate มาให้ และถ้าไคลเอนต์ต้องการทรัพยากรจะต้องมีการ Authenticate ฝั่งไคลเอนต์ด้วย
3. ไคลเอนต์ใช้ข้อมูลที่ส่งมาจากเซิร์ฟเวอร์ในการ Authenticate เซิร์ฟเวอร์
4. ไคลเอนต์ทำการใช้ข้อมูลที่สร้างจากการ Handshake สร้าง Premaster secret แล้วเข้ารหัสด้วยฟังก์ชันของเซิร์ฟเวอร์ส่งไปยังเซิร์ฟเวอร์
5. ถ้าเซิร์ฟเวอร์ต้องการการ Authentication ของไคลเอนต์ ตัวไคลเอนต์เองต้องนำข้อมูลที่ถูกเซ็นและ Certificate ส่งมาให้เซิร์ฟเวอร์เพื่อทำการ Authenticate
6. เซิร์ฟเวอร์ทำการถอดรหัสด้วย ไพรวาทีย์แล้วนำ Premaster secret มาสร้าง Master secret
7. ทั้ง เซิร์ฟเวอร์และไคลเอนต์ใช้ Master secret ในการสร้าง Session keys ซึ่งเป็นคีย์แบบสมมาตรจะใช้ในการคุยกันใน SSL session
8. ไคลเอนต์ส่งข้อความไปบอกเซิร์ฟเวอร์ว่า เสร็จการ Handshake แล้ว
9. เซิร์ฟเวอร์ส่งข้อความไปบอกไคลเอนต์ว่า เสร็จการ Handshake แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. SSL Handshake เสร็จแล้วจะเกิด SSL session ขึ้นแล้วการติดต่อกันระหว่างเซิร์ฟเวอร์และไคลเอนต์ จะใช้ Session key ในการเข้ารหัสและถอดรหัส

#### 14.11 โมเดลความปลอดภัยของ COM+

จุดประสงค์ของโมเดลมี 4 ประการคือ

1. Activation Control
2. Access Control
3. Authentication Control
4. Identify Control

Activation Control เป็นการควบคุมว่าใครสามารถ launch คอมโพเนนต์ได้ Access Control เป็นการจำกัดการเข้าถึงออบเจกต์ของคอมโพเนนต์ ในบางกรณีผู้ใช้อาจได้รับอนุญาตและไม่อนุญาตให้เข้าถึงฟังก์ชันบางส่วนของคอมโพเนนต์ ตัวอย่างเช่น ผู้ใช้ทำการติดต่อผ่านทาง WWW ได้รับอนุญาตให้เรียกใช้บางเมธอดของคอมโพเนนต์ และมีบางส่วนของคอมโพเนนต์ที่สงวนให้ใช้ได้เฉพาะผู้ดูแลระบบ ดังนั้นจึงมี Authentication Control เพื่อจำกัดการเข้าถึงทรัพยากรจากผู้ใช้ประเภทต่าง ๆ นอกจากนั้นระดับของ Authentication Control ยังเป็นการกำหนดการเข้ารหัสข้อมูลก่อนส่งผ่านเครือข่าย Identify Control เป็นการกำหนดหลักฐานด้านความปลอดภัยของคอมโพเนนต์ที่ทำงาน ซึ่งมีรายละเอียดของยูสเซอร์แอคเคานต์หรือรายละเอียดของไคลเอนต์

การกำหนดความปลอดภัยของ COM+ คอมโพเนนต์ สามารถทำได้ 2 วิธีคือ

1. Declarative Security
2. Programmatic Security

Declarative Security เป็นการกำหนดความปลอดภัยลงในรีจิสทรีซึ่งกระทำโดยผู้ดูแลระบบ Programmatic Security เป็นการเขียนโปรแกรมเพื่อควบคุมความปลอดภัย ทำโดยผู้พัฒนาคอมโพเนนต์ โมเดลความปลอดภัยทั้ง 4 แบบสามารถใช้ Declarative Security กำหนดได้ทั้งหมด

##### 14.11.1 Declarative Security

ก่อนหน้าที่จะมี COM+ เราสามารถกำหนดความปลอดภัยแบบ Declarative Security โดยใช้แอปพลิเคชัน dcomcnfg ปัญหาหลักของ dcomcnfg คือค่าความปลอดภัยจะครอบคลุมทั้งหมดของ COM เซิร์ฟเวอร์, COM คลาสทั้งหมดที่อยู่ใน COM เซิร์ฟเวอร์เดียวกันมีค่าความปลอดภัยเหมือนกันหมด ซึ่งไม่เหมาะสมหาก COM เซิร์ฟเวอร์ประกอบด้วยคลาสหรืออินเตอร์เฟสที่ต้องการค่าความปลอดภัยแตกต่างกัน ตัวอย่างเช่น เรามีคอมโพเนนต์ที่ประกอบด้วย 2 อินเตอร์เฟส อินเตอร์เฟสแรกประกอบด้วยเมธอดต่าง ๆ ที่จัดเตรียมไว้ให้ผู้ใช้ทั่วไปใช้งาน ส่วนอินเตอร์เฟสที่สองเป็นของผู้ดูแลระบบเพื่อใช้ในการจัดการควบคุมคอมโพเนนต์โดยไม่อนุญาตให้ผู้ทั่วไปใช้อินเตอร์เฟสนี้ ในกรณีตัวอย่างที่กล่าวมา dcomcnfg ไม่เอกลักษณะเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถทำได้ วิธีเดียวที่ทำได้คือใช้ API ของ COM Security อย่างไรก็ตามในการใช้ COM+ รันไทม์ ก็ยังสามารถใช้ประโยชน์จากวิธีนี้ได้เมื่อต้องการให้ Role ของผู้ใช้เป็นเงื่อนไขสำหรับการทำงานของเมธอดในคอมโพเนนต์ เช่นให้บริการที่แตกต่างกันจากเมธอด เมธอดเดียวกัน เมื่อผู้ใช้อยู่ใน Role ที่แตกต่างกัน เป็นต้น

สำหรับ COM+ เราสามารถควบคุมการใช้เซิร์ฟเวอร์ได้ตั้งแต่ระดับคลาส อินเตอร์เฟส และเมธอดซึ่งเรียกว่า Fine-Grained Security โดยที่ไม่ต้องเขียนโปรแกรมเพิ่มเติม การใช้ Fine-Grained Security เราต้องทำความเข้าใจเกี่ยวกับโรล (Role) ซึ่งเป็นสิ่งสำคัญในการกำหนดความปลอดภัยบน COM+

#### 14.11.2 Role

COM+ ใช้โมเดลความปลอดภัยแบบใหม่โดยแสดงอยู่ในรูปแบบของโรล โรลคือกลุ่มของผู้ใช้ COM+ แอปพลิเคชันที่มีค่าความปลอดภัยเหมือนกัน การกำหนดโรลมีลักษณะเหมือนชีวิตจริงที่มีการกำหนดบทบาทให้บุคคลว่ามีบทบาทอะไรในสังคม สามารถทำหรือไม่อนุญาตให้ทำอะไรได้บ้าง ตัวอย่างเช่น COM+ แอปพลิเคชันที่ออกแบบสำหรับธนาคารแบ่งโรลออกเป็น Teller role, Manager role และ Loan Office role ผู้ใช้ที่อยู่ในโรลเดียวกันมีค่าความปลอดภัยเหมือนกัน แต่ละโรลอนุญาตและไม่อนุญาตให้ทำอะไรเหมือนกัน เมื่อกำหนดโรลให้กับ COM+ แอปพลิเคชัน เราสามารถเพิ่มผู้ใช้หรือกลุ่มของผู้ใช้เข้าไปในโรลได้ 2 วิธี คือใช้คอมโพเนนต์เซอร์วิสเอ็กชพลอเรอร์ (Declarative Security) หรือเขียนโปรแกรมโดยเรียกข้อมูงจากคอนเท็กซ์เพื่อดูว่าผู้ใช้มีโรลอะไร หากเป็นโรลที่ไม่อนุญาตก็สามารถหยุดการใช้งานหรือแจ้งความผิดพลาดที่เกิดขึ้น (Programmatic Security)

#### 14.11.3 การสร้าง Role

เราสามารถเพิ่มโรลให้ COM+ Application โดยมีขั้นตอนดังนี้

1. คลิกขวาที่โฟลเดอร์ Roles ของแอปพลิเคชันที่ต้องการเพิ่มโรล
2. เลือก New > Role จะปรากฏไดอะล็อก Role
3. ตั้งชื่อโรลที่ต้องการแล้วคลิก OK

เมื่อสร้างโรลแล้วต่อมาคือเพิ่มผู้ใช้เข้าไปในโรลและกำหนดส่วนต่าง ๆ ของแอปพลิเคชัน (คอมโพเนนต์, อินเตอร์เฟส, เมธอด) ว่าต้องการให้โรลใดเรียกใช้ได้

#### 14.11.4 การเพิ่มผู้ใช้เข้าไปในโรล

มีขั้นตอนดังนี้

1. คลิกขวาที่โฟลเดอร์ Users ที่อยู่ในไคลเรทอรีของโรลที่ต้องการเพิ่มผู้ใช้
2. เลือก New > Users จะปรากฏไดอะล็อกให้เลือกผู้ใช้ (ไดอะล็อกที่ปรากฏขึ้นอยู่ว่ามีใช้ Active Directory หรือไม่)
3. เลือกผู้ใช้ที่ต้องการแล้วคลิก OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 14.11.5 การกำหนดความปลอดภัยให้แต่ละส่วนของแอปพลิเคชัน

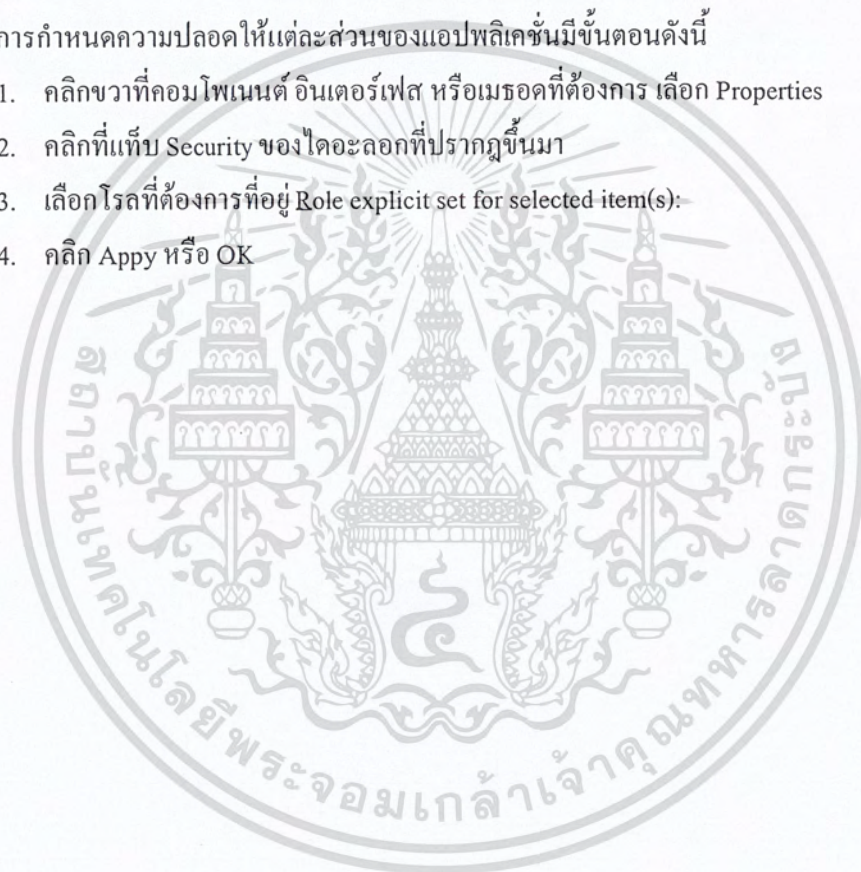
ก่อนอื่นเราต้องเลือกอบชั้น Access Control จึงจะใช้งานได้ โดยค่าดีฟอลต์แล้ว Access Control จะไม่ได้เลือกไว้ การเลือกอบชั้น Access Control มีขั้นตอนดังนี้

1. เปิดคอมพิวเตอร์เซิร์ฟเวอร์
2. คลิกขวาที่แอปพลิเคชันที่ต้องการ เลือก Properties
3. คลิกที่แท็บ Security ของไดอะล็อกคอมพิวเตอร์ที่ปรากฏขึ้นมา
4. เลือกเช็คบ็อกซ์ Enforce access checks for this application แล้วคลิก Apply หรือ OK

การเปลี่ยนแปลงค่าความปลอดภัยของ COM+ จะยังไม่เกิดผลทันทีจนกว่าเราจะรีสตาร์ทแอปพลิเคชันใหม่

การกำหนดความปลอดภัยให้แต่ละส่วนของแอปพลิเคชันมีขั้นตอนดังนี้

1. คลิกขวาที่คอมพิวเตอร์ อินเทอร์เน็ต หรือเมธอดที่ต้องการ เลือก Properties
2. คลิกที่แท็บ Security ของไดอะล็อกที่ปรากฏขึ้นมา
3. เลือกโรลที่ต้องการที่อยู่ Role explicit set for selected item(s):
4. คลิก Apply หรือ OK



## บทที่ 15

### เอกทิฟเซิร์ฟเวอร์เพจและอินเทอร์เน็ตอินฟอร์เมชันเซอร์วิส

#### 15.1 แนวคิดการทำงานของเอกทิฟเซิร์ฟเวอร์เพจ

เอกทิฟเซิร์ฟเวอร์เพจ (Active Server Page) หรือเอเอสพี (ASP) เป็นเทคโนโลยีการเขียนภาษาสคริปต์บนฝั่งเซิร์ฟเวอร์ (Server-side Script) ทำให้เราสามารถสร้างเว็บเพจแบบไดนามิก สามารถโต้ตอบกับผู้ใช้เว็บเพจได้ ไฟล์เอเอสพีเป็นแอสกีไฟล์ที่มีนามสกุล .asp มีขั้นตอนการทำงานคือ เมื่อไอไอเอสได้รับการร้องขอไฟล์เอเอสพี มันจะประมวลผลสคริปต์ในไฟล์ที่ฝั่งเซิร์ฟเวอร์ และสร้างเอกสารเอชทีเอ็มแอลส่งกลับไปยังบราวเซอร์ที่ร้องขอ เอเอสพีรองรับภาษาสคริปต์ทั้งวิชวลเบสิกสคริปต์และจาวาสคริปต์

#### 15.2 ออบเจกต์พื้นฐานของเอเอสพี

เอเอสพีมีออบเจกต์พื้นฐานทั้งหมด 6 ชนิด ได้แก่ออบเจกต์รีเควส (Request Object) ออบเจกต์เรสพอนซ์ (Response Object) ออบเจกต์แอปพลิเคชัน (Application Object) ออบเจกต์เซสชัน (Session Object) ออบเจกต์เซิร์ฟเวอร์ (Server Object) และออบเจกต์เอเอสพีเออเรอร์ (ASPError Object) ซึ่งทั้ง 6 ชนิด เราสามารถเขียนสคริปต์เพื่อควบคุม และเรียกใช้งานได้ทันที

ออบเจกต์รีเควสและออบเจกต์เรสพอนซ์จะทำหน้าที่ติดต่อกับผู้ใช้งาน โดยออบเจกต์รีเควสทำหน้าที่รับข้อมูลจากผู้ใช้งานเข้ามา ส่วนออบเจกต์เรสพอนซ์ทำหน้าที่ส่งข้อมูลจากเว็บเซิร์ฟเวอร์กลับไปให้ผู้ใช้งานโดยตรง ออบเจกต์ที่อยู่ถัดมาเป็นออบเจกต์เซสชันซึ่งทำหน้าที่เก็บข้อมูลของผู้ใช้งานแต่ละรายที่เข้ามาใช้แอปพลิเคชันเอเอสพี โดยสร้างเซสชันเป็นจำนวนเท่ากับจำนวนผู้ใช้งานที่เรียกใช้สำหรับออบเจกต์แอปพลิเคชันทำหน้าที่ติดต่อกับควบคุมการทำงานของแอปพลิเคชันเซิร์ฟเวอร์ และ ออบเจกต์เซิร์ฟเวอร์ทำหน้าที่ติดต่อกับระหว่างแอปพลิเคชันเอเอสพีกับเว็บเซิร์ฟเวอร์

ออบเจกต์เอเอสพีเออเรอร์ทำหน้าที่เก็บรายละเอียดการทำงานที่ผิดพลาด โดยเราจะตรวจสอบรายละเอียดความผิดพลาดของแอปพลิเคชันเอเอสพีที่เกิดขึ้น ได้จากรายละเอียดต่างๆ ที่เก็บอยู่ภายใน แล้วแก้ไขให้การทำงานต่างๆเป็นไปอย่างถูกต้อง

รายละเอียดของออบเจกต์พื้นฐานต่างๆ มีดังนี้

ออบเจกต์	คำอธิบาย
Request	ออบเจกต์ที่ใช้จัดการข้อมูลที่ได้รับเข้ามาจากผู้ใช้งาน
Response	ออบเจกต์ที่ใช้จัดการข้อมูลที่ส่งกลับไปยังผู้ใช้งาน
Session	ออบเจกต์ที่เก็บรายละเอียดของผู้ใช้งานแอปพลิเคชันเอเอสพีแต่ละคน
Application	ออบเจกต์ที่ใช้แลกเปลี่ยนข้อมูลกับแอปพลิเคชันเอเอสพี
Server	ออบเจกต์ที่ทำหน้าที่จัดการและบริหารทรัพยากรของเว็บเซิร์ฟเวอร์
ASPError	ออบเจกต์ที่ทำหน้าที่จัดการเมื่อเกิดความผิดพลาดจากการทำงานของแอปพลิเคชันเอเอสพี

#### ตารางที่ 15-1 ออบเจกต์พื้นฐานของเอเอสพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 15.2.1 ออบเจกต์รีเควส

ออบเจกต์รีเควสทำหน้าที่รับเอาข้อมูลจากบราวเซอร์ที่เรียกเข้ามา แล้วส่งข้อมูลไปให้กับออบเจกต์อื่นที่เกี่ยวข้อง ซึ่งข้อมูลที่ได้รับเข้ามาจากผู้ใช้งานอาจเป็นการร้องขอธรรมดาหรืออาจเป็นชุดข้อมูลที่ส่งผ่านฟอร์มก็ได้

ในบางครั้งข้อมูลที่ได้รับเข้ามาอาจมีการเข้ารหัสด้วยการใช้งานร่วมกับ โพรโตคอลเอสเอสแอล (Secure Sockets Layers) เพื่อความปลอดภัยได้ด้วย

#### คุณสมบัติของออบเจกต์รีเควส

*TotalBytes* แสดงจำนวนไบต์ของข้อมูลที่อ่านเข้ามา

#### เมธอดของออบเจกต์รีเควส

*BinaryRead* เป็นการอ่านข้อมูลจากผู้ใช้งานส่งเข้ามาด้วยวิธีโพสต์ (Post) ซึ่งการส่งด้วยวิธีการโพสต์จะเหมาะในการส่งข้อมูลขนาดใหญ่ และข้อมูลที่ไม่ใช่ตัวอักษร เช่น ภาพกราฟิก ไฟล์เสียง วิดีโอ เป็นต้น

#### คอลเล็กชันของออบเจกต์รีเควส

*Form* เป็นคอลเล็กชันที่รับเอาข้อมูลจากผู้ใช้งานกรอกเข้ามาทางฟอร์ม ข้อมูลที่ป้อนเข้ามาภายใต้แท็ก <FORM >

*QueryString* เป็นคอลเล็กชันที่รับข้อมูลที่ส่งมาโดยแปะท้ายมากับยูอาร์แอล (URL)

*Cookies* เป็นคอลเล็กชันที่รับข้อมูลในส่วนที่เป็นคุกกี้ (Cookie) ซึ่งเก็บไว้ในเครื่องคอมพิวเตอร์ของผู้ใช้งานนำมาประมวลผล

*ClientCertificate* เป็นคอลเล็กชันที่เก็บรายละเอียดเกี่ยวกับสิทธิในการใช้งาน โดยใช้กับแอปพลิเคชันที่ต้องการความมั่นใจเกี่ยวกับความปลอดภัยของข้อมูล

*ServerVariables* เป็นคอลเล็กชันที่เก็บค่าตัวแปรของเซิร์ฟเวอร์ โดยเก็บรายละเอียดด้านเทคนิคและสภาพการทำงานของเว็บเซิร์ฟเวอร์เอาไว้ซึ่งสามารถนำมาประกอบการใช้งานในแอปพลิเคชันเอสพีได้ ตัวอย่างของ *ServerVariables* ที่สำคัญ ๆ เช่น

ตัวแปร	คำอธิบาย
REQUEST_PROTOCOL	ชื่อของโพรโตคอล และเวอร์ชันที่รองรับรีเควสที่เข้ามาเช่น HTTP/1.0
REQUEST_METHOD	วิธีการส่งข้อมูลไปให้กับเซิร์ฟเวอร์ (เช่นวิธีเกทหรือวิธีโพสต์)
EXECUTABLE_PATH	เป็นพาทที่เราใช้เก็บโปรแกรมที่จะประมวลผล (เป็นพาทเสมือน (Virtual Path) เซิร์ฟเวอร์ต้องนำมาแปลงเป็นพาทจริง (Physical Path))
QUERY_STRING	เป็นข้อมูลขนาดเล็กที่ส่งไปพ่วงกับยูอาร์แอลเพื่อใช้เป็นอินพุตให้แอปพลิเคชันเอสพี
SERVER_SOFTWARE	ชื่อของซอฟต์แวร์ที่เซิร์ฟเวอร์รันอยู่ (ขึ้นอยู่กับโพรโตคอล) เช่น ชื่อของเว็บเซิร์ฟเวอร์
SERVER_NAME	ชื่อของเซิร์ฟเวอร์หรือไอพีแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำหรือดัดแปลงเอกสารทุกครั้งที่มีการนำไปใช้

SERVER_PORT	หมายเลขพอร์ตที่ใช้ (ปกติจะเป็นพอร์ต 80)
SERVER_ADMIN	อีเมลแอดเดรสของผู้ที่ดูแลเซิร์ฟเวอร์
REMOTE_HOST	เป็นชื่อคอมพิวเตอร์ของไคลเอนต์หรือโดเมนเนมของไคลเอนต์
CONTENT_TYPE	ประเภทข้อมูลที่ส่งไปยังเซิร์ฟเวอร์ด้วยเมธอดโพสต์ จะเป็นชนิดข้อมูลที่แบ่งไว้ใน MINE (Multipurpose Internet Mail Extensions)
CONTENT_LENGTH	ขนาดของข้อมูลหน่วยไบต์ ที่ถูกส่งไปยังเซิร์ฟเวอร์ด้วยเมธอด โพสต์

ตารางที่ 15-2 (ต่อ) ตัวอย่างของ *ServerVariables*

### 15.2.2 ออบเจกต์เรสพอนซ์

ออบเจกต์เรสพอนซ์ทำหน้าที่ส่งผลการทำงานที่ได้กลับไปยังผู้ใช้งานที่ได้รับร้องขอมา ซึ่งจะส่งออกไปในรูปของเอกสารเอกซ์เอ็มแอล ข้อความธรรมดา หรือคูกี้

#### คุณสมบัติของออบเจกต์เรสพอนซ์

##### *Buffer*

เป็นการเลือกว่าจะกักข้อมูลไว้ ก่อนจะส่งไปให้กับไคลเอนต์หรือไม่ ถ้ากักไว้ก็จะรอให้ข้อมูลครบถ้วนทั้งหมดแล้วส่งไปที่เดียว ซึ่งการใช้บัฟเฟอร์จะมีผลตั้งแต่บรรทัดแรกที่เรากำหนดว่าให้มีการใช้บัฟเฟอร์ (ถ้าต้องการใช้เรามากกำหนดในบรรทัดแรก)

##### *CharSet*

เป็นการกำหนดรูปแบบการแสดงผลประเภทข้อความที่แสดงที่บราวเซอร์ ซึ่งจะมีการตีความแตกต่างกันเมื่อใช้งานกันคนละภาษา

##### *ContentType*

เป็นชนิดของข้อมูลที่ส่งกลับไป ซึ่งเราจะแยกความแตกต่างด้วยชนิดข้อมูลที่แบ่งไว้ใน MINE (Multipurpose Internet Mail Extensions)

##### *Expires*

เป็นการกำหนดเวลาในหน่วยวินาที ที่เว็บเพจหมดอายุจากแคชของบราวเซอร์ เพราะว่าทุกครั้งที่เราเรียกแอปพลิเคชันเอ็นเอสพี มันต้องถูกประมวลผลทุกครั้ง ซึ่งบางครั้งการเรียกใช้ในระยะเวลากลี่ยังคงกันอาจไม่จำเป็นต้องประมวลผลให้เสียเวลาก็ได้ เพราะฉะนั้นจึงมีการกำหนดเวลาที่ข้อมูลพอจะมีความถูกต้องเอาไว้ อีกเหตุผลหนึ่งก็เพื่อลดภาระของเว็บเซิร์ฟเวอร์

##### *ExpiresAbsolute*

เป็นการระบุวันเวลาที่ชัดเจนที่เว็บเพจหมดอายุในแคชของบราวเซอร์

##### *IsClientConnected*

เป็นการเพิ่มความสามารถของออบเจกต์เรสพอนซ์ โดยจะถามว่าบราวเซอร์ยังเชื่อมต่อกับเซิร์ฟเวอร์หรือไม่ เราจะใช้ตรวจสอบว่าไคลเอนต์ที่ร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์นั้นยังเชื่อมต่ออยู่หรือไม่ ถ้าขาดการเชื่อมต่อไปก็สามารถยกเลิกการส่งข้อมูลได้

##### *Status*

เป็นสถานะที่เป็นผลลัพธ์ของการตอบสนองของเว็บเซิร์ฟเวอร์ ต่อการร้องขอของไคลเอนต์ ซึ่งมักถูกนำไปใช้บ่อย ๆ ในการบังคับให้ผู้ใช้งานป้อนรหัสผ่านก่อนเข้าใช้เว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### เมธอดของออบเจกต์เรสพอนซ์

<i>Write</i>	เป็นการเขียนข้อความไปยังไคลเอนต์
<i>BinaryWrite</i>	เป็นการเขียนข้อมูลที่ไม่ใช่ข้อความ เช่น รูปภาพ มีเดียไฟล์ ไปยังไคลเอนต์
<i>Clear</i>	เป็นการล้างข้อมูลที่อยู่ในบัฟเฟอร์ออกไปให้หมด
<i>End</i>	เป็นการสั่งให้นำข้อมูลในบัฟเฟอร์ส่งไปให้กับไคลเอนต์ (ซึ่งเติมบัฟเฟอร์แล้ว)
<i>Flush</i>	เป็นการสั่งให้นำข้อมูลในที่มีอยู่ในขณะนั้นส่งไปให้ไคลเอนต์ทันที
<i>Redirect</i>	เป็นการสั่งให้เบราว์เซอร์ย้ายไปอ่านหรือส่งต่อการทำงานไปยังเว็บเพจอื่นที่ได้ระบุไว้
<i>AddHeader</i>	เป็นการเขียนสตริงไปที่เฮดที่ทีพีเอสเคอร์ (HTTP Header) ซึ่งเป็นส่วนที่เราไม่ใช้แสดงผล แต่ใช้เพื่อส่งข้อมูลพิเศษอื่นๆ เพื่อใช้ควบคุมการทำงานของแอปพลิเคชัน
<i>AppendToLog</i>	เป็นการเขียนข้อมูลลงไปทีล็อกของเซิร์ฟเวอร์

ออบเจกต์เรสพอนซ์และออบเจกต์รีควีสควรมีคอลเล็กชันที่สำคัญคือคุกกี้ คุกกี้คือเท็กซ์ไฟล์ขนาดเล็กที่เว็บเซิร์ฟเวอร์ส่งเข้าไปเก็บในคอมพิวเตอร์ของผู้ใช้งานทั่วไป โดยจะเป็นข้อมูลให้กับเว็บเซิร์ฟเวอร์เรียกมาใช้งานเมื่อผู้ใช้งานเข้ามาใช้งานอีกครั้ง

เราอาจเห็นการใช้งานคุกกี้ในเว็บไซค์ที่มีการให้บริการสมาชิก ซึ่งการตรวจสอบการเป็นสมาชิกอาจทำได้ด้วยการใช้คุกกี้หรือการที่เก็บข้อมูลของผู้เข้าเยี่ยมชมเอาไว้ในคุกกี้ เมื่อเขากลับมาเยี่ยมชมอีกครั้ง เราสามารถนำเสนอเนื้อหาที่สัมพันธ์กับการเยี่ยมชมคราวก่อน โดยอ่านจากคุกกี้ที่มีอยู่ก็ได้

สำหรับการเก็บข้อมูลในคอลเล็กชันคุกกี้มีรูปแบบดังนี้

Response.Cookies(ชื่อคุกกี้)(คีย์ที่ใช้).แอตทริบิวต์ที่ใช้ = ค่าที่เก็บ

ชื่อคุกกี้เราจะกำหนดเป็นอะไรก็ได้ ส่วนคีย์ที่ใช้เป็นชื่อข้อมูลที่เก็บ ส่วนค่าที่เก็บจะสัมพันธ์กับชื่อของคีย์ สำหรับแอตทริบิวต์ (Attribute) ซึ่งจะถูกใช้ในการควบคุมการทำงานของคุกกี้ได้แก่

<i>Expires</i>	เป็นการกำหนดวันที่คุกกี้หมดอายุ
<i>Domain</i>	เป็นการจำกัดว่าคุกกี้มาจากเว็บเซิร์ฟเวอร์ตัวใด (เพราะบางทีอาจตั้งชื่อคุกกี้เหมือนกันก็ได้)
<i>Path</i>	เป็นชื่อพารามิเตอร์ในเว็บเซิร์ฟเวอร์จะถูกส่งไปในช่องทางที่ปลอดภัยหรือไม่ ถ้าเป็นข้อมูลที่สำคัญ เช่น ข้อมูลเกี่ยวกับการเงินอาจจำเป็นต้องใช้
<i>HasKeys</i>	เป็นข้อมูลที่ใช้บอกถึงแอตทริบิวต์ตัวอื่นๆ ในที่เก็บไว้ในคุกกี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 15.2.3 ออบเจกต์เซสชัน

ออบเจกต์เซสชันทำหน้าที่ดูแลและเก็บข้อมูลของแต่ละไคลเอนท์ที่ส่งการร้องขอเข้ามา โดยกำหนดหมายเลขของผู้ใช้บริการ (SessionID) ซึ่งจะช่วยให้เว็บเซิร์ฟเวอร์สามารถแยกแยะแต่ละไคลเอนต์เพื่อให้บริการได้อย่างเหมาะสม

#### คุณสมบัติของออบเจกต์เซสชัน

*SessionID* เป็นหมายเลขที่ออบเจกต์เซสชันกำหนดให้กับไคลเอนต์ ที่เข้าใช้แอปพลิเคชันเอเอสพี

*TimeOut* เป็นเวลาที่เว็บเซิร์ฟเวอร์จะตัดการติดต่อกับไคลเอนต์ หากไม่สามารถติดต่อกับไคลเอนต์ได้ หรือไคลเอนต์ไม่ตอบสนองตามเวลาที่กำหนดไว้ (มีหน่วยเป็นนาที ค่าดีฟอลต์คือ 20 นาที)

*LCID* LCID หรือโลคอลไอเดนติไฟเออร์ (Locate Identifier) เป็นข้อมูลในลักษณะตัวย่อ ที่บอกให้บราวเซอร์ทราบว่าบราวเซอร์ควรแสดงผลให้เข้ากับท้องถิ่น เช่น เว็บเซิร์ฟเวอร์นั้นต้องการแสดงหน่วยเงินตราของอังกฤษ ก็จะกำหนด LCID เป็นเลขฐานสิบหก คือ 0809 หรือถ้าเป็นเมืองไทย ต้องกำหนดเป็น 041E เป็นต้น (ดูข้อมูลเพิ่มเติมได้จากเว็บไซต์ [www.w3c.org](http://www.w3c.org))

#### เมธอดของออบเจกต์เซสชัน

*Abandon* เป็นการสั่งให้จบการทำงานของไคลเอนต์

#### คอลเล็กชันของออบเจกต์เซสชัน

นอกจากจะอ่านและกำหนดค่าจากคุณสมบัติต่างๆของออบเจกต์เซสชัน แล้วเรายังสามารถเก็บข้อมูลต่างๆของแต่ละเซสชัน ให้เหมาะกับลักษณะของแอปพลิเคชันได้ เช่น เราสร้างเว็บไซต์ขายสินค้า เราอาจต้องการเก็บข้อมูลของแต่ละไคลเอนท์ที่เข้ามาเป็นชื่อลูกค้า หรือสินค้าที่ได้ออเดอร์เอาไว้ เป็นต้น

เราสามารถเก็บข้อมูลข้างต้นของแต่ละเซสชัน ด้วยตัวแปรที่สร้างขึ้นมาพิเศษ แล้วเก็บตัวแปรทั้งหมดไว้ในคอลเล็กชันคอนเท็นต์ส (Contents)

ตัวแปรที่อยู่ในแต่ละเซสชันจะมีอายุมากกว่าผู้ใช้จะสิ้นสุดการทำงานแอปพลิเคชัน ซึ่งก็ทำให้ SessionID นั้นหมดอายุไป

#### อีเวนต์ของออบเจกต์เซสชัน

*Session\_OnStart* เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ ไคลเอนต์เริ่มใช้งานแอปพลิเคชันเอเอสพี

*Session\_OnEnd* เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ ไคลเอนต์สิ้นสุดการทำงานแอปพลิเคชันเอเอสพี

### 15.2.4 ออบเจกต์แอปพลิเคชัน

ออบเจกต์แอปพลิเคชันทำหน้าที่จัดการติดต่อกับแอปพลิเคชันที่ทำงานอยู่บนเซิร์ฟเวอร์ ซึ่งทำให้เราสามารถใส่ข้อมูลร่วมกันระหว่างแอปพลิเคชันเอเอสพีกับข้อมูลของไคลเอนต์ที่เรียกใช้งาน

ในการใช้ข้อมูลร่วมกันระหว่างแอปพลิเคชันเอเอสพีกับไคลเอนต์ เราจะสามารถสร้างตัวแปรขึ้นมา ซึ่งตัวแปรที่สร้างขึ้นมานั้นจะเก็บข้อมูลไปจนกว่าแอปพลิเคชันเอเอสพีนั้นจะหยุดทำงานหรือปิดเซิร์ฟเวอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### เมธอดของออบเจกต์แอปพลิเคชัน

<i>Lock</i>	เป็นเมธอดที่ใช้ป้องกันไม่ให้ไคลเอนต์เปลี่ยนแปลงข้อมูลใดๆ ที่เก็บไว้ในออบเจกต์แอปพลิเคชัน ซึ่งมักจะเป็นช่วงที่เราต้องการอัปเดตข้อมูลบางอย่างในออบเจกต์แอปพลิเคชัน
<i>UnLock</i>	จะทำหน้าที่ตรงกันข้ามกับเมธอด <i>Lock</i> ซึ่งจะยอมให้ออบเจกต์แอปพลิเคชันสามารถถูกเปลี่ยนแปลงข้อมูลได้จากไคลเอนต์ที่ใช้งาน

### คอลเล็กชันของออบเจกต์แอปพลิเคชัน

ออบเจกต์แอปพลิเคชันมีคอลเล็กชันคอนเท็นต์ส ซึ่งมีแนวคิดการทำงานเหมือนกับออบเจกต์เสตชัน เพียงแต่จะมีขอบเขตการทำงานทั้งแอปพลิเคชัน คือ ตัวแปรต่างๆที่อยู่ในคอลเล็กชันคอนเท็นต์สของออบเจกต์แอปพลิเคชันจะมีอายุจนกว่าแอปพลิเคชันจะสิ้นสุดการทำงาน (มักจะเป็นเวลาที่ปิดเว็บเซิร์ฟเวอร์) หรือมีการคอมไพล์ไฟล์ *Global.asa* ใหม่

#### อีเวนต์ของออบเจกต์แอปพลิเคชัน

ทั้งสองอีเวนต์ของออบเจกต์แอปพลิเคชันจะเป็นเหตุการณ์ที่เกิดขึ้นเมื่อแอปพลิเคชันเริ่ม และสิ้นสุดการใช้งานซึ่งเรามักจะเพิ่มเติมโค้ดเพื่อจัดการแอปพลิเคชันเอาไว้ทั้งสองอีเวนต์ ได้แก่

*Application\_OnStart* เป็นเหตุการณ์ที่เกิดขึ้นเมื่อแอปพลิเคชันเอสพี เริ่มทำงานบนเซิร์ฟเวอร์ เราจะใช้อีเวนต์นี้ในการกำหนดค่าเริ่มต้นให้กับตัวแปร คุณสมบัติและคอลเล็กชันต่างๆ ที่ต้องใช้งานกันในแอปพลิเคชัน

*Application\_OnEnd* เป็นเหตุการณ์ที่เกิดขึ้นเมื่อแอปพลิเคชันเอสพี จบการทำงาน

### 15.2.5 ออบเจกต์เซิร์ฟเวอร์

ออบเจกต์เซิร์ฟเวอร์ทำหน้าที่ติดต่อ และจัดการกับตัวเซิร์ฟเวอร์ที่ทำงานอยู่

#### คุณสมบัติของออบเจกต์เซิร์ฟเวอร์

*ScriptTimeout* เป็นเวลาสูงสุดที่ยอมให้แอปพลิเคชันเอสพีประมวลผล (ปกติกำหนดไว้ 90 วินาที) เพราะถ้าหากใช้เวลามากกว่านี้ จะทำให้การตอบสนองผู้ใช้อย่างไม่มีประสิทธิภาพ เป็นการใช้ทรัพยากรของระบบมากเกินไปแล้ว ดังนั้นถ้าเกินเวลาที่กำหนดให้ก็ให้จบการทำงาน และรายงานข้อผิดพลาดให้ทราบ

#### เมธอดของออบเจกต์เซิร์ฟเวอร์

<i>MapPath("url")</i>	เป็นเมธอดที่ใช้ขอพาธจริงพร้อมชื่อไฟล์ โดยเราจะต้องระบุพารามิเตอร์เป็นยูอาร์แอล
<i>Execute("url")</i>	เป็นการหยุดการทำงานในเว็บเพจปัจจุบัน แล้วย้ายการทำงานไปยังเว็บเพจอื่นที่ได้ระบุไว้ในพารามิเตอร์ยูอาร์แอล
<i>GetLastError</i>	เป็นการอ้างอิงถึงออบเจกต์เอสพีเออเรอร์ ซึ่งจะเป็นการขอรายละเอียดของข้อผิดพลาดครั้งล่าสุดนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<i>CreateObject</i>	เป็นเมธอดที่สำคัญมาก ซึ่งทำหน้าที่สร้างอินสแตนซ์ของออบเจกต์เอเอสพีขึ้นมาใช้งาน
<i>HTMLEncode</i>	เป็นการเข้ารหัสเอกสารเอชทีเอ็มแอล ทำให้เราสามารถนำเอาแท็กของเอชทีเอ็มแอลไปแสดงผลที่เบราว์เซอร์ได้อย่างถูกต้อง
<i>URLEncode</i>	เป็นการเข้ารหัสข้อความที่ต้องการให้เป็นคิวรีสตริง (QueryString) ซึ่งไปต่อท้ายยูอาร์แอล

### 15.2.6 ออบเจกต์เอเอสพีเอเรอร์

ออบเจกต์เอเอสพีเอเรอร์ เป็นออบเจกต์ตัวใหม่ที่พบในเอเอสพี 3.0 ซึ่งเริ่มทำงานเมื่อเรียกใช้งานเมธอด `GetLastError` ของออบเจกต์เซิร์ฟเวอร์

เมธอดของออบเจกต์เอเอสพีเอเรอร์

<i>ASPCode</i>	เป็นหมายเลขของข้อผิดพลาด (สร้างโดยเว็บเซิร์ฟเวอร์)
<i>ASPDescription</i>	เป็นคำอธิบายของหมายเลขข้อผิดพลาดนั้น
<i>File</i>	เป็นชื่อของไฟล์ที่เป็นตัวกำเนิดของข้อผิดพลาด
<i>Line</i>	เป็นเลขที่บรรทัดที่อยู่ในไฟล์ที่เป็นต้นกำเนิดของข้อผิดพลาด
<i>Source</i>	เป็นส่วนของซอร์สโค้ด ที่ทำให้เกิดข้อผิดพลาด
<i>Category</i>	เป็นประเภทของความผิดพลาดว่ามาจากเอเอสพีเอง มาจากภาษาสคริปต์ หรือมาจากตัวออบเจกต์
<i>Description</i>	เป็นคำอธิบายสั้น ๆ ของความผิดพลาดนั้น

นอกเหนือจากชุดออบเจกต์พื้นฐานทั้ง 6 ชนิดแล้ว บางครั้งเราอาจต้องการออบเจกต์ที่มีคุณสมบัติพิเศษเพื่อเพิ่มเติมให้กับแอปพลิเคชัน หรือ ออบเจกต์ที่ใช้จัดการกับเว็บเซิร์ฟเวอร์ เราก็สามารถทำได้ โดยมี 3 ทางเลือก คือ

- ออบเจกต์แอกทีฟเซิร์ฟเวอร์ (Active Server Object) เป็นชุดออบเจกต์ที่เว็บเซิร์ฟเวอร์เตรียมไว้ให้ใช้นอกเหนือจากออบเจกต์พื้นฐาน 6 ชนิดที่เราได้เรียนรู้ไปแล้ว โดยออบเจกต์แต่ละตัวจะมีความสามารถในการทำงานเฉพาะด้านที่แตกต่างกัน
- คอมโพเนนต์อื่น ๆ (Third Party Component) เป็นออบเจกต์ที่มีบริษัทซอฟต์แวร์อื่นๆสร้างไว้ให้ โดยมีทั้งของฟรีและที่ต้องเสียเงินซื้อ
- ออบเจกต์ที่สร้างขึ้นมาจากซึ่งสามารถใช้ Visual Basic หรือ Visual C++ เขียนคอมโพเนนต์มาให้งานเฉพาะก็ได้ ซึ่งคอมโพเนนต์เหล่านั้นจะอยู่ในรูปของไฟล์ .dll ที่ทำงานบนเว็บเซิร์ฟเวอร์

### 15.3 อินเทอร์เน็ตอินฟอร์เมชันเซอร์วิส (ไอไอเอส)

การให้บริการข้อมูลข่าวสารทางอินเทอร์เน็ตหรืออินเทอร์เน็ตอินฟอร์เมชันเซอร์วิสนับเป็นหัวใจสำคัญของการให้บริการด้านเว็บของไมโครซอฟต์ ซึ่งเป็นหนึ่งในแอปพลิเคชันที่มีวิวัฒนาการเร็วที่สุดเท่า เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เคยมีมา ไอไอเอสมีความยืดหยุ่น สมรรถภาพ และเสถียรภาพ โดยเฉพาะไมโครซอฟต์วินโดวส์ 2000 เซิร์ฟเวอร์ ซึ่งนำไอไอเอสเวอร์ชัน 5 มารวมไว้เป็นคุณลักษณะใหม่ที่จะช่วยเพิ่มเสถียรภาพและความปลอดภัยให้กับเว็บเซิร์ฟเวอร์ มากขึ้น การจัดการสภาพแวดล้อมของการให้บริการเว็บ ออกแบบมาให้ลูกค้าหลายๆคนสามารถใช้งานบนเครื่องเดียวได้ ซึ่งทำให้ได้ประโยชน์สูงสุด

ไอไอเอส 5.0 ได้เพิ่มวิซาร์ดเข้ามารองรับงานบางอย่างที่ซับซ้อนยุ่งยากของไอไอเอส 4.0 ให้สามารถทำงานได้ตามกระบวนการสำเร็จรูปที่เตรียมไว้ให้ โดยนำเซอร์ทิฟิเคทวิซาร์ด (Certificate Wizard) มาแทนที่คีย์แมนเจอร์ (Key Manager) เพื่อสร้างการรับรองแบบเอสเอสแอล และมีเพอร์มิชชันวิซาร์ด (Permission Wizard) ทำให้คุณสามารถใช้วิธีง่าย ๆ มาปรับแต่งรายการ ควบคุมการเข้าถึงบนไฟล์ได้ วิซาร์ดเหล่านี้ทำให้ไอไอเอสใช้งานได้ง่ายขึ้นและกระจายความนิยมนภายในหมู่ผู้ใช้บริการในวงกว้างรวมถึงลดปัญหาที่ผู้ใช้ได้รับมาก่อนลง

เซอร์ทิฟิเคทวิซาร์ดเป็นคุณลักษณะตัวใหม่ที่เข้ามาทดแทนคีย์แมนเจอร์ โดยเฉพาะเรื่องของการใช้งานง่ายและอินเทอร์เฟซที่รวดเร็ว อนุญาตให้ผู้ดูแลระบบแจ้งการร้องขอการรับรองจากเซอร์ทิฟิเคชัน ออร์เทอริที (Certification Authorities) หรือซีเอ (CA) ได้ เช่นจากเวริไซน์ (Verisign) และร้องขอการรับรองจากเซอร์ทิฟิเคทเซอร์วิส (Certificate Services) ซึ่งอาจรันอยู่บนเครือข่ายท้องถิ่น

เอสพีเป็นเทคโนโลยีหนึ่งที่ใช้ไอไอเอสประมวลผลสคริปต์และส่งผลที่ได้จากการทำงานของสคริปต์ออกมาเป็นไฟล์เอชทีเอ็มแอล นักพัฒนาสามารถสร้างโค้ดเอสพีให้มีประสิทธิภาพเพิ่มขึ้นและส่งข้อความแจ้งเตือนความผิดพลาดกลับมาได้

ขีดความสามารถโดยรวมของเอสพีได้รับการปรับปรุงให้ดีขึ้น สามารถพลิกแพลงตัวมันเองเพื่อหลีกเลี่ยงการถูกปิดกั้น โดยโปรเซสเพียงตัวเดียว ซึ่งเวอร์ชันที่แล้วจะทำอยู่ในส่วนของโปรเซสไฟล์เอสพี ไม่เก็บโค้ดของสคริปต์เอสพีที่ถูกโปรเซสเหมือนกับไฟล์เอชทีเอ็มแอล แต่อนุญาตให้นักพัฒนาสร้างเอกสารทั้งหมดโดยมีส่วนขยายเป็น .asp ได้โดยไม่ทำให้ประสิทธิภาพของระบบเสียหายหรือลดลง

## บทที่ 16

### การออกแบบระบบ

#### 16.1 ความต้องการของระบบ

OlalaTour เป็นบริษัทที่ทำหน้าที่ นำเที่ยว จองโรงแรม และตั๋วเครื่องบิน ให้บริการผ่านอินเทอร์เน็ต ผู้ซึ่งใช้งานผ่านทางอินเทอร์เน็ตสามารถ ใช้งานระบบของเราแบ่งเป็นส่วนย่อย ได้ 4 ส่วนคือ

##### 16.1.1 ส่วนกลาง

###### การลงทะเบียนกับระบบ

ในระบบงานของเรานั้นจะมีการลงทะเบียนกับระบบงานเพื่อจะเก็บข้อมูลของผู้ใช้ ขั้นตอนนี้จะเกิดขึ้นจาก เหตุการณ์สอง เหตุการณ์คือ

- ผู้ใช้ทำการกรอกข้อมูลที่จะใช้ในการจองเที่ยวการเดินทาง โรงแรม หรือ ชุดการเดินทาง จะทำการลงทะเบียนตัวผู้จอง (ซึ่งอาจจะเป็นคนเดียวกัน หรือคนละคนกับผู้ใช้ก็ได้ เนื่องจากผู้จองอาจจะฝากผู้ให้มาจองเที่ยวการเดินทางให้แทน) โดยอัตโนมัติ

###### ประโยชน์

- สำหรับติดต่อกับผู้จองในกรณีที่เกิดปัญหา หรือเหตุขัดข้องต่าง ๆ ของการเดินทาง หรือ การจองโรงแรม โดยติดต่อกับผู้จองผ่านทาง ที่อยู่ หรือ สื่อต่าง ๆ ที่ผู้จองระบุไว้ เช่น การแจ้งเลื่อนการเดินทาง โดยทางเครื่องบินไป ประเทศอิรักเนื่องจากปัญหาสงคราม เป็นต้น
  - ส่งข่าวสารใหม่ ให้กับผู้จอง ผ่านทางสื่อ ต่าง ๆ ที่ผู้จองระบุไว้ เช่น แฟ้มเอกสารข่าวสาร ประจำปี 2001
  - เก็บข้อมูลการใช้บริการของผู้จอง เพื่อใช้ในการส่งเสริมการขายบางลักษณะ เช่น สะสมไมล์ของสายการบินไทย เพื่อลดราคาที่พักโรงแรมในเครือ โอเรียนเต็ล เป็นต้น
  - เก็บข้อมูลของผู้จอง เพื่อใช้เป็นข้อมูลในการวิเคราะห์ทางตลาด ในอนาคต เช่น ถามว่ารู้จักเว็บของเราได้อย่างไร, ดิชม หรือ แนะนำ เป็นต้น
- ผู้ใช้เลือกที่จะลงทะเบียนกับระบบงาน เพื่อรับข่าวสารจากทางบริษัท

###### ประโยชน์

- ส่งข่าวสารใหม่ ให้กับผู้ใช้ ผ่านทางสื่อ ต่าง ๆ ที่ผู้ใช้ระบุไว้
- เก็บข้อมูลของผู้ใช้ เพื่อใช้เป็นข้อมูลในการวิเคราะห์ทางตลาด ในอนาคต เช่น ถามว่ารู้จักเว็บของเราได้อย่างไร, ดิชม หรือ แนะนำ เป็นต้น

ข้อมูลที่จะได้รับจากผู้ใช้ในการลงทะเบียนมีดังนี้

- ชื่อ และ นามสกุล \*
- ที่อยู่ (Address)
- หมายเลขโทรศัพท์ (Phone Number)
- อีเมลแอดเดรส (Email Address) \*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพศ
- วัน/เดือน/ปีเกิด
- ศาสนา
- รหัสผ่าน

หมายเหตุ ในหัวข้อที่มีเครื่องหมาย \* หมายถึง ต้องระบุ

#### การพิสูจน์ตน

ในการที่ผู้ใช้จะยืนยันการจอง, ยกเลิก ผู้ใช้จะติดต่อเข้าระบบโดยใช้อีเมลล์แอดเดรส คู่กับรหัสผ่านที่ตนเองตั้งไว้

#### การจองตั๋วเดินทาง

- ระบุรายละเอียดของข้อมูลต่างๆเพื่อทำการค้นหา สายการบิน และ รถบัส
- จองตั๋วเดินทาง ของรถบัส และสายการบิน
- จัดเส้นทางการเดินทาง ตามเวลาที่กำหนด
- ปรับแต่ง รายละเอียดของการเดินทาง
- ยกเลิก หรือยืนยัน การจองเที่ยวเดินทาง

#### ระบุรายละเอียดของข้อมูลต่างๆเพื่อทำการค้นหา สายการบิน และ รถบัส

เป็นขั้นตอนแรกๆที่ผู้ใช้จะเริ่มใช้บริการจากระบบงาน ไม่ว่าจะผู้ใช้จะรู้รายละเอียดที่แน่นอนในระดับหนึ่งก็ตาม เมื่อผู้ใช้ทำการกรอกรายละเอียดที่ต้องการหา แล้วผลลัพธ์ที่ออกมา จะเป็นรายละเอียดต่าง ๆ ของเที่ยวการเดินทาง ที่เกี่ยวข้องกับ คีย์เวิร์ดที่ผู้ใช้ป้อนเข้าไป ผู้ใช้สามารถเลือกในแต่ละเที่ยวการเดินทาง เพื่อดูรายละเอียด เพิ่มเติม หรือปรับแต่งรายละเอียดของเที่ยวการเดินทางนั้น ได้ การค้นหาข้อมูลเที่ยวบินมีข้อมูลที่สามารถใช้เป็นเงื่อนไขได้ดังนี้

- สถานที่ต้นทาง
- สถานที่ปลายทาง
- ตัวเลือกว่าต้องการเดินทาง ไปอย่างเดียว หรือ ไป-กลับ
- วันที่ เวลา ของการออกเดินทาง ไป
- หากเลือกการเดินทางไปกลับสามารถระบุ วันที่ เวลา ของการเดินทาง
- ชนิดของที่นั่ง
- บริษัทขนส่งที่ต้องการใช้บริการ
- การเรียงลำดับจากตัวเลือกต่างๆที่ค้นหาเจอ เช่น เรียงจากราคาที่ต่ำสุด, ใช้เวลาการเดินทางน้อยที่สุด

โดยผลลัพธ์จากการค้นหาจะมีรายละเอียดของสายการบิน ดังนี้

- ชื่อเที่ยวบิน
- ชื่อสายการบิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สถานีต้นทาง – ปลายทาง
- เวลาขึ้นเครื่อง/เวลาลงเครื่อง
- ราคา

การค้นหาข้อมูลรถบัสมีดังนี้

ขั้นตอนนี้จะทำงานเมื่อผู้ใช้เลือกรายการค้นหาสายการเดินรถ โดยผู้ใช้จะป้อนคุณสมบัติโดยคร่าว ๆ ของสายการเดินรถที่ตนต้องการจะใช้งาน

โดยรายละเอียดข้อมูลในการค้นหารถบัสมีดังนี้

- สถานีต้นทาง \*
- สถานีปลายทาง \*
- ตัวเลือกว่าต้องการเดินทางไปอย่างเดียว หรือ ไป-กลับ
- วันที่ เวลา ของการออกเดินทางไป
- หากเลือกการเดินทางไปกลับสามารถระบุ วันที่ เวลา ของการเดินทาง
- ชนิดของที่นั่ง
- ชื่อบริษัทที่ต้องการใช้บริการ
- การเรียงลำดับจากตัวเลือกต่างๆที่ค้นหาเจอ เช่น เรียงจากราคาที่ต่ำสุด, ใช้เวลาการเดินทางน้อยที่สุด

ผลลัพธ์จากการค้นหาจะมีรายละเอียดรถบัส มีดังนี้

- ชื่อบริษัทขนส่ง
- ชื่อสายการเดินรถ
- สถานีต้นทาง
- สถานีปลายทาง
- เวลาารถออก
- เวลาถึงที่หมาย
- ราคา

การจองโรงแรม และที่พัก

- ระบุรายละเอียดของข้อมูลต่างๆเพื่อทำการค้นหา ที่พัก
- จองที่พัก
- เปลี่ยนแปลงรายละเอียดของที่พัก.
- ยกเลิก หรือ ยืนยันการจองที่พัก

ระบุรายละเอียดของข้อมูลต่างๆเพื่อทำการค้นหา ที่พัก

โดยรายละเอียดข้อมูลในการค้นหา ที่พัก มีดังนี้

1. วันที่,เวลา check in – check out
2. จำนวนคนที่เข้าพัก
3. จำนวนห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์จากการค้นหาจะมีรายละเอียด ที่พัก ดังนี้

จะขึ้นเป็นรายละเอียดโดยย่อของแต่ละที่พักซึ่งมีรายละเอียดโดยย่อต่างๆดังนี้

1. ชื่อที่พัก
2. ที่อยู่
3. เบอร์โทรศัพท์
4. อัตราค่าบริการ

### จองที่พัก

จะเกิดหลังจากที่ผู้ใช้ได้ ทำการตรวจสอบรายละเอียด และค่าใช้จ่าย เมื่อผู้ใช้เลือกรายการที่ต้องการได้ แล้ว ผู้ใช้อาจจะทำการปรับแต่งคุณสมบัติของที่พัก จนเป็นที่พอใจ แล้วจึงถึงขั้นตอนการ จองที่พัก

### ปรับแต่งรายละเอียดของที่พักที่ต้องการ

เป็นขั้นตอนที่ผู้ใช้จะเลือกคุณสมบัติ ต่างๆ ของที่พักไม่จำเป็นที่ผู้ใช้จะต้องเข้ามาใช้บริการในขั้นตอน นี้ ขั้นตอนนี้จะเกิดขึ้นหลังจากที่ผู้ใช้ได้เลือกที่พักแล้ว โดยจะแสดงข้อมูลอย่างละเอียดของที่พักในบางตัว เลือกที่เราเลือกให้โดยอัตโนมัติ ผู้ใช้อาจไม่เป็นที่พอใจ จึงอาจปรับเปลี่ยน หรือ แก้ไขคุณสมบัติบางอย่าง หลังจากนั้นจึงเป็นขั้นตอนในการจองที่พักต่อไป ซึ่งข้อมูลที่แสดงในรายละเอียดของที่พักมีดังนี้

- จำนวนเตียง(เช่น เตียงเดี่ยว, เตียงคู่)
- ตำแหน่งห้อง(เช่น มุมตึก,มองเห็นทะเล)

### การซื้อชุดเดินทาง(package)

- ระบุรายละเอียดของข้อมูลต่างๆเพื่อทำการค้นหา ชุดเดินทาง
- จองชุดเดินทาง
- ปรับแต่งรายละเอียดของชุดเดินทาง
- ยกเลิก หรือ ยืนยันการซื้อชุดเดินทาง

### รายละเอียดของข้อมูลต่างๆเพื่อทำการค้นหา ชุดเดินทาง

โดยรายละเอียดข้อมูลในการค้นหา ชุดเดินทาง มีดังนี้

- ค้นหาสถานที่ที่จะไปจาก
  - 1) ค้นหาจากชื่อสถานที่
  - 2) ค้นหาจากชื่อประเทศ
  - 3) ค้นหาจากชื่อทวีป
  - 4) วันที่ เวลา ของการออกเดินทางไป และกลับ
- ชนิดของอัตรา package / 1 คน
- ชื่อเทศกาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ยกเลิก หรือ ยืนยันการซื้อชุดเดินทาง ดูในหัวข้อการพิสูจน์ตน

### หมายเหตุ

1. ผู้จองต้องกรอกข้อมูลทุกครั้งที่จอง
2. ผู้จองมี e-mail address เนื่องจากจะใช้ e-mail address ที่มีอยู่นั้นเป็น LoginName

### 16.1.2 ธนาคาร

ทำหน้าที่เก็บรหัสบัตรเครดิต และยอดเงิน รวมถึงรายละเอียดต่างๆ ของผู้ใช้ ระบบงานจะติดกับธนาคาร โดยมีลักษณะ ดังนี้

- 1) ในกรณีที่ลูกค้าจองตั๋ว หรือชุดการเดินทาง เพื่อทำการโอนเงินจากบัญชีในธนาคารของลูกค้า มาบัญชีในธนาคารของระบบงานของเรา ซึ่งเราก็จะต้องโอนเงินจากบัญชีของเราไปให้กับ สายการบิน บริษัทเดินรถ และ โรงแรมอีกที่หนึ่ง
- 2) กรณีที่ลูกค้ายกเลิก ต้อง โอนเงินคืนให้กับ ลูกค้า

### บริการของธนาคาร

เนื่องจากธนาคารเป็นองค์ประกอบย่อยของระบบ เราจะไม่ลงลึกในรายละเอียดของธนาคาร เพียงแต่จะทำเฉพาะในส่วนที่เราต้องติดต่อเท่านั้นงานบางอย่างของธนาคาร เราก็จะเข้าไป (เช่น การคิดดอกเบี้ย เป็นต้น) ซึ่งสามารถแบ่งบริการของธนาคารได้เป็น 2 ส่วนคือ

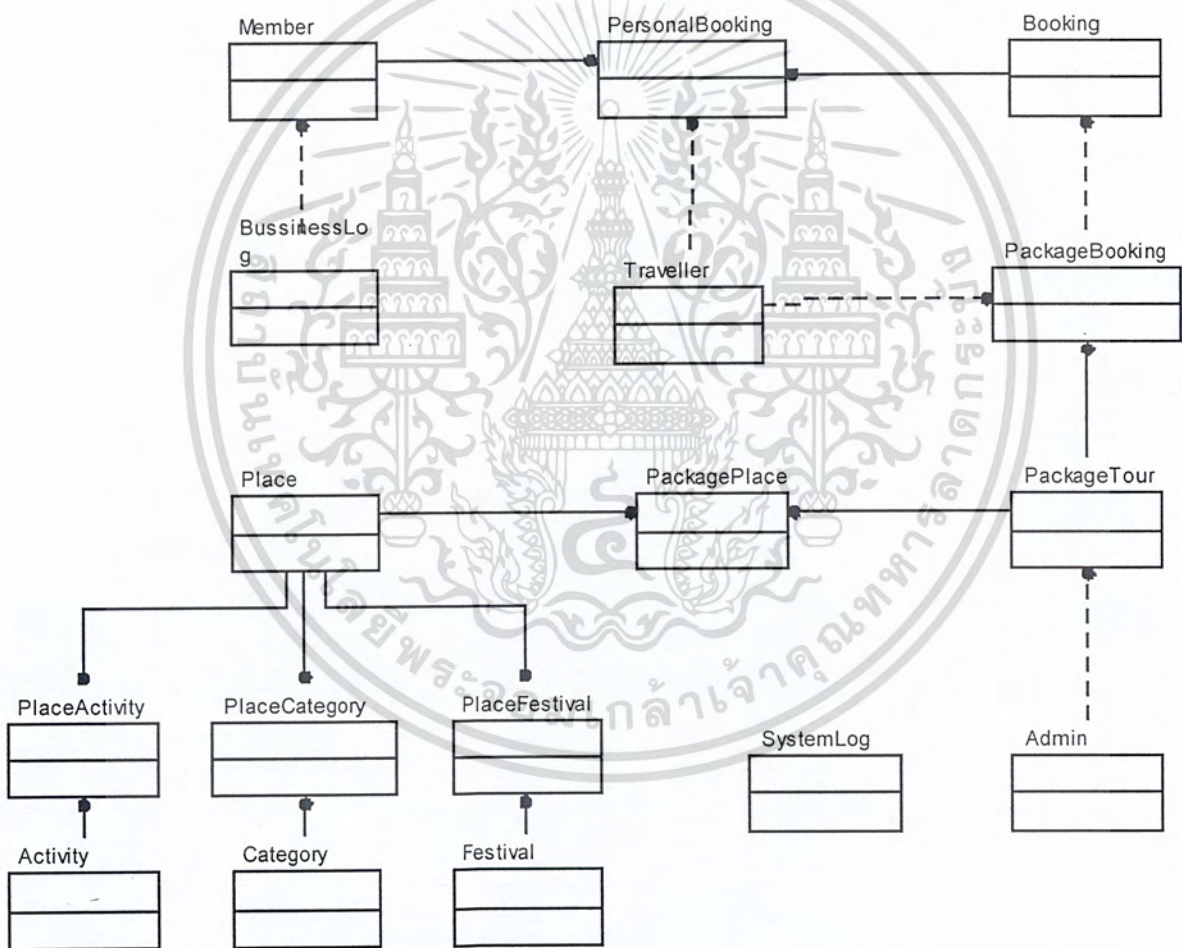
- 1) ส่วนที่ติดต่อกับผู้ใช้โดยตรง  
บริการที่สนใจคือ  
- โอนเงิน
- 2) ส่วนที่ติดต่อกับผู้ใช้โดยอ้อม  
บริการที่สนใจคือ  
ฝากเงิน  
ถอนเงิน

## 16.2 การออกแบบระบบงาน OlalaTour

ในการออกแบบระบบงาน OlalaTour โดยนำความต้องการของระบบซึ่งได้กล่าวไว้ในหัวข้อที่ผ่านมา โดยทำการออกแบบฐานข้อมูลเชิงสัมพันธ์โดยใช้ ER Diagram ใช้ DFD (Data Flow Diagram) ในการอธิบายการไหลของขั้นตอนการทำงาน (Process) และใช้ UML (Unified Modeling Language) บางส่วนในการออกแบบ Use Case, Class Diagram และ Component Diagram ซึ่งรายละเอียดในการออกแบบระบบ OlalaTour มีดังนี้

การออกแบบฐานข้อมูลโดยใช้ ER Diagram

จากความต้องการของระบบสามารถจัดวางข้อมูลไว้ในตารางฐานข้อมูล ที่มีความสัมพันธ์กันได้ ดังรูปที่ 16-1 ซึ่งมีตารางฐานข้อมูลทั้งหมด 17 ตาราง



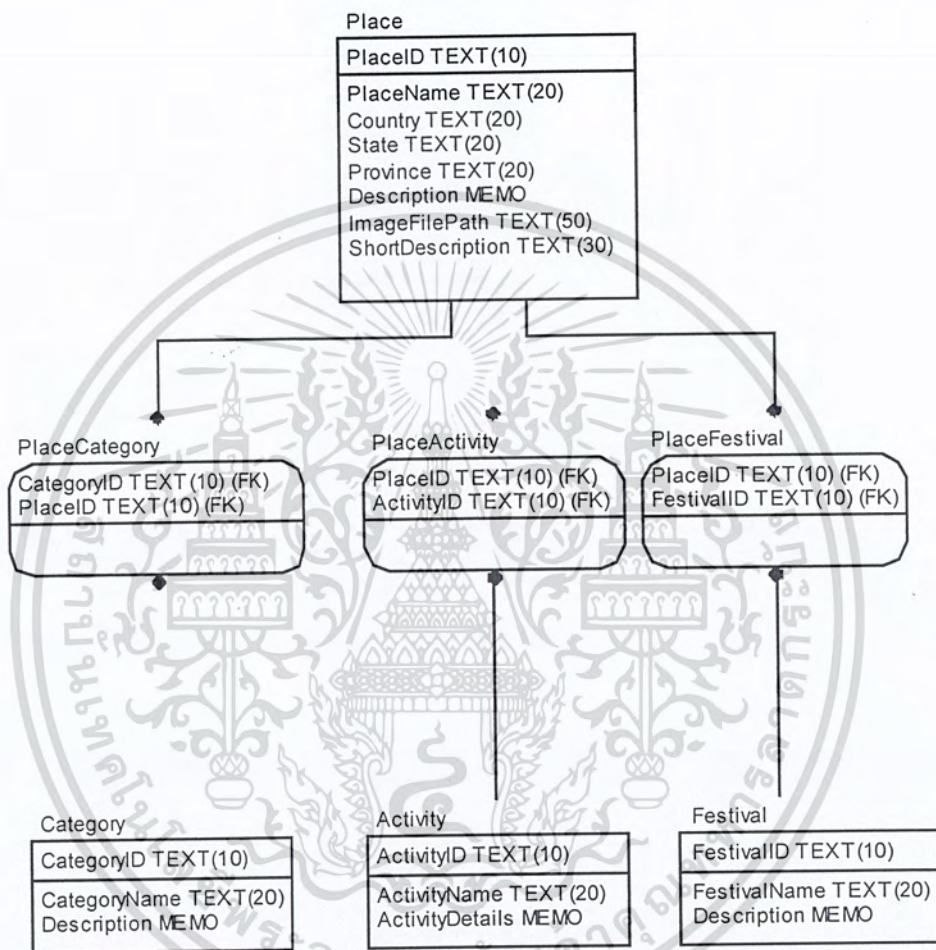
รูปที่ 16-1 แสดงตารางฐานข้อมูลและความสัมพันธ์ระหว่างตารางต่าง ๆ ในระบบ OlalaTour

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งระบบนี้สามารถแบ่งออกเป็น ส่วนย่อย 3 ส่วนเพื่อความสะดวกในการอธิบายได้ดังนี้

1) ส่วนของสถานที่ และ คุณสมบัติของสถานที่

ในส่วนนี้จะเป็นส่วนของตารางฐานข้อมูลจากรูปภาพ 16-1 ซึ่งจะมีจำนวน 7 ตารางดังรูปที่ 16-2



รูปที่ 16-2 แสดงความสัมพันธ์และแอททริบิวต์ของตารางฐานข้อมูล ส่วน สถานที่ ของระบบ OlalaTour

ความสัมพันธ์ และความหมายของตารางฐานข้อมูล ต่าง ๆ มีดังนี้

ตาราง Activity อธิบาย กิจกรรมที่จะทำในสถานที่ต่าง ๆ เช่น การปั่นเขา, การล่องแพ เป็นต้น

ตาราง Category อธิบายถึงลักษณะของสถานที่ เช่น เป็นชายหาด, ป่าดงดิบ เป็นต้น

ตาราง Festival อธิบายงานเทศกาลประจำท้องถิ่นต่าง ๆ ที่มีชื่อเสียง เช่น ลอยกระทง เป็นต้น

ตาราง PlaceActivity, PlaceCategory และ PlaceFestival อธิบายถึงความสัมพันธ์ระหว่าง สถานที่ต่าง กับ กิจกรรม, ลักษณะ และ งานเทศกาลตามลำดับ

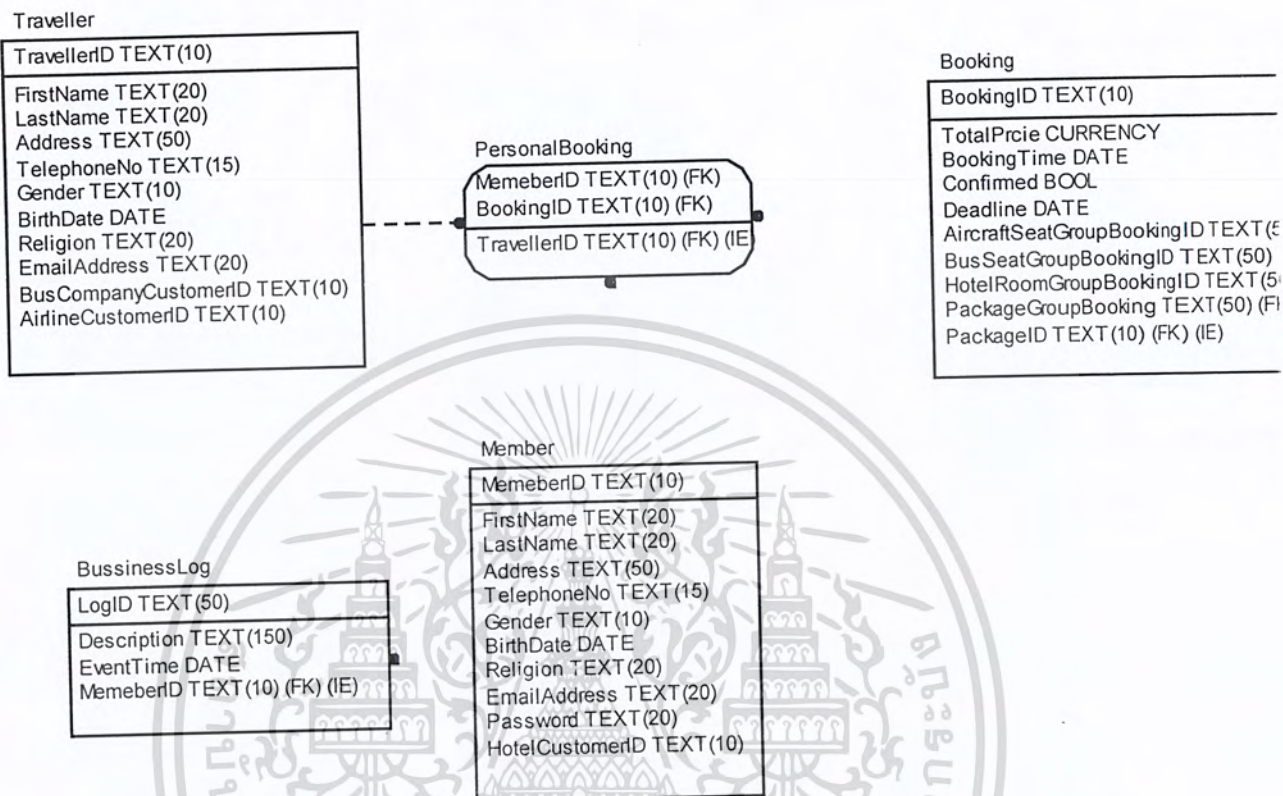
ตาราง Place อธิบายถึงตำแหน่งที่ตั้ง และชื่อของสถานที่ต่าง ๆ โดยตารางนี้จะมีความ

สัมพันธ์กับตาราง PackagePlace ซึ่งจะกล่าวถึงในส่วน ชุดการท่องเที่ยว ของระบบ OlalaTour

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2) ส่วนการจอง และ ลูกค้า

ในส่วนนี้จะมีตารางฐานข้อมูลทั้งหมด 5 ตาราง ดังรูปที่ 16-3



รูปที่ 16-3 แสดงความสัมพันธ์ และ แอททริบิวต์ของระบบ การจองและลูกค้า ของระบบ OlalaTour

ความสัมพันธ์ และความหมายของตารางฐานข้อมูลต่าง ๆ มีดังนี้

ตาราง Traveler อธิบายถึงข้อมูลของผู้เดินทางในการจองตั๋วเดินทางแต่ละครั้งผู้จองจะเป็นสมาชิกของระบบ OlalaTour ซึ่งอาจจะเป็นบุคคลเดียวกันกับ ผู้เดินทางก็ได้โดยในการจองที่นั่งของรถโดยสาร และเครื่องบินจะสร้างรหัสของลูกค้าให้กับผู้เดินทาง

ตาราง Member อธิบายถึงข้อมูลของสมาชิก ในระบบ OlalaTour นอกจากนั้นยังอธิบายถึงรหัสสมาชิกของโรงแรมด้วยเนื่องจากการจองห้องพักจะให้สมาชิกจองห้องพักโดยไม่ต้องใส่ข้อมูลผู้เดินทาง

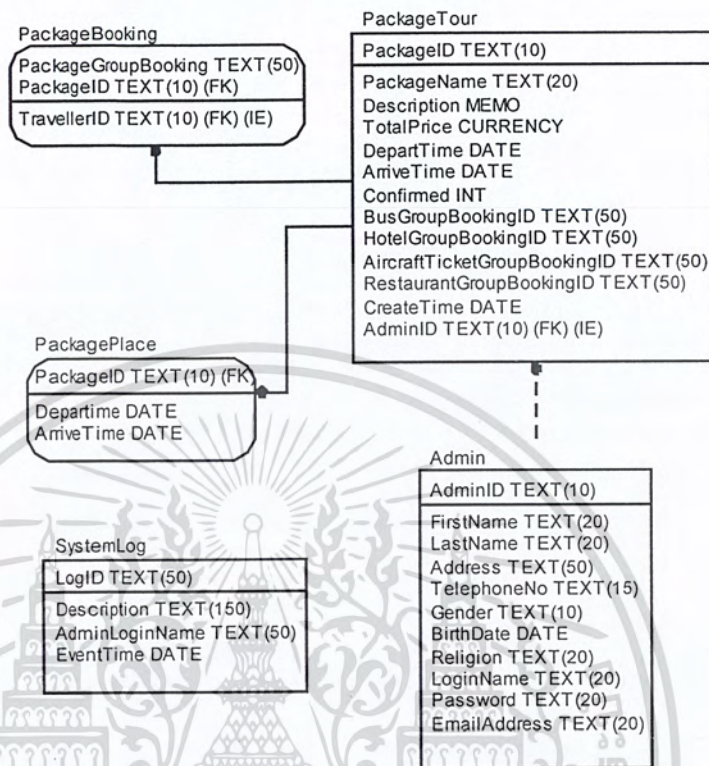
ตาราง PersonalBooking อธิบายถึงความสัมพันธ์ระหว่างการจอง กับ ผู้เดินทาง และ สมาชิก โดยการจองครั้งหนึ่ง นั้น สมาชิกจะทำการจอง การเดินทาง และที่พัก ให้กับผู้เดินทางได้หลาย ๆ คน

ตาราง Booking อธิบายความสัมพันธ์ระหว่างการจองแต่ละครั้งกับกลุ่มการจองต่าง ๆ

ตาราง BussinessLog ใช้ในการบันทึกเหตุการณ์ของต่าง ๆ ของลูกค้าให้กับระบบซึ่งผู้ดูแลระบบ OlalaTour สามารถจะมาตรวจสอบได้ภายหลัง โดยใช้โปรแกรม OlalaTourAdmin ซึ่งจะได้อีกภายในภาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3) ส่วนชุดการท่องเที่ยว



รูปที่ 16-4 แสดงความสัมพันธ์ และแอททริบิวต์ ของส่วน ชุดการเดินทางของระบบ OlalaTour

ความสัมพันธ์ และความหมายของตารางฐานข้อมูลต่าง ๆ มีดังนี้

ตาราง PackageTour อธิบายถึงข้อมูลต่าง ๆ ของตัวชุดการเดินทาง รวมถึงรหัสซึ่งอ้างอิงไปยังการจองที่พัก ร้านอาหาร รถโดยสาร และ ที่นั่งเครื่องบินที่จะใช้ในการสร้างชุดการเดินทาง

ตาราง PackageBooking อธิบายถึงรายละเอียดของการจองชุดการเดินทางว่าในการจองหนึ่งครั้งนั้นสมาชิกได้จองให้กับผู้เดินทางคนไหนบ้าง

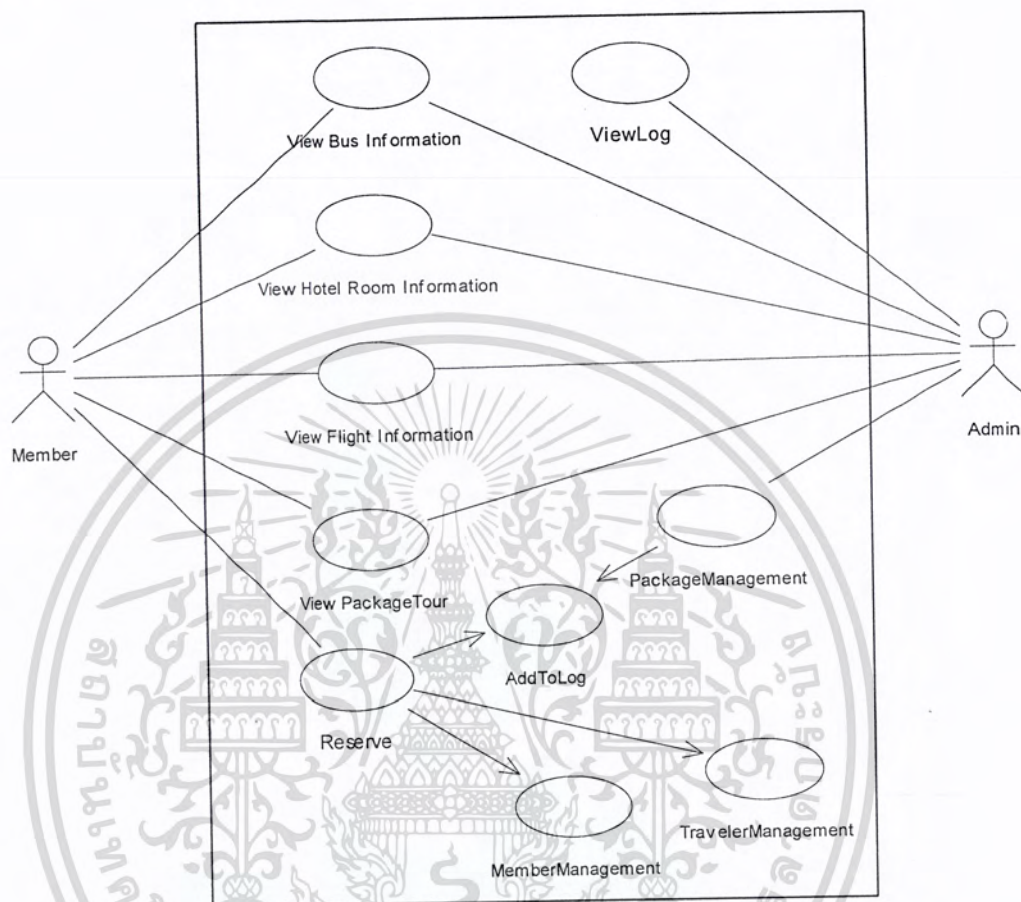
ตาราง PackagePlace อธิบายถึงสถานที่ที่ชุดการเดินทางนั้นจะเดินทางไปรวมถึงบอกเวลาไปถึงและเวลากลับจากสถานที่เหล่านั้น

ตาราง Admin แสดงถึงรายละเอียดของผู้ดูแลระบบ

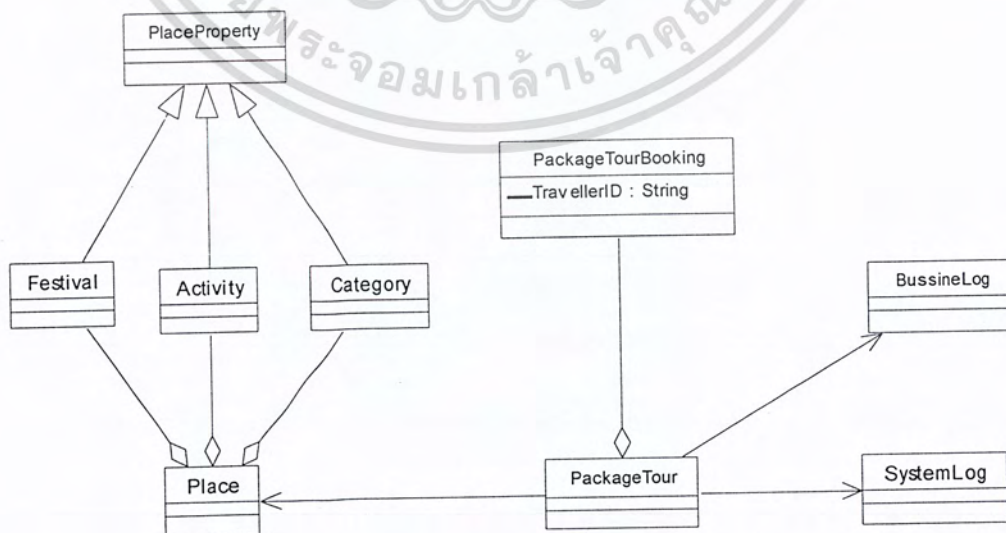
ตาราง SystemLog บันทึกรายละเอียดเหตุการณ์ที่เกี่ยวข้องกับการ เพิ่มเติม แก้ไขข้อมูลภายในระบบ เช่นการเพิ่ม ผู้ดูแลใหม่ สร้าง ชุดการเดินทางใหม่ เป็นต้น

16.3 การสร้าง Use Case Diagram และ แผนภาพคลาส

จากความต้องการของระบบสามารถสร้าง Use Case Diagram และแผนภาพคลาสได้ดังรูปที่ 16-5 ถึงรูปที่ 16-8

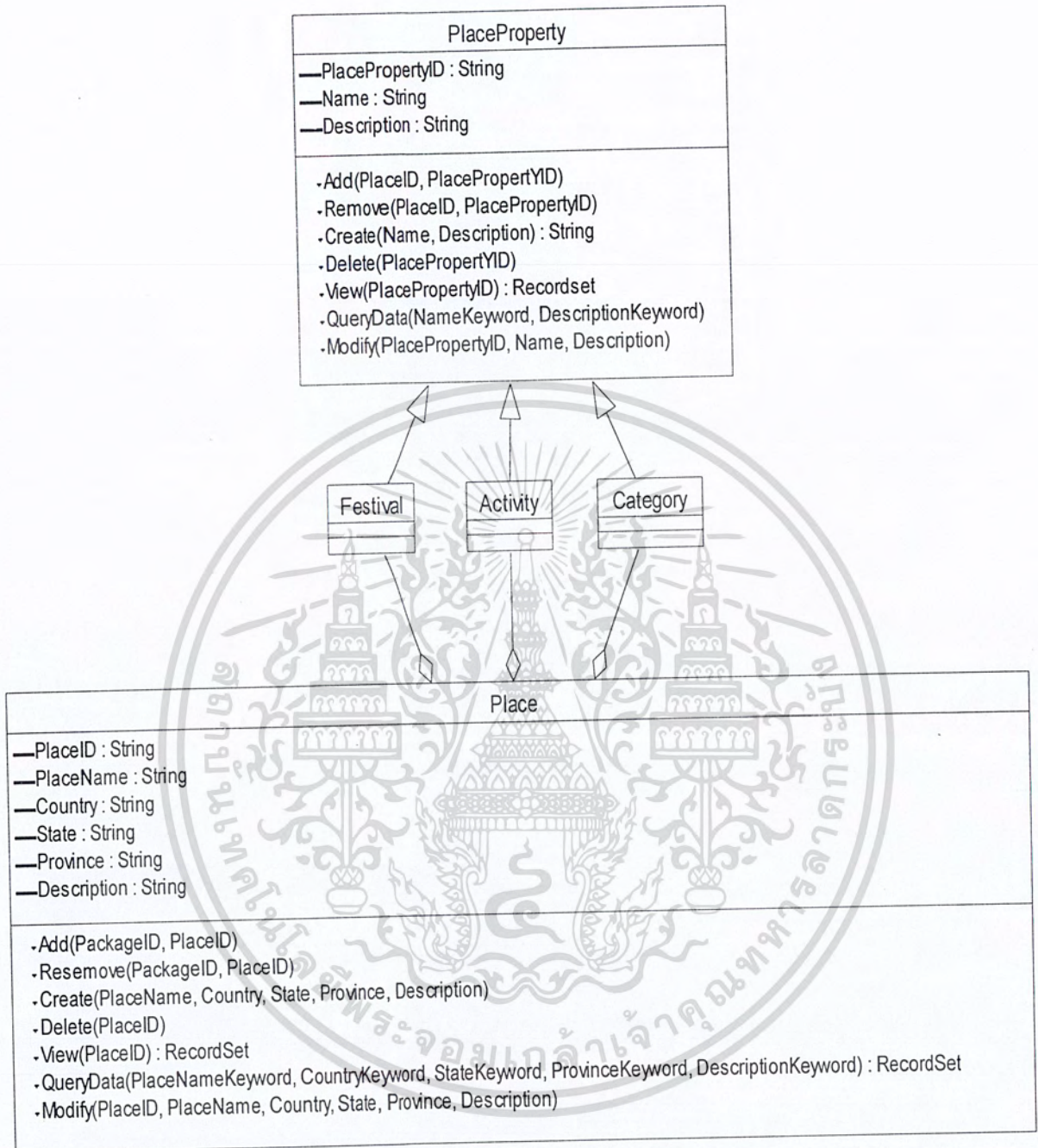


รูปที่ 16-5 แสดง Use Case Diagram ของระบบ OlalaTour



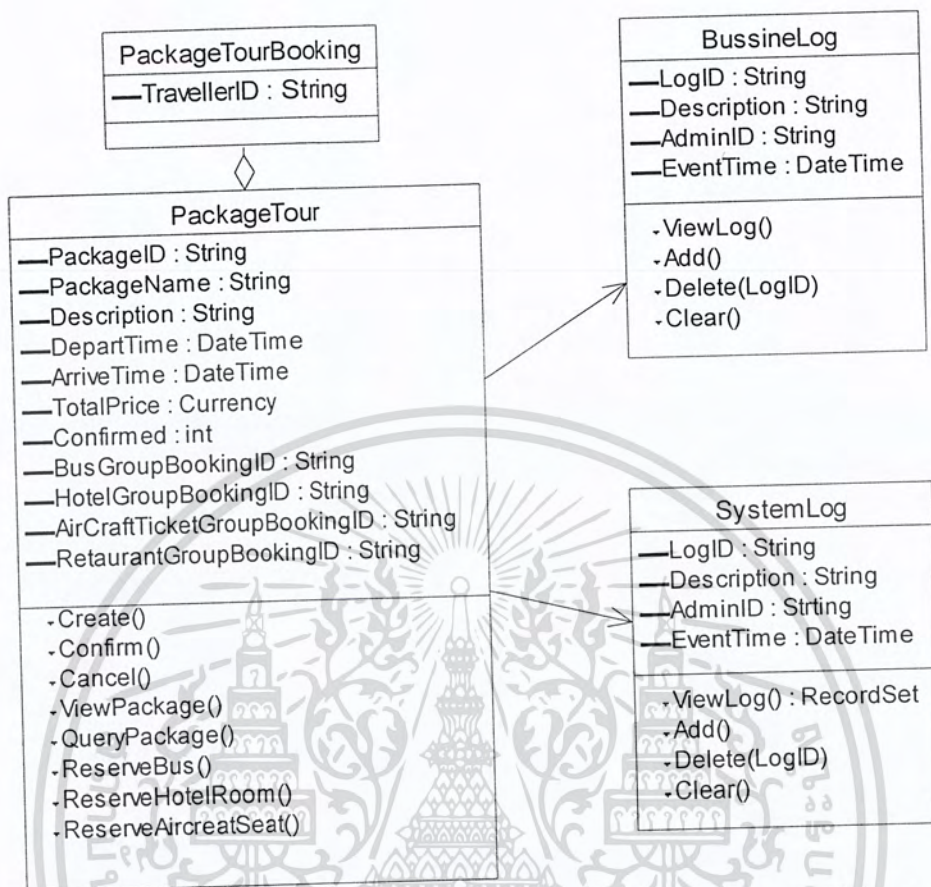
รูปที่ 16-6 แสดงแผนภาพคลาส ของระบบ OlalaTour ในส่วน ของสถานที่, ชุดการท่องเที่ยว และ Log ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 16-7 แสดงเมธอด และแอททริบิวต์ของคลาสในส่วนที่เกี่ยวข้องกับสถานที่โดยละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 16-8 แสดงรายละเอียดของเมธอด และแอททริบิวต์ของคลาสในส่วนที่เกี่ยวข้องกับชุดการ  
ท่องเที่ยว และ Log

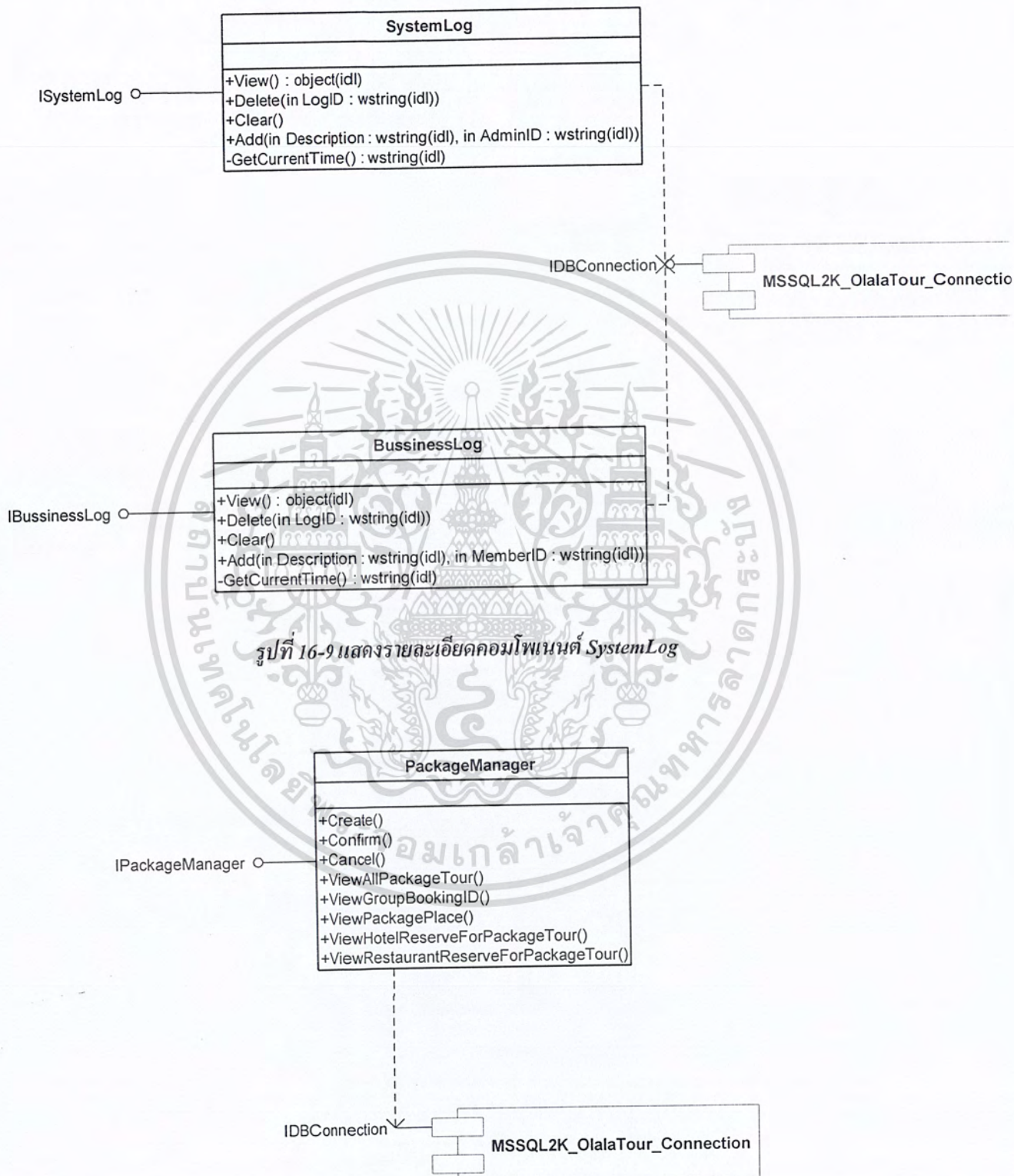
#### 16.4 การสร้างแผนภาพคอมโพเนนต์ (Component Diagram)

ในการสร้างแผนภาพคอมโพเนนต์นั้นเป็นขั้นตอนสุดท้ายในการออกแบบระบบ ซึ่งในการออกแบบแผนภาพคอมโพเนนต์นี้จะแตกต่างจากการออกแบบระบบในหัวข้อก่อนหน้าที่ในการออกแบบฐานข้อมูล หรือ ออกแบบแผนภาพคลาส นั้นไม่จำเป็นต้องสนใจต่อผลกระทบที่จะเกิดขึ้นกับการพัฒนาระบบงาน แต่ การออกแบบแผนภาพคอมโพเนนต์นี้ จำเป็นต้องมีการปรับเปลี่ยนแผนภาพคลาส ไปเล็กน้อยเนื่องจากข้อจำกัดในการที่จะให้ คอมโพเนนต์เหล่านั้นทำงานใน มิดเดิลเทียร์ (Middle-tier) ด้วย COM+ เนื่องจาก การสร้างคอมโพเนนต์ที่อยู่ในเทียร์ นี้มันควรจะไม่มีสถานะ หรือแอททริบิวต์ เพราะจะทำให้เกิดปัญหาในการทำ Object-Pooling และ Just-In-Time Activation ได้

ทางแก้ปัญหาเหล่านี้มีได้หลายวิธี เช่นการใช้ อินเตอร์เฟส ในกลุ่ม IPersist แต่ในที่นี้ จะใช้วิธีการเก็บสถานะต่าง ๆ ของคอมโพเนนต์ไว้ที่ แหล่งเก็บข้อมูลถาวร (Persistence Storage) โดยจะใช้การจัดเก็บลงในฐานข้อมูลซึ่งได้ออกแบบมาแล้วในส่วนการออกแบบฐานข้อมูลเชิงสัมพันธ์โดยใช้ ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการออกแบบ แพคเกจคอมโพเนนต์จากแผนภาพคลาสนั้นในที่นี้จะใช้การรวบรวมคลาสที่มีความเกี่ยวข้องกันและมีพฤติกรรมไปในทางเดียวกันรวมกลุ่มกันเป็น หนึ่งคอมโพเนนต์ ซึ่งสามารถแบ่งได้เป็น 3 ตัวคือ PlaceSystem, PackageSystem และ LogSystem ซึ่งมีรายละเอียดดังรูปที่ 16-9 ถึง 11



รูปที่ 16-9 แสดงรายละเอียดคอมโพเนนต์ SystemLog

รูปที่ 16-10 แสดงรายละเอียดคอมโพเนนต์ PackageSystem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 17

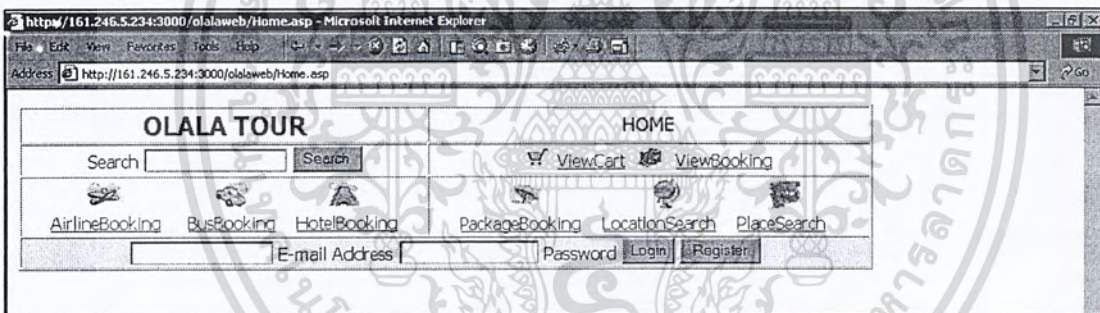
## ลักษณะแอปพลิเคชัน

## 17.1 ระบบงาน Olala Tour บนอินเทอร์เน็ต

Olala Tour บนอินเทอร์เน็ตเป็นลักษณะการทำธุรกิจเชิงพาณิชย์ที่ทำผ่านระบบเครือข่ายอินเทอร์เน็ต โดยที่ลูกค้าสามารถเข้าไปเลือกชมสินค้าและบริการต่างๆผ่านเว็บเบราว์เซอร์(Web Browser) ที่สามารถเข้าใช้บริการได้ตลอดเวลา เมื่อลูกค้าตกลงใจเลือกจองสินค้าแล้ว ก็ทำการตกลงชำระเงินผ่านสื่อต่างๆ เช่น เครดิตการ์ด เดบิตการ์ด การชำระเงินผ่านบัญชีธนาคาร เป็นต้น เมื่อผู้ให้บริการได้รับหลักฐานทุกอย่างครบถ้วนแล้วก็จะทำการจัดส่งสินค้าหรือให้บริการด้านนั้นๆแก่ลูกค้าก็เป็นอันสิ้นสุดขบวนการทำงาน

ในปัจจุบันเครือข่ายอินเทอร์เน็ตได้มีใช้กันอย่างแพร่หลายมากขึ้น โดยเพื่อให้เกิดการทำงานนี้เกิดขึ้นต้องอาศัยเทคนิคในหลายรูปแบบทั้งในด้านฮาร์ดแวร์และซอฟต์แวร์ ซึ่งโปรเจกต์นี้จะเกี่ยวกับการโปรแกรมโดยอาศัยเทคโนโลยีด้าน COM ,DCOM ,DNS ,DNA ,MTS ,COM+ ,Active Directory ,IIS ,ASP เป็นหลักสำคัญในการประมวลผลในการทำงานของฟังก์ชันต่างๆและในการติดต่อฐานข้อมูลของทั้ง ระบบOlala Tour ระบบสายการบิน ระบบรถบัส ระบบ โรงแรมและลูกค้า

## 17.1.1 หน้าจอหลัก



รูปที่ 17-1 หน้าจอหลัก

ที่หน้าจอหลักจะมี link ไปที่การให้บริการอื่นๆคือ

- AirlineBooking เป็นการค้นหาสายการบินที่เราต้องการ
- BusBooking เป็นการค้นหาเที่ยวการเดินทางที่เราต้องการ
- HotelBooking เป็นการค้นหาโรงแรมที่เราต้องการ
- PackageBooking เป็นการค้นหาชุดการท่องเที่ยวที่เราต้องการ
- LocationSearch และ PlaceSearch เป็นการค้นหาสถานที่ที่เราต้องการ
- ViewCart เป็นการตรวจดูรถเข็นที่เราใส่สินค้าไว้
- ViewBooking เป็นการตรวจดูสินค้าที่เราทำการจองไว้
- Login เป็นการเข้าสู่ระบบ
- Register เป็นการสมัครสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 17.1.2 การสมัครเป็นสมาชิก

http://161.246.5.234:3000/olalaweb/MemberInfo.asp - Microsoft Internet Explorer

Address http://161.246.5.234:3000/olalaweb/MemberInfo.asp

**OLALA TOUR** **MEMBER INFORMATION**

Search  Search

[ViewCart](#) [ViewBooking](#) [Home](#)

[AirlineBooking](#) [BusBooking](#) [HotelBooking](#) [PackageBooking](#) [LocationSearch](#) [PlaceSearch](#)

**Fill Information for New Member**

Firstname\*

Lastname\*

E-mail\*

Password\*

Re-Password\*

Address

TelephoneNo

Gender

BirthDate\*

Religion

รูปที่ 17-2 การสมัครสมาชิก

**OLALA TOUR** **HOME**

Search  Search

[ViewCart](#) [ViewBooking](#) [Home](#)

[AirlineBooking](#) [BusBooking](#) [HotelBooking](#) [PackageBooking](#) [LocationSearch](#) [PlaceSearch](#)

**Weather**

London,UK 30-34.5F

Bangkok,TH 21-25F

Tokyo,Ja 5-10F

**Calendar**

May 2001  2001

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		
1	2	3	4	5	6	7

**One Way**  **Round Trip**

**Fill Data For Airline Search**

How many of passengers? \*

Total number of travellers?

How many are aged 2-11?

Where would you like to go? \*

Leaving From

Going To

When would you like to travel?

Departing

Time

**Passenger and Pricing Information**

Ticket Type

Seat Position

Smoking  Show me non-stop flight first

What airlines do you like prefer?

Airline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก รูปที่ 17-3 การค้นหาสายการบิน ของเอกสารทุกครั้งที่มีการนำไปใช้

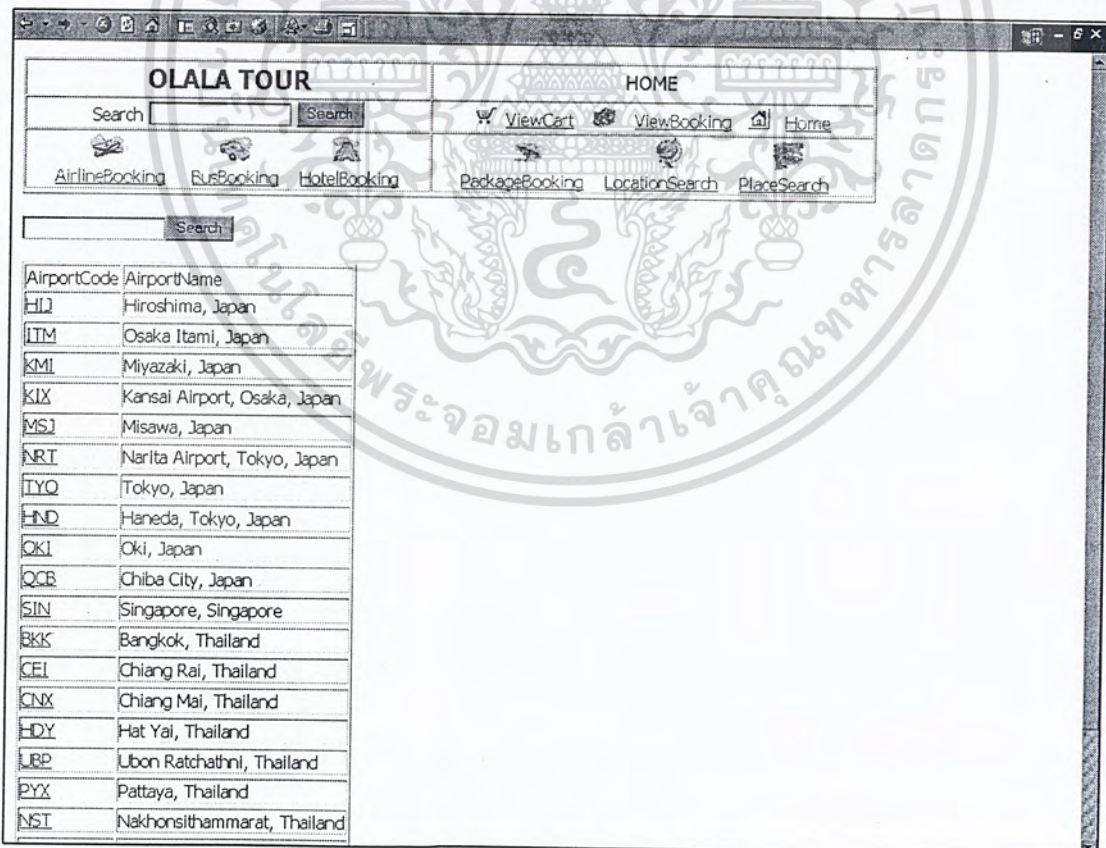
ในกรณีที่ลูกค้าต้องการซื้อหรือจองสินค้าจำเป็นต้องสมัครเป็นสมาชิกก่อน โดยการทำงานคือให้ลูกค้ากรอกรายละเอียดต่างๆเพื่อเข้าเป็นสมาชิก และลูกค้าสามารถที่จะใช้ชื่ออีเมลและรหัสผ่านในการเข้าใช้งานในครั้งต่อไป

แต่ในกรณีที่ลูกค้าต้องการค้นหาสินค้าและทำการเลือกสินค้าใส่รถเข็นลูกค้ายังไม่จำเป็นต้องสมัครเป็นสมาชิกก่อน เมื่อจะทำการจองสินค้าจึงทำการเข้าระบบหรือสมัครสมาชิกใหม่

### 17.1.3 การค้นหาสินค้าและการเลือกสินค้าใส่รถเข็น

1. การค้นหาตั๋วเครื่องบินมีข้อมูลที่ใช้ในการค้นหาคือ

- ชนิดตั๋วไปอย่างเดียวหรือตั๋วไป-กลับ โดยกดที่ปุ่ม Round Trip
- จำนวนผู้โดยสาร โดยจะให้เลือกจำนวนผู้โดยสารทั้งหมดและจำนวนเด็กอายุระหว่าง 2-11 ปี เพื่อคำนวณราคาที่แตกต่างกัน
- สถานที่ต้นทางและปลายทางที่จะไป สามารถที่จะเลือกสนามบินได้โดยกด Choose Airport
- วันที่จะเดินทาง
- ชนิดของตั๋วและตำแหน่งที่นั่ง
- บริษัทสายการบิน



รูปที่ 17-4 การค้นหาสนามบิน

เมื่อกดปุ่ม show me airline จะแสดงตัวการเดินทางที่เป็นไปได้ โดยที่หน้าจอนี้สามารถที่จะเปลี่ยนแปลงการค้นหาได้และสามารถเรียงลำดับการค้นหาตามค่าเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชื่อสายการบิน
- ราคาตั๋ว
- เวลาในการเดินทาง
- เวลาที่ออกเดินทาง
- เวลาที่ถึงปลายทาง

The screenshot shows the OLALA TOUR website interface. At the top, there are navigation links for 'HOME', 'ViewCart', and 'ViewBooking'. Below this is a search bar and a menu with options like 'AirlineBooking', 'BusBooking', 'HotelBooking', 'PackageBooking', 'LocationSearch', and 'PlaceSearch'. The main content area is divided into two sections: 'Modify Search' and 'SortBy'.

**Modify Search:**

- Departure Airport: BKK
- Destination Airport: LAX
- Depart Time: MAY 7 2001, Morning
- Airline: Search All
- Ticket Type: Select All
- Seat Position: Select All
- Smoking:
- Show me non-stop flight:
- Passengers: Total: 1, Children: 0

**SortBy:**

- Lowest Price (selected)
- Shortest Path
- Depart Time
- Arrive Time


**Search Results:**

Flight Details	Total Price	Action
Air Canada Flight SN625 Departs 7 พฤษภาคม 2001 From Bangkok, Thailand at 15:00:00 To Los Angeles, USA at 13:00:00 Ticket Type First Class Total Time 22 hr 0 mn	52400	add to cart
Air Canada Flight SN625 Departs 7 พฤษภาคม 2001 From Bangkok, Thailand at 15:00:00 To Los Angeles, USA at 13:00:00 Ticket Type Business Class Total Time 22 hr 0 mn	47300	add to cart
Air Canada Flight SN625 Departs 7 พฤษภาคม 2001 From Bangkok, Thailand at 15:00:00 To Los Angeles, USA at 13:00:00 Ticket Type Economy Class Total Time 22 hr 0 mn	42200	add to cart

รูปที่ 17-5 การแสดงผลการค้นหาสายการบิน

สามารถดูรายละเอียดของตั๋วได้จากกรกดปุ่ม Detail และหีบรยกรนั้ นใส่รถเข็น โดยกดปุ่ม add to cart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OLALA TOUR		AIRLINE DETAIL			
Search	<input type="text"/>	<input type="button" value="Search"/>	<input type="button" value="ViewCart"/>	<input type="button" value="ViewBooking"/>	<input type="button" value="Home"/>
<input type="button" value="AirlineBooking"/>	<input type="button" value="BusBooking"/>	<input type="button" value="HotelBooking"/>	<input type="button" value="PackageBooking"/>	<input type="button" value="LocationSearch"/>	<input type="button" value="PlaceSearch"/>
 Air Canada		<input type="button" value="กลับไปหน้ารายการ"/>			
Flight	SN625				
TicketType	First Class				
From	Bangkok, Thailand				
To	Los Angeles, USA				
Departs	วันจันทร์ที่ 7 พฤษภาคม 2001				
DepartTime	15:00:00				
ArriveTime	13:00:00				
Seat Position	window				
Distance	20000				
Passenger	1				
TotalPrice	52400				

รูปที่ 17-6 การดูรายละเอียดของตั๋วการเดินทาง

การค้นหาตั๋วเครื่องบินชนิดไป-กลับ โดยการกดปุ่ม Round Trip ที่หน้าการค้นหาตั๋วเครื่องบินจะมีตัวเลือกเวลาการเดินทางกลับให้ใส่และเมื่อทำการค้นหาจะแสดงรายการตั๋วเครื่องบินเที่ยวไปที่เป็นไปได้ตามข้อมูลในการค้นหาก่อนและเมื่อเลือกรายการนั้นแล้วจึงทำการเลือกตั๋วเครื่องบินเที่ยวกลับ

OLALA TOUR		HOME			
Search	<input type="text"/>	<input type="button" value="Search"/>	<input type="button" value="ViewCart"/>	<input type="button" value="ViewBooking"/>	<input type="button" value="Home"/>
<input type="button" value="AirlineBooking"/>	<input type="button" value="BusBooking"/>	<input type="button" value="HotelBooking"/>	<input type="button" value="PackageBooking"/>	<input type="button" value="LocationSearch"/>	<input type="button" value="PlaceSearch"/>
<b>Modify Search</b> Departure Airport <input type="text" value="BKK"/> Destination Airport <input type="text" value="LAX"/> DepartTime MAY 7 2001 <input type="button" value="Morning"/> ReturnTime MAY 8 2001 <input type="button" value="Morning"/> Airline <input type="text" value="Search All"/> Ticket Type <input type="button" value="Select All"/> Seat Position <input type="button" value="Select All"/> <input type="checkbox"/> Smoking <input type="checkbox"/> Show me non-stop flight Passengers Total : <input type="text" value="1"/> Children : <input type="text" value="0"/> <input type="button" value="Modify Search"/>		<b>SortBy</b> <input checked="" type="radio"/> Airline Name <input type="radio"/> Lowest Price <input type="radio"/> Shortest Path <input type="radio"/> Depart Time <input type="radio"/> Arrive Time Depart 1 adult 52400 and 0 child 47160 Total Price 52400 <input type="button" value="Select and Continue"/> Flight SN625 Departs วันจันทร์ที่ 7 พฤษภาคม 2001 From Bangkok, Thailand at 15:00:00 To Los Angeles, USA at 13:00:00 Air Canada Ticket Type First Class Total Time 22 hr 0 mn Detail			
		1 adult 47300 and 0 child 42570 Total Price 47300 <input type="button" value="Select and Continue"/> Flight SN625 Departs วันจันทร์ที่ 7 พฤษภาคม 2001 From Bangkok, Thailand at 15:00:00 To Los Angeles, USA at 13:00:00 Air Canada Ticket Type Business Class Total Time 22 hr 0 mn Detail			
		1 adult 42200 and 0 child 37980 Total Price 42200 <input type="button" value="Select and Continue"/> Flight SN625 Departs วันจันทร์ที่ 7 พฤษภาคม 2001 From Bangkok, Thailand at 15:00:00 To Los Angeles, USA at 13:00:00 Air Canada Ticket Type Economy Class Total Time 22 hr 0 mn Detail			

รูปที่ 17-7 การเลือกสายการบินขาไปของตั๋วการเดินทางแบบไป-กลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OLALA TOUR		HOME	
Search <input type="text"/>	<input type="button" value="Search"/>	ViewCart	ViewBooking  Home
AirlineBooking	BusBooking	HotelBooking	PackageBooking
		LocationSearch	PlaceSearch
<b>Depart</b>			
1 adult 47300 and 0 child 42570	Total Price 47300	<a href="#">New Flight Search</a>	
Air Canada <a href="#">Detail</a>	Flight Departs From To Ticket Type Total Time	SN625 วันจันทร์ที่ 7 พฤษภาคม 2001 Bangkok, Thailand at 15:00:00 Los Angeles, USA at 13:00:00 Business Class 22 hr 0 mn	
<b>Return</b>			
1 adult 52400 and 0 child 47160	Total Price 52400	<a href="#">add to cart</a>	
Air Canada <a href="#">Detail</a>	Flight Return From To Ticket Type Total Time	SN676 วันอังคารที่ 8 พฤษภาคม 2001 Los Angeles, USA at 9:00:00 Bangkok, Thailand at 6:00:00 First Class 21 hr 0 mn	
1 adult 47300 and 0 child 42570	Total Price 47300	<a href="#">add to cart</a>	
Air Canada <a href="#">Detail</a>	Flight Return From To Ticket Type Total Time	SN676 วันอังคารที่ 8 พฤษภาคม 2001 Los Angeles, USA at 9:00:00 Bangkok, Thailand at 6:00:00 Business Class 21 hr 0 mn	

### รูปที่ 17-8 การเลือกสายการบินขากลับของตัวการเดินทางแบบไป-กลับ

2. การค้นหาตัวการเดินทางมีข้อมูลที่ใดในการค้นหาคือ
  - ชนิดตัวไปอย่างเดียวหรือตัวไป-กลับ โดยกดที่ปุ่ม Round Trip
  - จำนวนผู้โดยสาร โดยจะให้เลือกจำนวนผู้โดยสารทั้งหมด
  - สถานที่ต้นทางและปลายทางที่จะไป
  - วันที่จะเดินทาง
  - ชนิดของตัวและตำแหน่งที่นั่ง
  - บริษัทสายการบินรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**OLALA TOUR** HOME

Search

[ViewCart](#) [ViewBooking](#) [Home](#)

[AirlineBooking](#) [BusBooking](#) [HotelBooking](#) [PackageBooking](#) [LocationSearch](#) [PlaceSearch](#)

**Weather**  
 London,UK 30-34.5F  
 Bangkok,TH 21-25F  
 Tokyo,Ja 5-10F

**Calendar**  
 May 2001   

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

One Way |

**Fill Data For Bus Search**

How many of passengers? \*  
 Total number of travellers?

Where would you like to go?(Internal Country) \*  
 Country   
 From   
 To

When would you like to travel?  
 Departing     
 Time

**Passenger and Pricing Information**  
 Ticket Type   
 Seat Position

What bus company do you like prefer?  
 Bus Company

### รูปที่ 17-9 การค้นหาเที่ยวการเดินรถ

เมื่อกดปุ่ม show me bus จะแสดงตัวการเดินรถที่เป็นไปได้ โดยที่หน้าจอจะสามารถที่จะเปลี่ยนแปลงการค้นหาได้และสามารถเรียงลำดับการค้นหาตาม

- ชื่อสายการเดินรถ
- ราคาตั๋ว
- เวลาในการเดินทาง
- เวลาที่ออกเดินทาง
- เวลาที่ถึงปลายทาง

สามารถดูรายละเอียดของตัวได้จากกรกดปุ่ม Detail และหขยบรายการนั้นใส่รถเข็น โดยกด

ปุ่ม add to cart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>OLALA TOUR</b>	<b>HOME</b>
Search <input type="text"/> <input type="button" value="Search"/>	<input type="button" value="ViewCart"/> <input type="button" value="ViewBooking"/> <input type="button" value="Home"/>
<input type="button" value="AirlineBooking"/> <input type="button" value="BusBooking"/> <input type="button" value="HotelBooking"/>	<input type="button" value="PackageBooking"/> <input type="button" value="LocationSearch"/> <input type="button" value="PlaceSearch"/>

<b>Modify Search</b> Thailand Departure Bus มหานคร2 Destination Bus ขอนแก่น DepartTime MAY 17 2001 Morning BusCompany <input type="button" value="Search All"/> Ticket Type <input type="button" value="Select All"/> SeatPosition <input type="button" value="Select All"/> Passengers Total : 1 <input type="button" value="Modify Search"/>	<b>SortBy</b> <input type="radio"/> Bus Name <input type="radio"/> Lowest Price <input type="radio"/> Shortest Path <input type="radio"/> Depart Time <input type="radio"/> Arrive Time  <table style="width: 100%;"> <tr> <td>Price/Seat 477</td> <td>Total Price 477</td> <td style="text-align: right;"><input type="button" value="add to cart"/></td> </tr> <tr> <td></td> <td>Flight B1</td> <td></td> </tr> <tr> <td></td> <td>Departs 17 พฤษภาคม 2001</td> <td></td> </tr> <tr> <td></td> <td>From มหานคร2 at 18:00:00</td> <td></td> </tr> <tr> <td></td> <td>To ขอนแก่น at 23:30:00</td> <td></td> </tr> <tr> <td>407 พัดนก</td> <td>Ticket Type ป.บ.1 VIP</td> <td></td> </tr> <tr> <td><input type="button" value="Detail"/></td> <td>Total Time 5 hr 30 mn</td> <td></td> </tr> </table> <table style="width: 100%;"> <tr> <td>Price/Seat 397.5</td> <td>Total Price 397.5</td> <td style="text-align: right;"><input type="button" value="add to cart"/></td> </tr> <tr> <td></td> <td>Flight B3</td> <td></td> </tr> <tr> <td></td> <td>Departs 17 พฤษภาคม 2001</td> <td></td> </tr> <tr> <td></td> <td>From มหานคร2 at 20:00:00</td> <td></td> </tr> <tr> <td></td> <td>To ขอนแก่น at 1:30:00</td> <td></td> </tr> <tr> <td>407 พัดนก</td> <td>Ticket Type ป.บ.1</td> <td></td> </tr> <tr> <td><input type="button" value="Detail"/></td> <td>Total Time 5 hr 30 mn</td> <td></td> </tr> </table> <table style="width: 100%;"> <tr> <td>Price/Seat 318</td> <td>Total Price 318</td> <td style="text-align: right;"><input type="button" value="add to cart"/></td> </tr> <tr> <td></td> <td>Flight B7</td> <td></td> </tr> <tr> <td></td> <td>Departs 17 พฤษภาคม 2001</td> <td></td> </tr> <tr> <td></td> <td>From มหานคร2 at 22:00:00</td> <td></td> </tr> <tr> <td></td> <td>To ขอนแก่น at 3:30:00</td> <td></td> </tr> <tr> <td>407 พัดนก</td> <td>Ticket Type ป.บ.2</td> <td></td> </tr> <tr> <td><input type="button" value="Detail"/></td> <td>Total Time 5 hr 30 mn</td> <td></td> </tr> </table>	Price/Seat 477	Total Price 477	<input type="button" value="add to cart"/>		Flight B1			Departs 17 พฤษภาคม 2001			From มหานคร2 at 18:00:00			To ขอนแก่น at 23:30:00		407 พัดนก	Ticket Type ป.บ.1 VIP		<input type="button" value="Detail"/>	Total Time 5 hr 30 mn		Price/Seat 397.5	Total Price 397.5	<input type="button" value="add to cart"/>		Flight B3			Departs 17 พฤษภาคม 2001			From มหานคร2 at 20:00:00			To ขอนแก่น at 1:30:00		407 พัดนก	Ticket Type ป.บ.1		<input type="button" value="Detail"/>	Total Time 5 hr 30 mn		Price/Seat 318	Total Price 318	<input type="button" value="add to cart"/>		Flight B7			Departs 17 พฤษภาคม 2001			From มหานคร2 at 22:00:00			To ขอนแก่น at 3:30:00		407 พัดนก	Ticket Type ป.บ.2		<input type="button" value="Detail"/>	Total Time 5 hr 30 mn	
Price/Seat 477	Total Price 477	<input type="button" value="add to cart"/>																																																														
	Flight B1																																																															
	Departs 17 พฤษภาคม 2001																																																															
	From มหานคร2 at 18:00:00																																																															
	To ขอนแก่น at 23:30:00																																																															
407 พัดนก	Ticket Type ป.บ.1 VIP																																																															
<input type="button" value="Detail"/>	Total Time 5 hr 30 mn																																																															
Price/Seat 397.5	Total Price 397.5	<input type="button" value="add to cart"/>																																																														
	Flight B3																																																															
	Departs 17 พฤษภาคม 2001																																																															
	From มหานคร2 at 20:00:00																																																															
	To ขอนแก่น at 1:30:00																																																															
407 พัดนก	Ticket Type ป.บ.1																																																															
<input type="button" value="Detail"/>	Total Time 5 hr 30 mn																																																															
Price/Seat 318	Total Price 318	<input type="button" value="add to cart"/>																																																														
	Flight B7																																																															
	Departs 17 พฤษภาคม 2001																																																															
	From มหานคร2 at 22:00:00																																																															
	To ขอนแก่น at 3:30:00																																																															
407 พัดนก	Ticket Type ป.บ.2																																																															
<input type="button" value="Detail"/>	Total Time 5 hr 30 mn																																																															

### รูปที่ 17-10 การแสดงผลการค้นหาเที่ยวการเดินทาง

การค้นหาเที่ยวการเดินทางชนิดไป-กลับ โดยการกดปุ่ม Round Trip ที่หน้าการค้นหาเที่ยวการเดินทาง จะมีตัวเลือกเวลาการเดินทางกลับให้ได้และเมื่อทำการค้นหาจะแสดงรายการเที่ยวการเดินทางที่เข้าไปที่เป็นไปได้ตามข้อมูลในการค้นหาและเมื่อเลือกรายการนั้นแล้วจึงทำการเลือกเที่ยวการเดินทางที่ขากลับ

3. การค้นหาโรงแรมมีข้อมูลที่ใส่ในการค้นหาคือ

- ชนิดของโรงแรม
- ระดับของโรงแรม
- เลือกโรงแรมได้จากใกล้สนามบินหรือชื่อที่อยู่ของโรงแรมหรือใกล้สถานที่สำคัญอะไร
- วันที่จะเข้าพักและออก
- ตัวเลือกอื่นๆของโรงแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Search

AirlineBooking BusBooking HotelBooking PackageBooking LocationSearch PlaceSearch

### Weather

London, UK 30-34.5F

Bangkok, TH 21-25F

Tokyo, Ja 5-10F

### Calendar

Check in

May 2001  2001

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Check out

May 2001  2001

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

### Fill Data For Hotel Search

Property Hotel

Star

Where do you need a hotel? (Choose Only 1 Choice)

Search for hotel by an airport

Search for hotel by hotel name   
by address

Search for hotel near attraction place

When would you like to stay?

Check in

Check out

Select up to special amenities

Business Center  Fitness Center  Pool

Casino  Golf  Restaurant

Children's Program  Meeting Facils  Tennis

Dry Cleaning  Pets Allowed  Wheelchair Access

### รูปที่ 17-11 การค้นหาโรงแรม

เมื่อกดปุ่ม show me hotel จะแสดงโรงแรมที่เป็นไปได้ โดยที่หน้าจอนี้สามารถที่จะเปลี่ยนแปลงการค้นหาได้ และหีบรายนการนั้นใส่รถเข็นโดยกดปุ่ม add to cart

OLALA TOUR HOME

Search

AirlineBooking BusBooking HotelBooking PackageBooking LocationSearch PlaceSearch

### Modify Search

Property Hotel

Star

Search for hotels

by an airport

by hotel name   
by address

near attraction place

When would you like to stay?

Checkin

★★★★★ **royal garden**

Location on  
454 phuket, thailand

Description  
near sea has golf club  
Check in วันพฤหัสบดีที่ 17 พฤษภาคม 2001  
Check out วันศุกร์ที่ 18 พฤษภาคม 2001

Single Bed 0 Beds  
Couple Bed 2 Beds  
Maximum Guest 4  
★★★★★

500 BATH  Rooms

Single Bed 2 Beds  
Couple Bed 0 Beds  
Maximum Guest 5  
★★★★★

600 BATH  Rooms

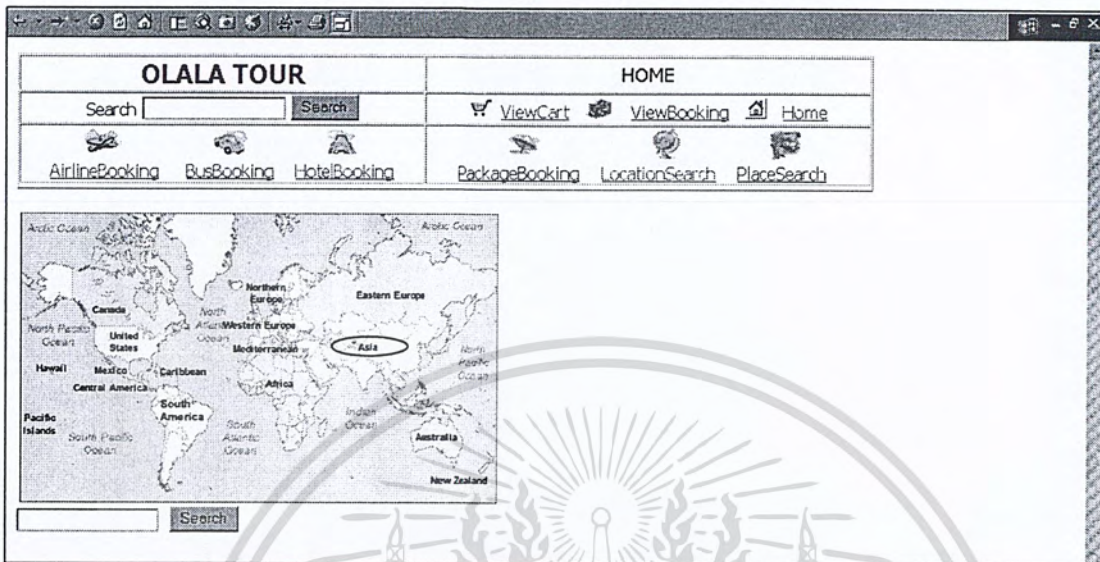
### รูปที่ 17-12 การแสดงผลการค้นหาโรงแรมพร้อมใส่จำนวนห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

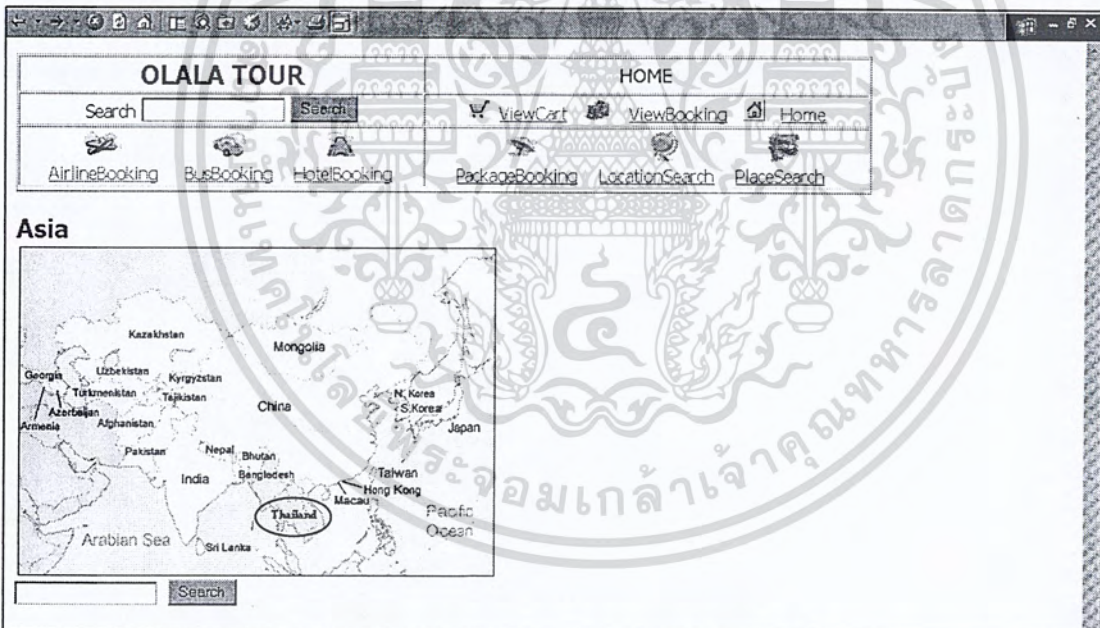
#### 4. การค้นหาสถานที่

- Location Search

สามารถเลือกที่ชื่อประเทศหรือจะใส่ข้อมูลในการค้นหาแล้วกดปุ่ม search



รูปที่ 17-13 การค้นหาสถานที่จากแผนที่ 1



รูปที่ 17-14 การค้นหาสถานที่จากแผนที่ 2

เมื่อกดเลือกที่ Asia และ Thailand จะแสดงข้อมูลในการค้นหาเหมือนกับการค้นหาจากPlace Search ที่ใส่ชื่อประเทศเป็น Thailand

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OLALA TOUR		HOME	
Search <input type="text"/> <input type="button" value="Search"/>		<a href="#">ViewCart</a>	<a href="#">Home</a>
<a href="#">AirlineBooking</a>	<a href="#">BusBooking</a>	<a href="#">HotelBooking</a>	
		<a href="#">PackageBooking</a>	<a href="#">LocationSearch</a>
		<a href="#">PlaceSearch</a>	
<b>Modify Search</b>			
Where would you like to go?			
Country	<input type="text" value="Thailand"/>	Country	Thailand
State	<input type="text"/>	State	Bangkok
Province	<input type="text"/>	Province	Bangkok
Place Name	<input type="text"/>	Description	My Institute
Description	<input type="text"/>		
Activity	<input type="text"/>	Country	Thailand
Category	<input type="text"/>	State	Bangkok
Festival	<input type="text"/>	Province	Bangkok
<input type="button" value="Modify Search"/>		Description	Fresh Market Near KMITL
		Country	Thailand
		State	Bangkok
		Province	Bangkok
		Description	Food Center Near Kmitl

### รูปที่ 17-15 การค้นหาสถานที่จากตัวเลือก

- Place Search

การค้นหาสถานที่ที่มีข้อมูลที่ใส่ในการค้นหาคือ

- ประเทศ รัฐ และจังหวัด
- ชื่อสถานที่
- กิจกรรมที่สถานที่นั้นมี
- ประเภทของสถานที่นั้น
- งานเทศกาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 17.1.4 การดูสินค้าในรถเข็น

กดที่ปุ่ม View Cart จะเป็นการแสดงรายการที่เราทำการเลือกสินค้าใส่รถเข็นเอาไว้ โดยจะแสดงรายละเอียดของรายการนั้น จำนวน ราคาต่อหน่วย ราคารวมของรายการนั้น

OLALA TOUR		CART		
Search <input type="text"/> Search		<a href="#">ViewCart</a>	<a href="#">ViewBooking</a>	<a href="#">Home</a>
<a href="#">AirlineBooking</a>	<a href="#">BusBooking</a>	<a href="#">HotelBooking</a>	<a href="#">PackageBooking</a>	<a href="#">LocationSearch</a> <a href="#">PlaceSearch</a>
<b>ตระกร้าของคุณมีรายการดังนี้</b>				
สินค้า	รายการ	จำนวน	ราคา/หน่วย	ราคารวม
<input type="checkbox"/>	Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 To :Los Angeles, USA at 13:00:00	1	52400/47160	52400
<input type="checkbox"/>	Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 To :Los Angeles, USA at 13:00:00	1	94600/85140	94600
<input type="checkbox"/>	Flight : SN676 Departs : วันอังคารที่ 8 พฤษภาคม 2001 From :Los Angeles, USA at 9:00:00 To :Bangkok, Thailand at 6:00:00			
<input type="checkbox"/>	Flight : B7 Departs : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 From :หมอชิต2 at 22:00:00 To :ขอนแก่น at 3:30:00	1	318	318
<input type="checkbox"/>	Check in : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 Check out : วันศุกร์ที่ 18 พฤษภาคม 2001 0 SB 2 CB 4 Maximum 4 Class 2 SB 0 CB 5 Maximum 5 Class	1 2	500 600	500 1200
โปรดคำนวณราคาใหม่ก่อนทำการจองทุกครั้ง		รวม	0	
<a href="#">Clear Cart</a>				

รูปที่ 17-16 การแสดงข้อมูลในรถเข็น

เมื่อจะทำการจองต้องยืนยันและคำนวณราคาก่อนจะได้ราคารวมทั้งหมดแล้วจึงทำการจองต่อไป ซึ่งถ้าไม่ต้องการรายการที่อยู่ในรถเข็นก็ไม่ต้องทำการยืนยัน หรือสามารถที่จะลบรายการทั้งหมดที่อยู่ในรถเข็นโดยการกดปุ่ม Clear Cart

OLALA TOUR			CART		
Search <input type="text"/>		Search	ViewCart	ViewBooking	Home
AirlineBooking	BusBooking	HotelBooking	PackageBooking	LocationSearch	PlaceSearch

### ตระกร้าของคุณมีรายการดังนี้

ยืนยัน	รายการ	จำนวน	ราคา/หน่วย	ราคารวม
<input checked="" type="checkbox"/>	Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 To :Los Angeles, USA at 13:00:00	1	52400/47160	52400
<input checked="" type="checkbox"/>	Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 To :Los Angeles, USA at 13:00:00	1	94600/85140	94600
<input checked="" type="checkbox"/>	Flight : SN676 Departs : วันอังคารที่ 8 พฤษภาคม 2001 From :Los Angeles, USA at 9:00:00 To :Bangkok, Thailand at 6:00:00			
<input checked="" type="checkbox"/>	Flight : B7 Departs : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 From :หมอชิต2 at 22:00:00 To :ขอนแก่น at 3:30:00	1	318	318
<input checked="" type="checkbox"/>	Check in : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 Check out : วันศุกร์ที่ 18 พฤษภาคม 2001 0 SB 2 CB 4 Maximum 4 Class 2 SB 0 CB 5 Maximum 5 Class	1 2	500 600	500 1200
โปรดคำนวณราคาใหม่ก่อนทำการจองทุกครั้ง			<b>รวม</b>	<b>243618</b>

รูปที่ 17- 17 การคำนวณราคา

## 17.1.5 การซื้อหรือจองสินค้าจากรถเข็น

เมื่อต้องการซื้อหรือจองสินค้าก็กดปุ่ม **จอง** ที่หน้า View Cart

OLALA TOUR			CART		
Search <input type="text"/>		Search	ViewCart	ViewBooking	Home
AirlineBooking	BusBooking	HotelBooking	PackageBooking	LocationSearch	PlaceSearch

### กรุณาใส่ข้อมูลการเดินทาง

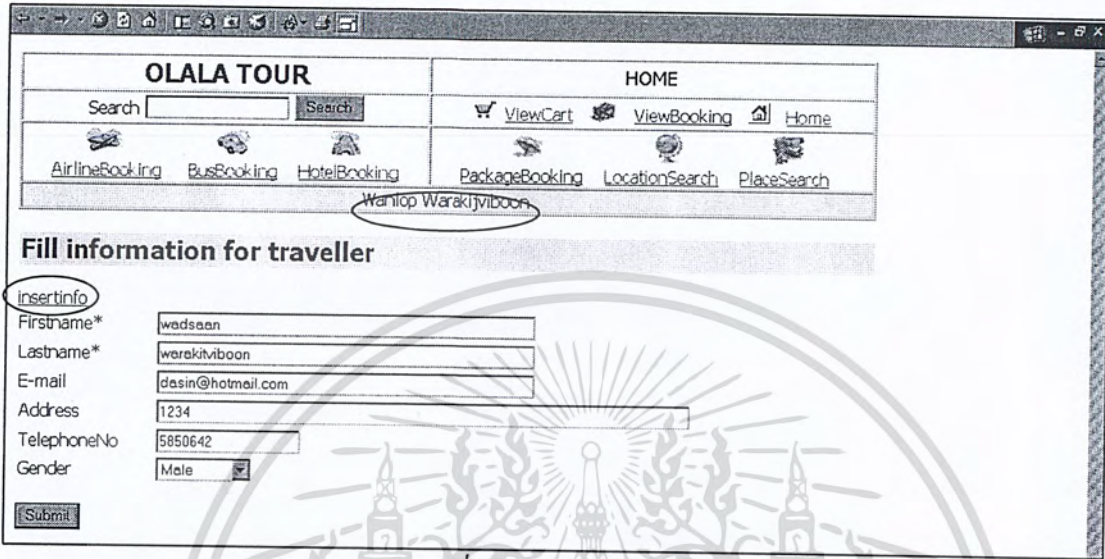
รายการ	จำนวน	ราคา/หน่วย	ราคารวม
Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 To :Los Angeles, USA at 13:00:00	1	52400/47160	52400 <a href="#">insertinfo</a>
Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 To :Los Angeles, USA at 13:00:00	1	94600/85140	94600 <a href="#">insertinfo</a>
Flight : SN676 Departs : วันอังคารที่ 8 พฤษภาคม 2001 From :Los Angeles, USA at 9:00:00 To :Bangkok, Thailand at 6:00:00			
Flight : B7 Departs : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 From :หมอชิต2 at 22:00:00 To :ขอนแก่น at 3:30:00	1	318	318 <a href="#">insertinfo</a>
Check in : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 Check out : วันศุกร์ที่ 18 พฤษภาคม 2001 0 SB 2 CB 4 Maximum 4 Class 2 SB 0 CB 5 Maximum 5 Class	1 2	500 600	500 1200
<b>รวม</b>			<b>290918</b>

รูปที่ 17- 18 การใส่ข้อมูลผู้เดินทาง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

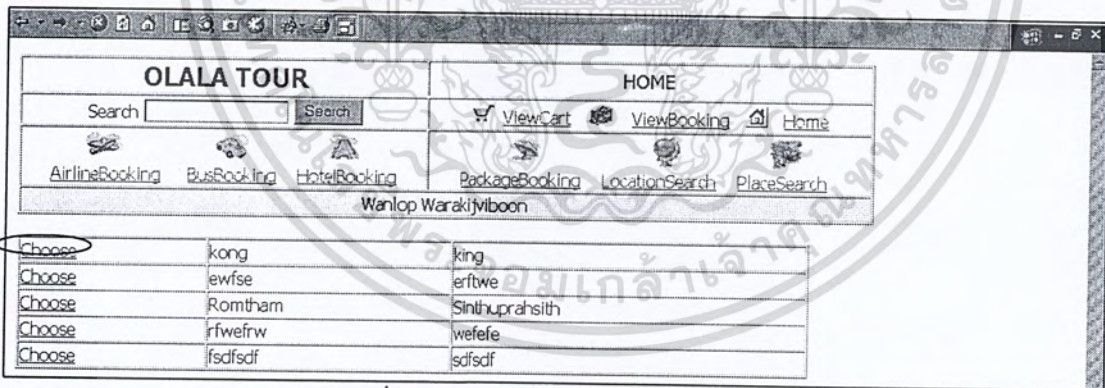
ทำการใส่ข้อมูลของผู้เดินทางได้โดยการกดปุ่ม insertinfo

ถ้ายังไม่ทำการเข้าสู่ระบบจะมีหน้าจอให้ทำการเข้าสู่ระบบหรือทำการสมัครสมาชิกใหม่และจะไปยังหน้าจอในกรอกข้อมูลสำหรับผู้เดินทางโดยที่การจองโรงแรมไม่จำเป็นต้องใส่ข้อมูลในส่วนนี้เพราะสามารถอ้างอิงได้จากข้อมูลของสมาชิก



รูปที่ 17- 19 การใส่ข้อมูลผู้เดินทาง 2

สามารถที่จะเลือกผู้เดินทางที่อยู่ในข้อมูลของสมาชิกได้โดยกดที่ปุ่ม insertinfo จะมีรายชื่อของคนที่เคยเดินทางโดยใช้ชื่อผู้จองเป็นเรามาให้เลือกและทำการ Submit เป็นอันใส่ข้อมูลสำหรับผู้เดินทางเรียบร้อยแล้วจะต้องใส่ข้อมูลสำหรับผู้เดินทางให้ครบทุกรายการ



รูปที่ 17- 20 การเลือกผู้เดินทางจากข้อมูลของสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OLALA TOUR		CART		
Search <input type="text"/> Search		ViewCart	ViewBooking	Home
AirlineBooking	BusBooking	HotelBooking	PackageBooking	LocationSearch
Wanlop Warakijviboon				
<b>กรุณาใส่ข้อมูลการเดินทาง</b>				
รายการ	จำนวน	ราคา/หน่วย	ราคารวม	
Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 Air Canada To :Los Angeles, USA at 13:00:00	1	52400/47160	52400	<a href="#">editinfo</a>
Flight : SN625 Departs : วันจันทร์ที่ 7 พฤษภาคม 2001 From :Bangkok, Thailand at 15:00:00 Air Canada To :Los Angeles, USA at 13:00:00	1	94600/85140	94600	<a href="#">editinfo</a>
Flight : SN676 Departs : วันอังคารที่ 8 พฤษภาคม 2001 From :Los Angeles, USA at 9:00:00 Air Canada To :Bangkok, Thailand at 6:00:00				
Flight : B7 Departs : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 From :หมอชิต2 at 22:00:00 407 พัดนา To :ขอนแก่น at 3:30:00	1	318	318	<a href="#">editinfo</a>
Check in : วันพฤหัสบดีที่ 17 พฤษภาคม 2001 Check out : วันศุกร์ที่ 18 พฤษภาคม 2001 royal garden 0 SB 2 CB 4 Maximum 4 Class 2 SB 0 CB 5 Maximum 5 Class	1 2	500 600	500 1200	
<a href="#">ViewCart</a>		<b>รวม</b>	<b>290918</b>	

รูปที่ 17-21 การแก้ไขข้อมูลของผู้เดินทาง

เมื่อใส่ข้อมูลสำหรับผู้เดินทางแล้วสามารถทำการแก้ไขข้อมูลสำหรับผู้เดินทางโดยคลิกปุ่ม [editinfo](#) เมื่อทำครบทุกรายการแล้วก็กดปุ่ม [จอง](#) อีกครั้ง

Wanlop Warakijviboon	
TicketNo	A1
AirFlight	SN625
From	Bangkok, Thailand at 15:00:00
To	Los Angeles, USA at 13:00:00
Depart	วันจันทร์ที่ 7 พฤษภาคม 2001
TicketNo	C1
AirFlight	SN625
From	Bangkok, Thailand at 15:00:00
To	Los Angeles, USA at 13:00:00
Depart	วันจันทร์ที่ 7 พฤษภาคม 2001
TicketNo	C1
AirFlight	SN676
From	Los Angeles, USA at 15:00:00
To	Bangkok, Thailand at 13:00:00
Depart	วันอังคารที่ 8 พฤษภาคม 2001
TicketNo	A1
BusFlight	B7
From	หมอชิต2 at 22:00:00
To	ขอนแก่น at 3:30:00
Depart	วันพฤหัสบดีที่ 17 พฤษภาคม 2001
TicketNo	201
HotelName	royal garden
Property	0 SB 2 CB 4 Maximum 4 Class
Check in	วันพฤหัสบดีที่ 17 พฤษภาคม 2001
Check out	วันศุกร์ที่ 18 พฤษภาคม 2001
TicketNo	202
HotelName	royal garden
Property	2 SB 0 CB 5 Maximum 5 Class
Check in	วันพฤหัสบดีที่ 17 พฤษภาคม 2001
Check out	วันศุกร์ที่ 18 พฤษภาคม 2001

รูปที่ 17-22 แสดงผลการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการจองเรียบร้อยแล้วจะแสดงหมายเลขที่นั่งหรือหมายเลขห้องที่ได้

### 17.1.6 การชำระเงิน

ระบบจะเก็บรายการต่างๆที่ได้จองเอาไว้อยู่ในข้อมูลของสมาชิกเมื่อจะทำการชำระเงินต้องทำการเข้าสู่ระบบก่อนโดยคลิกปุ่ม View Booking จะแสดงรายการพร้อมทั้งรายละเอียดต่างๆที่ได้จองเอาไว้ในชื่อของสมาชิกนั้น สามารถทำการชำระเงินได้โดยการกด Confirm และทำการยกเลิกการจองนั้นได้โดยการกด Cancel

To	Los Angeles, USA at 13:00:00
Seat No	C1
Airline Company	Air Canada
Depart	8/5/2544
From	Los Angeles, USA at 9:00:00
To	Bangkok, Thailand at 6:00:00
Seat No	A1
Airline Company	Air Canada
Depart	7/5/2544
From	Bangkok, Thailand at 15:00:00
To	Los Angeles, USA at 13:00:00
Seat No	A1
Bus Company	407 พัฒนา
Depart	17/5/2544
From	หมอชิต2 at 22:00:00
To	ขอนแก่น at 3:30:00
Room No	201
Hotel Name	royal garden
Checkin	17/5/2544
Checkout	18/5/2544
Room No	202
Hotel Name	royal garden
Checkin	17/5/2544
Checkout	18/5/2544
Room No	101
Hotel Name	royal garden
Checkin	17/5/2544
Checkout	18/5/2544
BookingTime	7/5/2544 18:33:45
Deadline	7/5/2544
Total Price	149018
Confirm	Cancel

รูปที่ 17-23 การดูข้อมูลที่ได้จองไว้

OLALA TOUR		HOME	
Search <input type="text"/>	<input type="button" value="Search"/>	<input type="button" value="ViewCart"/>	<input type="button" value="ViewBooking"/>
<input type="button" value="Home"/>	<input type="button" value="Home"/>	<input type="button" value="Home"/>	<input type="button" value="Home"/>
<input type="button" value="AirlineBooking"/>	<input type="button" value="BusBooking"/>	<input type="button" value="HotelBooking"/>	<input type="button" value="PackageBooking"/>
<input type="button" value="LocationSearch"/>	<input type="button" value="PlaceSearch"/>		
Wanlop Warakijviboon			
กรุณาทำการชำระเงินที่ <input type="button" value="ชำระ"/> โดยใส่หมายเลขบัญชีปลายทางเป็น {84EF71AE-9482-410E-AA8D-13E48FB72822} และกรุณากรอกจำนวนเงินให้ถูกต้องด้วย จำนวนเงิน 149018 บาท ใส่ TransactionID ที่ได้จากธนาคารลงใน <input type="text"/>			
<input type="button" value="Submit"/>			

รูปที่ 17-24 การชำระเงิน

เมื่อกด **Confirm** แสดงหน้าจอให้ไปที่ธนาคารเพื่อทำการโอนเงินเมื่อโอนเงินเรียบร้อยแล้ว  
TransactionID จากหน้าธนาคารนำมาใส่ใน **textbox** และคลิกปุ่ม **submit**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**BANK**

กรุณาเลือกวิธีการชำระเงิน

หมายเลขบัตร Credit	บัญชีปลายทาง	จำนวนเงิน	Submit
<input type="text"/>	<input type="text"/>	<input type="text"/>	
หมายเลขบัตร Debit	บัญชีปลายทาง	จำนวนเงิน	Submit
<input type="text"/>	<input type="text"/>	<input type="text"/>	
บัญชีต้นทาง	บัญชีปลายทาง	จำนวนเงิน	Password
{0A9D26D4-9F4F-42E}	{84EF71AE-9482-410E}	149018	<input type="password"/>

รูปที่ 17-25 การโอนเงินผ่านทางธนาคาร

ที่หน้าจอของธนาคารจะมีให้เลือกวิธีการชำระเงิน โดยจะต้องใส่บัญชีปลายทางเป็น {84EF71AE-9482-410E-AA8D-13E48FB72822} ใส่จำนวนเงินให้ตรงกับที่ระบบแจ้งให้ทราบและกดปุ่ม submit ธนาคารจะให้หมายเลขมาหนึ่งชุดนำกลับไปใส่ที่หน้าจอของระบบ Olala Tour และกด Submit ก็จะทำให้การชำระเงินเรียบร้อย

**OLALA TOUR HOME**

Search  Search

ViewCart ViewBooking Home

AirlineBooking BusBooking HotelBooking PackageBooking LocationSearch PlaceSearch

Wanlop Warakijviboon

กรุณาทำการชำระเงินที่หน้า [ธนาคาร](#) โดยใส่หมายเลขบัญชีปลายทางเป็น {84EF71AE-9482-410E-AA8D-13E48FB72822} และกรอกจำนวนเงินให้ถูกต้องด้วย จำนวนเงิน 149018 บาท ใส่ TransactionID ที่ได้จากธนาคารลงใน

{AEFFA3CD-8114-46d}

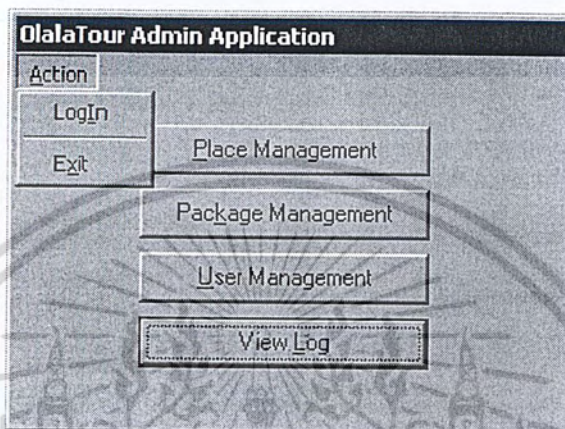
รูปที่ 17-26 การยืนยันการชำระเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 17.2 Admin application

เป็นแอปพลิเคชันสำหรับ ผู้ดูแลระบบในการจัดการเกี่ยวกับสถานที่ท่องเที่ยว, แพคเกจท่องเที่ยว, การจัดการกับผู้ใช้ของระบบ, การดูประวัติการทำงานจากระบบ

### 17.2.1 เข้าสู่การใช้งาน Admin application



รูปที่ 17-27 แสดงการเริ่มทำงาน

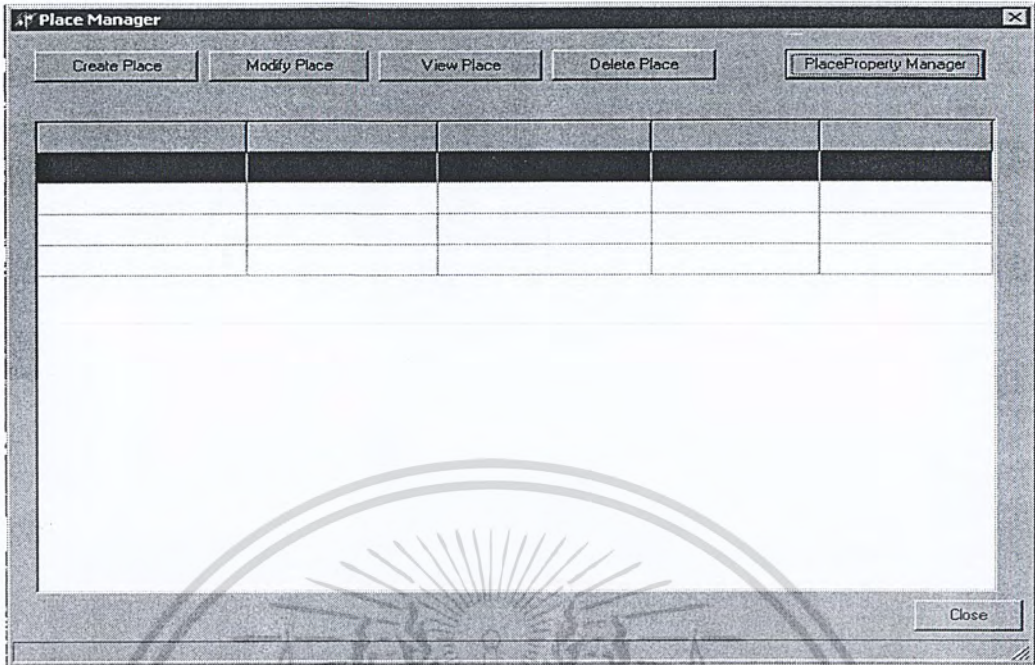
ทำการ Login เข้าสู่ระบบดังรูป

รูปที่ 17-28 แสดงการ Login

### 17.2.2 การจัดการเกี่ยวกับสถานที่ท่องเที่ยว

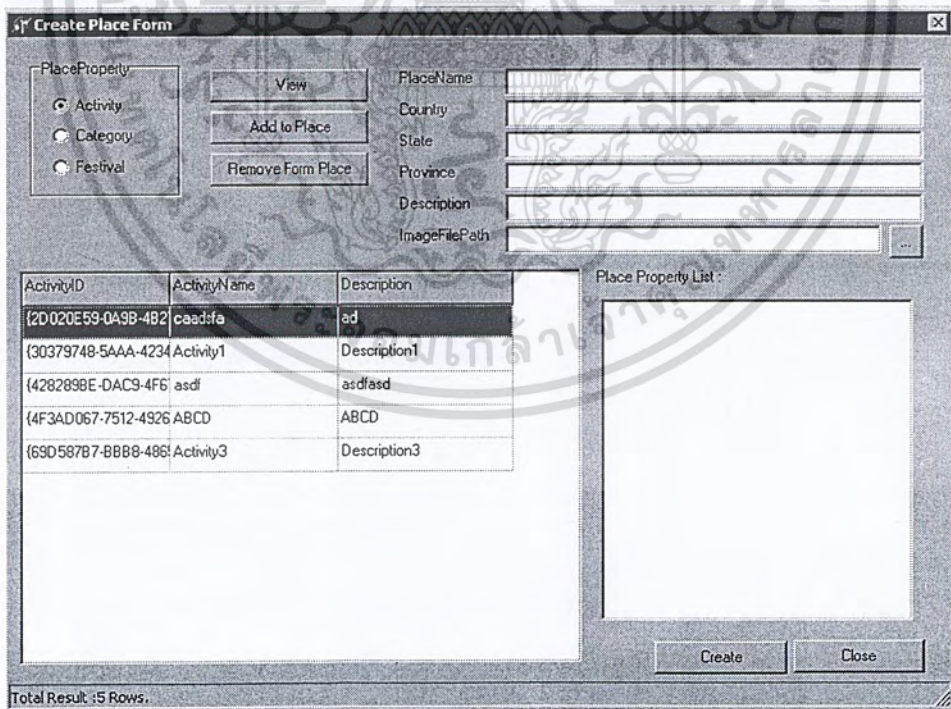
โดยการจัดการกับสถานที่ท่องเที่ยวสามารถทำการ สร้าง, แก้ไข, ดูข้อมูล, ลบ, จัดการคุณสมบัติของสถานที่ ซึ่งการทำงานจะอธิบาย ดังนี้

เอกสารนี้เป็นเริ่มเข้าสู่การจัดการสถานที่โดย Admin application ทำการเลือก Place Management ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 17-29 แสดงหน้าจอการจัดการสถานที่

หลังจากเข้าสู่การจัดการสถานที่แล้วสามารถทำการ สร้างสถานที่ให้ระบบรู้จักโดยกดปุ่ม Create place



รูปที่ 17-30 แสดงการสร้างสถานที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำการแก้ไขสถานที่ทำได้โดยการกดปุ่ม Modify place จะได้น้ำจอดังรูป

CategoryID	CategoryName	Description
{3529A8AB-ABDA-428	adfas	asdfs
{42B55100-40F5-45CF	edfadsfasd	asdfsdfasdfa
{F271DE86-3E5C-475E	adsf	asdfs

รูปที่ 17-31 แสดงการแก้ไขสถานที่

เมื่อแก้ไขเรียบร้อยแล้วสามารถดูข้อมูลของสถานที่ได้โดยกดปุ่ม View place

PlaceID	PlaceName	Country	State	Province
{28CABFCD-61DE-42AA-8CB	KMITL	Thailand		Bangkok
{6F35F53F-9780-4E5D-9613-5	Hua-Ta-Kae	Thailand		Bangkok
{7F619741-279F-4CA1-AD7D-1	SorJin-Da	Thailand		Bangkok
{83542CAD-2E68-4229-AE0D	GIT	USA	California	LA
{C15C5B9C-8306-4D70-8F9D-1	PlaceName	CountryName	StateName	ProvinceName

รูปที่ 17-32 แสดงการดูข้อมูลสถานที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถทำการลบสถานที่ที่ออกจากระบบ โดยคลิกปุ่ม Delete place  
และทำการจัดการกับคุณสมบัติของสถานที่ที่ได้โดยคลิกปุ่ม PlaceProperty Manager

**PlaceProperty Manager**

PlaceProperty

Activity  
 Category  
 Festival

PlacePropertyID: {542052AE-9541-4EF8-9759-28FB78D669D8}

Name: \_\_\_\_\_

Description: \_\_\_\_\_

Create Modify Delete View

FestivalID	FestivalName	Description
{39B38065-96AA-44AE-AECE-7E6F6E30967}	1235	Description2
{5AF9544D-91F9-45A0-9D68-C528251A8500}	ABCD	ABCD

Close

รูปที่ 17-33 แสดงการจัดการกับคุณสมบัติของสถานที่

### 17.2.3 การจัดการแพคเกจการท่องเที่ยว

สามารถทำการสร้าง, ดูข้อมูลการจอง, ตกลงหรือปฏิเสธการจองได้ซึ่งการทำงานจะอธิบาย ดังนี้  
ที่หน้าจอแรกของ Admin application ทำการเลือก Package management

**Package Manager Form**

View package

Create package

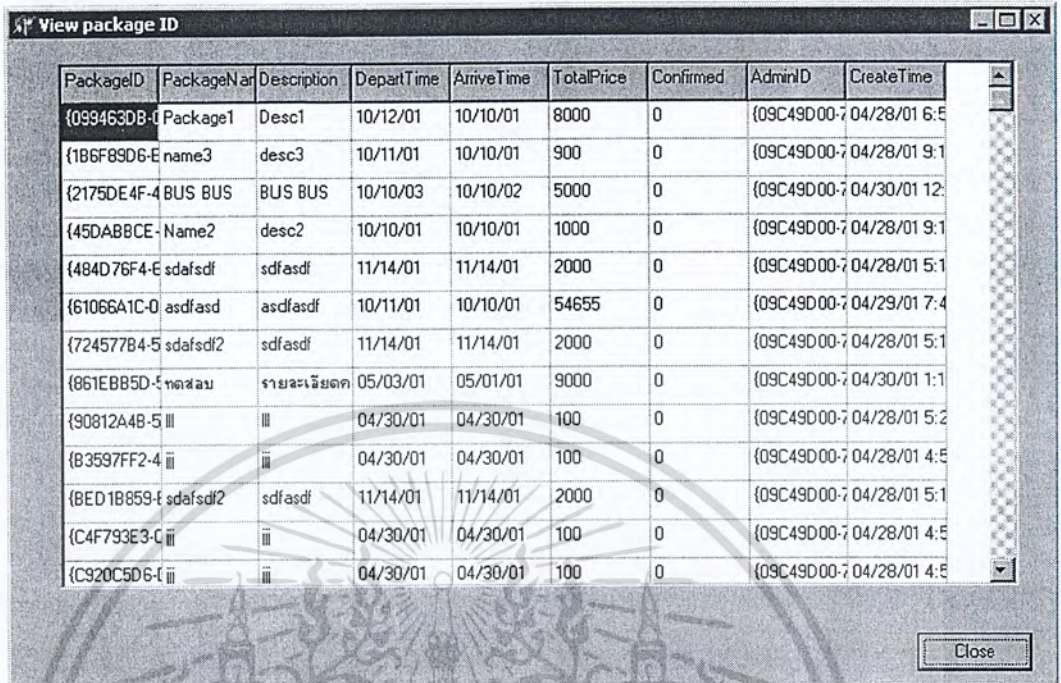
Confirm package

Close

รูปที่ 17-34 แสดงการจัดการแพคเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

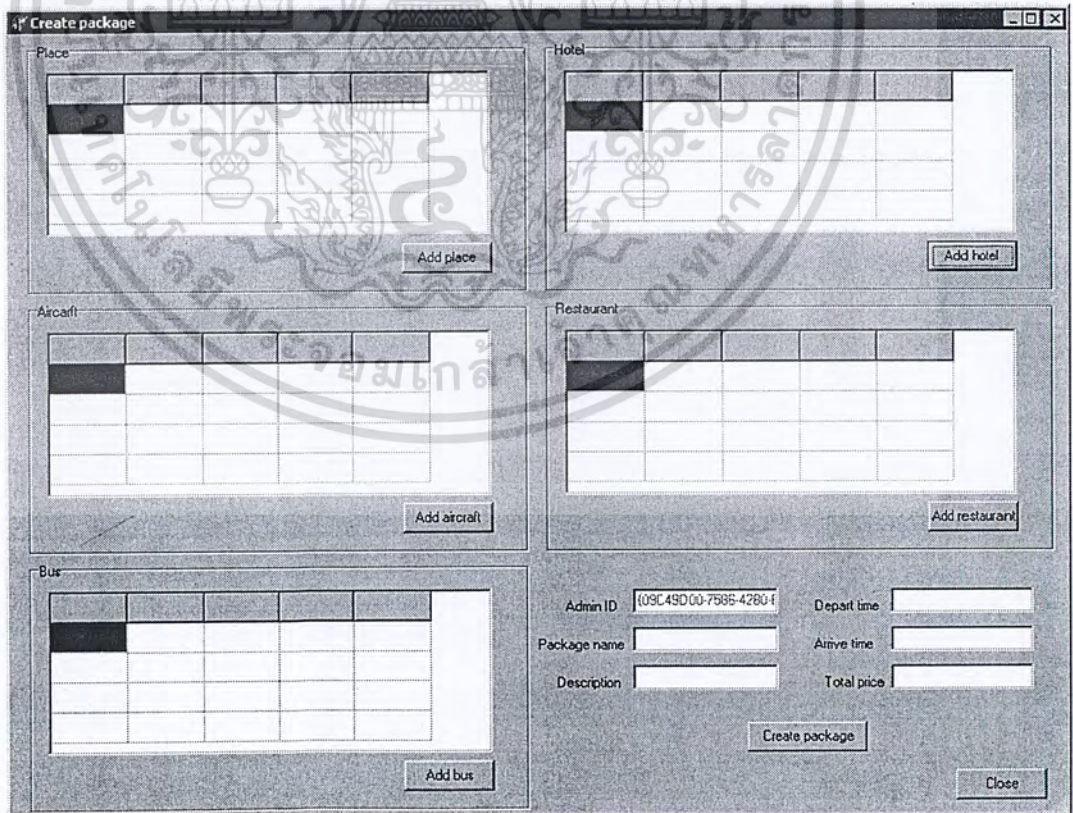
## เลือกการดูแพคเกจโดยกลุ่ม View package



PackageID	PackageName	Description	DepartTime	ArriveTime	TotalPrice	Confirmed	AdminID	CreateTime
{099463DB-f	Package1	Desc1	10/12/01	10/10/01	8000	0	{09C49D00-7	04/28/01 6:5
{1B6F89D6-E	name3	desc3	10/11/01	10/10/01	900	0	{09C49D00-7	04/28/01 9:1
{2175DE4F-4	BUS BUS	BUS BUS	10/10/03	10/10/02	5000	0	{09C49D00-7	04/30/01 12:
{45DABBCE-	Name2	desc2	10/10/01	10/10/01	1000	0	{09C49D00-7	04/28/01 9:1
{484D76F4-E	sdfasdf	sdfasdf	11/14/01	11/14/01	2000	0	{09C49D00-7	04/28/01 5:1
{61066A1C-0	asdfasf	asdfasf	10/11/01	10/10/01	54655	0	{09C49D00-7	04/29/01 7:4
{724577B4-5	sdfasdf2	sdfasdf	11/14/01	11/14/01	2000	0	{09C49D00-7	04/28/01 5:1
{861EBB5D-	รายละเอียด	รายละเอียด	05/03/01	05/01/01	9000	0	{09C49D00-7	04/30/01 1:1
{90812A4B-5	iii	iii	04/30/01	04/30/01	100	0	{09C49D00-7	04/28/01 5:2
{B3597FF2-4	iii	iii	04/30/01	04/30/01	100	0	{09C49D00-7	04/28/01 4:5
{BED1B859-f	sdfasdf2	sdfasdf	11/14/01	11/14/01	2000	0	{09C49D00-7	04/28/01 5:1
{C4F793E3-C	iii	iii	04/30/01	04/30/01	100	0	{09C49D00-7	04/28/01 4:5
{C920C5D6-f	iii	iii	04/30/01	04/30/01	100	0	{09C49D00-7	04/28/01 4:5

รูปที่ 17-35 แสดงการดูข้อมูลการจองแพคเกจ

ทำการสร้างแพคเกจโดยการเลือก Create package จะเห็นหน้าจอดังรูป



The 'Create package' window contains several sections for adding items to a package:

- Place:** A table with 5 columns and 3 rows, and an 'Add place' button.
- Hotel:** A table with 5 columns and 3 rows, and an 'Add hotel' button.
- Aircraft:** A table with 5 columns and 3 rows, and an 'Add aircraft' button.
- Restaurant:** A table with 5 columns and 3 rows, and an 'Add restaurant' button.
- Bus:** A table with 5 columns and 3 rows, and an 'Add bus' button.
- Form Fields:**
  - Admin ID: {09C49D00-7585-4280-f
  - Package name: [ ]
  - Description: [ ]
  - Depart time: [ ]
  - Arrive time: [ ]
  - Total price: [ ]
- Buttons:** 'Create package' and 'Close'.

รูปที่ 17-36 แสดงก่อนการสร้างแพคเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการเลือกสถานที่ของแพคเกจ โดยคลิกปุ่ม Add place

PlaceID	PlaceName	Country	State	Province	Description	ImageFilePath
(2BCABFCD-KMITL		Thailand		Bangkok	My Institute	
(6F35F53F-9	Hua-Ta-Kae	Thailand		Bangkok	Fresh Market	
(7F619741-2	Soi Jin-Da	Thailand		Bangkok	Food Center	
(83542CAD-GIT		USA	California	L.A.	Paul Gilbert S	

รูปที่ 17-37 แสดงการใส่สถานที่เข้าสู่แพคเกจ

เสร็จแล้วทำการ กดปุ่ม Add to package แล้วปิดหน้าต่างนี้กลับไปทำการจองโรงแรมโดยคลิกปุ่ม Add hotel

HOTELID	HOTELNAM	DESCRIPTI	ADDRESS	STAR	TELEPHONE
(008574F8-D	royal garden	near sea has	454 phuket.lf	5	66761457896
(066FA05E-S	grand hotel	v a short walk	: viale castang	5	41919712213
(0A218097-D	bernini bristol	situated in the	bernini bristol	5	064883051
(10720EEA-x	anglo americ	centre city lo	anglo americ	4	0672941

PRICEPERD	DESCRIPTI	NUMBERD	NUMBERD	NUMBERD	ROOMCLAS
450	service	1	0	2	2
500	service	2	0	4	3
550	service	0	2	5	4

รูปที่ 17-38 แสดงการจองโรงแรมของแพคเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสร็จแล้วทำการ กดปุ่ม Add to package แล้วปิดหน้าต่างนี้กลับไปทำการจองเครื่องบินโดยกดปุ่ม Add aircraft

**Airline**

AirlineID	AirlineName
1	Thai Airways
2	China Airlines
3	Air Canada

**Airport**

PlaceID	PlaceName
HIJ	Hiroshima, Jc
ITM	Osaka Itami,
KMI	Miyazaki, Jap
KIX	Kansai Airpor
MSJ	Misawa, Jap
NRT	Narita Airpor
TYO	Tokyo, Japer
HND	Haneda, Tok

**Search flight**

Number of seat: 1 From: bkk To: lax  
 Going date: 5/20/01 Airline: 1 Seat type: All  
 Class type: 1  Non stop flight  Smoking Search

AirlineName	FlightID	AircraftName	ClassName	ClassID	DepartTime	ArriveTime	SIndex	DIndex
Thai Airways	TG700	Boing 737	First Class	1	5:00:00 AM	6:00:00 AM	0	1
Thai Airways	TG702	Boing 737	First Class	1	4:00:00 PM	2:00:00 PM	0	0

**Add to package**

Number of seat: 1 From: 0 To: 1  
 Going date: 5/20/01 Price: 60300 Flight: TG700  
 Class type: 1 Add to package

รูปที่ 17-39 แสดงการจองเครื่องบินของแพคเกจ

เสร็จแล้วทำการ กดปุ่ม Add to package แล้วปิดหน้าต่างนี้กลับไปทำการจองร้านอาหารโดยกดปุ่ม Add restaurant

**Search**

RESTAURA	RESTAURA	DESCRIPTION	TOTAL_SEA	ADDRESS	STAR	TELEPHONE
{29CA7F0C-1	cafe swiss	swiss reataur	60	swiss loadge	4	6622335254
{5471BF3D-3	the thai resta	thai food	150	101 west johr	5	0262859000
{67E06B6C-1	do do sushi	japanese sus	100	5300 beach	4	7145235959
{6CC7752E-3	delphi greek	greek cuisine	80	1383 westwo	4	3104782900
{72044D0C-1	delphi greek	greek cuisine	150	7051 westwo	5	0107054114

Name: \_\_\_\_\_ Description: \_\_\_\_\_  
 Address: \_\_\_\_\_ Class: 0 Search

**Add restaurant to package**

Restaurant ID: {29CA7F0C-CD12-4456- Meal: 1  
 Reserve date: 5/21/01 Seat amount: 20 Add to package

รูปที่ 17-40 แสดงการจองร้านอาหารของแพคเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสร็จแล้วทำการ กดปุ่ม Add to package แล้วปิดหน้าต่างนี้กลับไปทำการจองรถบัสโดยกดปุ่ม Add bus

**Add bus to package**

Search bus

Start date: 05/20/2001    Number of day: 3    Number of bus: 2    Number of seat: 60

AirConditioner     Toilet   

Company Name	Model Name			Total Seat	Rent Cost	Description	Model ID	Company ID
407 พัฒนา	AirBus	-1	-1	36	1500	Cherdchai Cc 4	1	1
407 พัฒนา	Coach Premi	-1	-1	32	2000	Benz C701	2	1
407 พัฒนา	PowerBus	0	0	48	750	Volvo Bus V 6		1
เชิดชัยทัวร์	AirBus Deluxe	-1	-1	32	1750	Cherdchai Cc 3		2
เชิดชัยทัวร์	Coach VIP	-1	-1	24	2500	Volvo Bus V 1		2

Add to package

Company ID: 1    Model ID: 4    Number of bus: 2

Start date: 05/20/2001    Number of day: 3   

รูปที่ 17-41 แสดงการจองรถบัสของแพคเกจ  
เสร็จแล้วทำการใส่ข้อมูลของแพคเกจให้ครบ จะได้หน้าต่างดังนี้

**Create package**

Place

Place ID	Arrive Time	Depart Time
2BCABFCD	5/20/01	5/25/01

Hotel

Hotel ID	Book check	Book check	Price per day	Description	Number
0A218097-E	5/20/01	5/22/01	450	service	1

Aircraft

Total seat	Flight ID	Srcldx	Dstlclx	Depart date	Class type
1	TG700	0	1	5/20/01	1

Restaurant

Restaurant ID	Meal	Reserve date	Seat Amount
23CA7F0C-1	1	5/21/01	20

Bus

Company ID	Model ID	Number of bus	Start date	Number of day
1	4	2	05/20/2001	3

Admin ID: 103C49C00-7566-4280-4  
 Package name: Losangeles Tour  
 Description: Go to losangeles

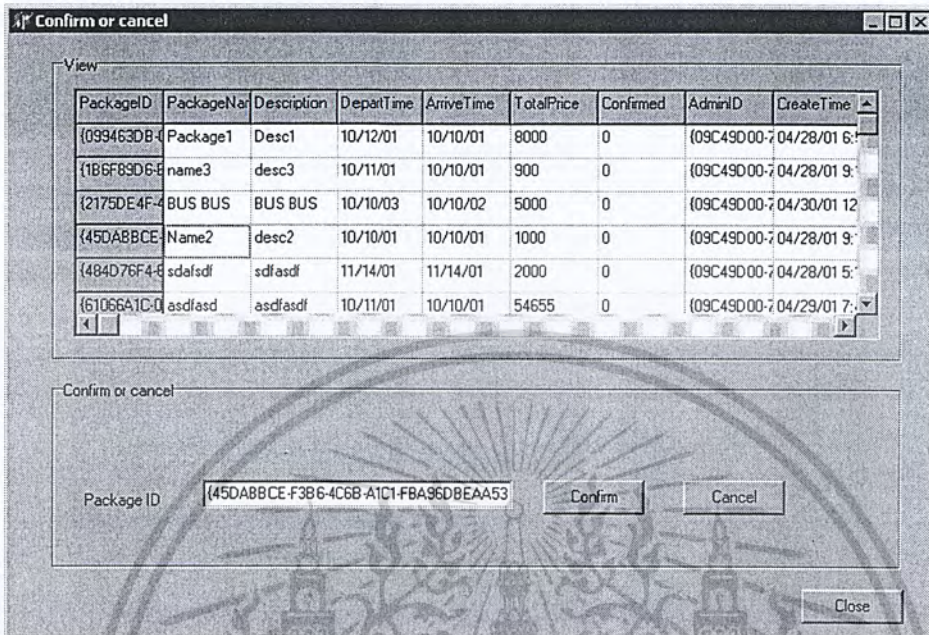
Depart time: 5/20/01  
 Arrive time: 5/25/01  
 Total price: 70000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเมื่อจองตั๋วเครื่องบินไปกลับแล้วไปใช้ประโยชน์ด้านการค้า  
รูปที่ 17-42 แสดงการจองแพคเกจเสร็จ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสร็จแล้วทำการ Create package เป็นอันเสร็จการสร้างแพคเกจ

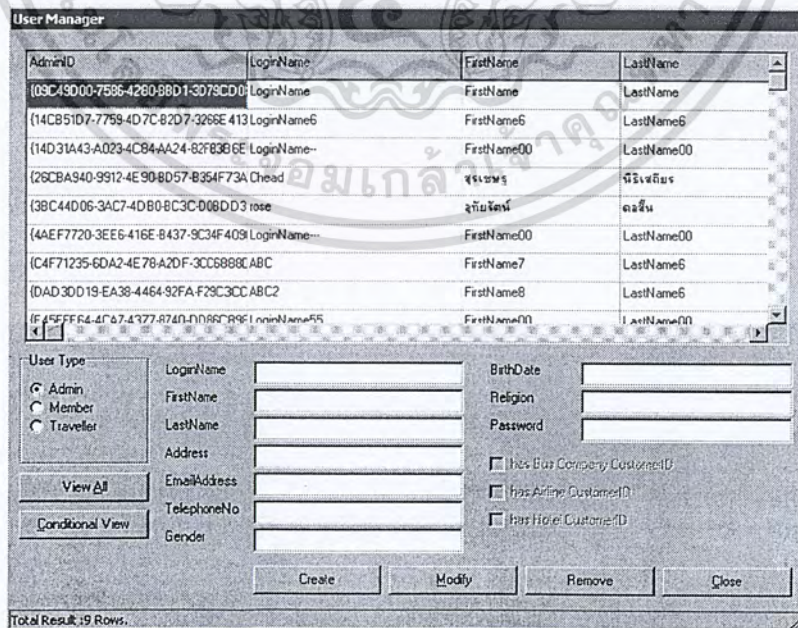
หลังจากการสร้างแพคเกจเสร็จแล้วสามารถกลับมา ตกลงหรือปฏิเสธได้ โดยการเลือก Confirm package ที่หน้าของ Package manager



รูปที่ 17-43 แสดงการตกลงหรือปฏิเสธการจอง

#### 17.2.4 การจัดการเกี่ยวกับสมาชิกของระบบ

สามารถทำการสร้าง, ลบ, แก้ไข เกี่ยวกับตัวของสมาชิกในระบบได้โดยที่หน้า Admin application ทำการเลือก User management



รูปที่ 17-44 แสดงการจัดการสมาชิกของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 17.2.5 การดูประวัติการทำงานของระบบ

สามารถดูการทำงานที่ทำได้แล้วของระบบได้โดยที่หน้า Admin application ทำการเลือก View log

LogID	Description	AdminLoginName	EventTime
{0C50A514-C18D-4C58-A1D0-1FB08AFC}	Create PackageName : name5	LoginName	04/28/01 9:44:07 PM
{3E02F499-9808-4121-BD67-78F023CE}	Create PackageName : ทดสอบ	LoginName	04/30/01 1:16:38 AM
{5C546217-0C9C-4687-97B2-782A2FFC}	Create PlaceName abcd: Place ID {7FE	LoginName	04/16/01 6:21:37 PM
{6F59C927-4431-4D74-86C2-22BC27C0}	DELETE Place	LoginName	05/07/01 4:11:41 PM
{759FF40B-F639-4C96-8F3E-80CB068A}	DELETE Place {843995BB-7675-47AE	LoginName	04/15/01 10:04:36 PM
{92E6D5FC-C5EF-4938-9CC4-E1BE4E3C}	Create PackageName : asdfasd	LoginName	04/29/01 7:47:25 PM
{9A5E1C6E-723E-4DA6-BA61-A051DB4C}	Create PlaceName PlaceName1234: Pl	LoginName	04/16/01 9:10:42 PM
{A26178CF-DFBC-4FEA-8F1A-AC63616F}	Delete Festival ID: {542052AB-9541-4EF		05/07/01 4:12:15 PM
{65675901-C078-4670-6E79-086F19C6}	Reserve Resturant for PackageTour ID		04/23/01 7:14:09 PM

LogType  
 SystemLog  BussinessLog

View Clear

Close

Total Result : 13 Rows.

รูปที่ 17-45 แสดงการดูประวัติการทำงานของระบบ

สามารถดูได้ทั้ง System log และ Business log

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 18

## เปรียบเทียบประสิทธิภาพระหว่าง COM+ และ EJB

นอกจาก COM+ และ DNA แล้วยังมีซอฟต์แวร์คอมโพเนนท์อื่นอีกซึ่งได้แก่ EJB และ J2E การเปรียบเทียบประสิทธิภาพระหว่าง COM+ และ EJB นั้นเป็นเรื่องที่ค่อนข้างที่จะยุ่งยากในการเปรียบว่าฝ่ายใดมีประสิทธิภาพเหนือกว่าฝ่ายใดอย่างชัดเจน เนื่องจากทั้งสองระบบก็มีข้อดี และข้อเด่นที่แตกต่างกันไปซึ่งขึ้นอยู่กับความต้องการของระบบ ได้แก่

- ลักษณะของระบบงาน หากมีความจำเป็นต้องใช้งานกับระบบงานที่หลากหลาย หรือต้องการขายซอฟต์แวร์คอมโพเนนท์ให้กับองค์กรภายนอก หรือองค์กรเหล่านั้นใช้งานระบบปฏิบัติการอื่นนอกจาก Windows 2000 EJB ก็เป็นทางเลือกที่เหมาะสมกว่า COM+ อยู่พอสมควร แต่หากว่าองค์กรนั้นพัฒนาซอฟต์แวร์คอมโพเนนท์ขึ้นเพื่อใช้งานเองภายใต้ระบบปฏิบัติการ Windows 2000 ทั้ง COM+ และ EJB ก็มีความเหมาะสมเสมอกัน
  - ความสามารถเดิมของผู้พัฒนา หากผู้พัฒนามีความชำนาญในตัวจาวาแล้ว EJB ก็เป็นทางเลือกที่เหมาะสมกว่า เช่นเดียวกันหากผู้พัฒนามีความชำนาญในตัวภาษาโปรแกรมอื่น COM+ จะเหนือกว่าเล็กน้อย เนื่องจาก EJB นั้นมีข้อจำกัดที่สามารถพัฒนาโดยใช้ภาษาจาวาเท่านั้นซึ่งแตกต่างจาก COM+
  - ค่าใช้จ่ายสำหรับเซิร์ฟเวอร์ เนื่องจากสภาพแวดล้อมใน Windows 2000 นั้นเหมาะสมกับการพัฒนา COM+ อยู่แล้วโดยไม่ต้องเพิ่มเติมโปรแกรมอื่น หากแต่การพัฒนาด้วย EJB นั้นต้องการคอนเทนเนอร์และเว็บเซิร์ฟเวอร์เพิ่มเติม จากระบบปฏิบัติการ
- ซึ่งสามารถสรุปได้ดังตารางที่ 18-1

คุณสมบัติ	COM+	EJB
ภาษาในการพัฒนาคอมโพเนนท์	ทุกภาษาโปรแกรม	จาวาเท่านั้น
แพลตฟอร์ม	Windows 2000	ทุกแพลตฟอร์ม
จำนวนบริษัทที่นำไปใช้ (ข้อมูลปลายปี พศ. 2543)	บริษัทไมโครซอฟท์	มากกว่า 30
เทคโนโลยีที่เกี่ยวข้อง	COM TI, MSMQ, OLE DB	RMI/JNI, COBRA
โพรโตคอล	DCOM	IIOOP
สถานะในคอมโพเนนท์	ไม่มี	มี
คอมโพเนนท์ที่ถาวร (Persistence Component)	มี	มี
ทรานแซกชันแบบเกี่ยวเนื่อง (Nested Transaction)	สนับสนุน	ไม่สนับสนุน
การไหลคบาลานซ์ซึ่งในมิดเดิลแวร์	มี (โดยใช้ Application Center 2000)	มี

ตารางที่ 18-1 เปรียบเทียบคุณสมบัติระหว่าง COM+ และ EJB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ	COM+	EJB
การแคชข้อมูลในมิดเดิลเทียร์	ไม่มี (In memory database ถูกถอด จากวินโดวส์ 2000 ก่อนออกจาก จำหน่ายจากข้อจำกัดในด้าน เทคโนโลยี)	มี
การทำงานแบบอะซิงโครนัส (โดยใช้ Queued Component)	มี	ไม่มี
มีผู้ผลิตรายเดียว	ใช่	ไม่ใช่
มิดเดิลแวร์มาพร้อมกับระบบปฏิบัติการ	ใช่	ไม่ใช่
จำนวนโปรเซสเซอร์ต่อเครื่องสูงสุด	32	เกิน 256
จำนวนโปรเซสเซอร์สูงสุดต่อคลัสเตอร์	ไม่จำกัด	ไม่จำกัด
เครื่องมือในการพัฒนา	ไม่จำกัด	ไม่จำกัด แต่แนะนำว่าควร ใช้ Microsoft Visual Studio

ตารางที่ 18-1(ต่อ) เปรียบเทียบคุณสมบัติระหว่าง COM+ และ EJB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 19

## บทวิจารณ์ และสรุป

## 19.1 สรุปผลการดำเนินงาน

จากการดำเนินงานโครงการนี้การดำเนินงานได้แบ่งเป็น 3 ส่วนคือ

- ศึกษาทฤษฎีพื้นฐานของ COM, COM+ , DNA และ องค์ประกอบอื่น ๆ ที่เกี่ยวข้อง
- ทำการออกแบบระบบงานตัวอย่าง OlalaTour
- ทำการสร้างระบบงานโดยใช้ DNA ร่วมกับ Borland Delphi 5.0 และ Microsoft Visual Basic 6.0

สำหรับระบบงานที่ได้ทดลองสร้างนั้นก็ได้มีการใช้เทคโนโลยีต่าง ๆ ที่ได้ศึกษามา ซึ่งก็ประสบความสำเร็จพอสมควร โดยที่สามารถสร้างระบบงานซึ่งมีลักษณะเป็น มัลติ-tier โดยมี tier ต่าง ๆ 4 ส่วนคือ

- โคลเอนต์ซึ่งเป็นทั้งเว็บเบราว์เซอร์ และ เป็นทั้งแพลตฟอร์มไคลเอนต์
- เว็บเซิร์ฟเวอร์ ที่ใช้ IIS และมีเซิร์ฟเวอร์สคริปเป็น ASP
- แอปพลิเคชันเซิร์ฟเวอร์ ใช้ COM+ รันไทม์
- ดาตาเบสเซิร์ฟเวอร์เป็น Microsoft SQL Server 2000

ข้อจำกัดในการเรียกใช้บางส่วนที่ต้องมีสถานะ ( Stateful ) ซึ่งได้แก่รถเงินทางกลุ่มแก้ปัญหาโดยใช้คุกกี้ ( Cookie ) เก็บสถานะของรถเงินไว้ในฝั่งไคลเอนต์โดยเก็บเป็นสตริง เมื่อผู้ใช้ทำการจอง ก็จะส่งสตริงนี้มา ฝั่งคอมโพเนนท์ก็จะทำการตัดสตริง แล้วส่งต่อไปยังคอมโพเนนท์ที่ต้องเรียกใช้อีกต่อไป

นอกจากนั้นยังมีข้อจำกัดในการเรียกใช้คอมโพเนนท์ข้ามโดเมน ทำให้สถาปัตยกรรมของระบบต้องเปลี่ยนจากเดิมที่จะให้คอมโพเนนท์อยู่ต่างโดเมน (โดยคอมโพเนนท์จะอยู่โดเมนเดียวกับเครื่องดาตาเบสเซิร์ฟเวอร์ของตัวมัน) เป็น คอมโพเนนท์อยู่โดเมนเดียวกันแล้วเรียกไปยังดาตาเบสเซิร์ฟเวอร์ซึ่งอยู่ต่าง โดเมน โดยต้องทำการทรีสตีโดเมนกัน

ปัญหาของการทำงานจะอยู่ในส่วนการศึกษาทฤษฎีและการออกแบบระบบ ซึ่งทำให้มีผลกระทบการสร้างระบบงาน ปัญหาของการศึกษาทฤษฎี นั้นเนื่องจากเทคโนโลยี COM, COM+ และ DNA นั้นยังเป็นเทคโนโลยีซึ่งยังใหม่ ทำให้มีข้อมูลเหล่านี้น้อย และข้อมูลบางส่วนก็ยังเป็นเพียงข้อมูล ที่ถูกตีพิมพ์ก่อนที่ตัวผลิตภัณฑ์จริงจะ ออกซึ่งทำให้เกิดความคลาดเคลื่อน เมื่อตัวผลิตภัณฑ์จริงออกมา นอกจากนี้อาจผลิตภัณฑ์บางตัวนั้นยังไม่เสร็จ และจำเป็นต้องใช้ในางงานบางส่วน ทำให้ต้องรอ และเมื่อผลิตภัณฑ์ออกมาก็ยังใหม่และไม่สมบูรณ์เท่าที่ควร ส่วนปัญหาในการออกแบบนั้นก็เนื่องจากยังไม่มีทฤษฎี ที่เป็นทางการของการออกแบบระบบงานเชิงคอมโพเนนท์ ทางกลุ่มจึงทำการลองผิดลองถูก และ ขาดประสบการณ์ในการออกแบบระบบ ทำให้ระบบที่ออกแบบมาในครั้งแรกนั้นบกพร่อง จนต้องแก้ไขโครงสร้างของระบบอยู่หลายครั้ง ซึ่งปัญหาทั้งสองส่วนนี้ส่งผลกระทบต่อ การทำงานในส่วนการสร้างระบบงาน ซึ่งมีเวลาจำกัด ทำให้งานที่ออกมายังไม่สมบูรณ์เสียทีเดียว

## 19.2 แนวทางการพัฒนาต่อ

สำหรับการพัฒนาซอฟต์แวร์เชิงคอมโพเนนท์ที่ทางกลุ่มได้ศึกษาและลองสร้าง ระบบงานจริงมานั้นทางกลุ่มยังไม่ได้ทดสอบความสามารถในเรื่องการ Load Balancing ซึ่งต้องใช้โปรแกรม Microsoft Application Center 2000 เนื่องจากตัวผลิตภัณฑ์ที่ทางกลุ่มได้มายังไม่สมบูรณ์ และขาดแคลนเครื่อง จึงไม่สามารถหาข้อสรุปในเรื่องนี้ได้ นอกจากนี้เนื่องจากข้อจำกัดทางเวลาทำให้ยังศึกษาในเรื่องของการเรียกใช้คอมโพเนนท์ที่เป็นเทคโนโลยี อื่นคือ EJB รวมทั้งการทำงานแบบอะซิงโครนัสโดยผ่าน MSMQ (Microsoft Message Queue) ซึ่งคิดว่าเป็นแนวทางในการพัฒนาซึ่งน่าจะเป็นสิ่งที่ควรศึกษาและพัฒนาต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### ยูนิฟิเคชันโมเดลลิงแลงเกวจ (Unification Modeling Language)

#### 1 ความรู้เบื้องต้น

ยูเอ็มแอล (UML) หรือยูนิฟิเคชัน โมเดลลิงแลงเกวจเป็นภาษาที่ถูกสร้างขึ้นเพื่อใช้แสดงโมเดลทางออบเจกต์โอเรียนเท็ด เกิดจากการพัฒนาของผู้นำทางเทคโนโลยีทางออบเจกต์ 3 ท่าน คือเกรดี บูช (Grady Booch) อิวาร์ จาคอบเซน (Ivar Jacobsen) และจิม รัมกอกซ์ (Jim Rumbaugh)

ยูเอ็มแอลนั้นได้รวมแนวความคิดของวิธีการต่างๆทางออบเจกต์เข้าด้วยกัน จุดประสงค์ของยูเอ็มแอลคือการสร้างโมเดลในการพัฒนาที่เข้าใจง่ายแต่สามารถนำไปใช้ประยุกต์ได้กับทุกๆระบบตัว ยูเอ็มแอลเองนั้นได้รับการสนับสนุนจากผู้นำในอุตสาหกรรมทางด้านคอมพิวเตอร์หลาย ๆ ราย

ใน UML มีโมเดลที่แสดงมุมมองต่อระบบที่ไม่เหมือนกัน ซึ่งประกอบไปด้วย

- ยูสเคสไดอะแกรม (Use-case Diagram) ใช้ในการแสดงการติดต่อระหว่างระบบกับผู้ใช้
- คลาสไดอะแกรม (Class Diagram) ใช้แสดงโครงสร้างทางลอจิกของระบบ
- ออบเจกต์ไดอะแกรม (Object Diagram) ใช้แสดงออบเจกต์และความสัมพันธ์ระหว่างออบเจกต์
- สเตทไดอะแกรม (State Diagram) ใช้แสดงพฤติกรรมของออบเจกต์แต่ละตัวในระบบ
- คอมโพเนนต์ไดอะแกรม (Component Diagram) ใช้แสดงโครงสร้างทางกายภาพของซอฟต์แวร์
- ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram) ใช้แสดงการติดต่อระหว่างฮาร์ดแวร์และซอฟต์แวร์
- อินเตอร์แอคทีฟไดอะแกรม (Interactive Diagram) ใช้แสดงพฤติกรรมของระบบ
- แอกทิวิตีไดอะแกรม (Activity Diagram) ใช้แสดงการไหลเวียนของอีเวนต์ในยูสเคส

#### 2 การวิเคราะห์ระบบ

โดยปกติระบบจะต้องติดต่อกับสภาพแวดล้อมภายนอกซึ่งการที่เราจะรู้ว่าระบบติดต่อกับสภาวะแวดล้อมภายนอกอย่างไรนั้นก็จำเป็นต้องมีการศึกษาความต้องการของระบบ หรือ Requirement Analysis จากเดิมในการวิเคราะห์ระบบแบบโครงสร้างเมื่อเราวิเคราะห์ความต้องการของระบบเป็นที่เรียบร้อยแล้ว จะได้ไดอะแกรม 2 ตัว คือ คอนเท็กซ์ไดอะแกรม และ ดาต้าโฟลว์ไดอะแกรม (Data flow Diagram) แต่สำหรับการวิเคราะห์ระบบด้วยออบเจกต์โอเรียนเท็ด หลังจากวิเคราะห์ความต้องการของระบบแล้วจะได้เป็น Use-Case Diagram และ External Event Context จะจำลองระบบเป็นเหมือนกับออบเจกต์ใหญ่ ๆ ตัวหนึ่ง ซึ่งประกอบด้วยออบเจกต์ย่อยภายใน และออบเจกต์ของระบบนี้จะแสดงการติดต่อสื่อสารทั้งการรับและส่งเมสเสจ กับออบเจกต์ที่เป็นแอกเตอร์ (Actor) ภายนอกส่วน Use-case จะแสดงฟังก์ชันการทำงาน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

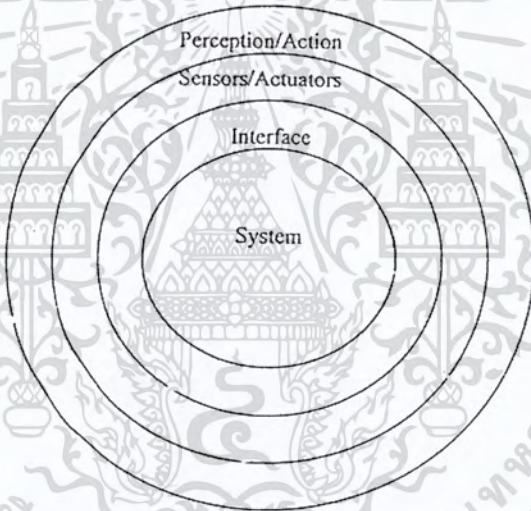
หลักของระบบและโพรโทคอล หรือข้อตกลงในการแลกเปลี่ยนเมสเสจที่จำเป็นเพื่อให้ระบบสามารถติดต่อกับแอกเตอร์ภายนอกได้อย่างถูกต้องตามความต้องการของระบบ

**อีเวนต์ภายนอก (External Event)**

ซิสเต็มคอนเท็กซ์ (System Context) เป็นสิ่งที่แสดงขอบเขตของระบบที่ผู้พัฒนาหรือผู้ใช้สนใจต่อระบบ ซึ่งเราจะใช้คอนเท็กซ์ไดอะแกรม (Context Diagram) ซึ่งจะแสดงสถานะแวดล้อมที่มีต่อระบบ ซึ่งประกอบไปด้วยผู้กระทำหรือแอกเตอร์ที่กระทำต่อระบบ นอกจากนี้ยังแสดงการแลกเปลี่ยนเมสเสจและอีเวนต์ของระบบกับสถานะแวดล้อมภายนอกอีกด้วย

**ออบเจกต์ระดับคอนเท็กซ์ (Context-Level Objects)**

คอนเท็กซ์ไดอะแกรมจะทำหน้าที่แสดงการติดต่อระหว่างออบเจกต์ของระบบกับออบเจกต์ภายนอกได้ แบ่งออบเจกต์ของระบบและออบเจกต์ภายนอกโดยการจำลองระบบออกมาเป็นหัวหอม (Onions) ซึ่งจะมีผิวเป็นชั้น ๆ ซึ่งแต่ละชั้นก็มีความสำคัญต่อระบบที่แตกต่างกันไปดังรูปที่ ก-1



รูปที่ ก-1 การเปรียบเทียบระบบแบบหัวหอม

สำหรับผิวนอกสุดของภาพเปรียบเทียบเป็น Perception/Action ซึ่งเป็นส่วนแสดงแอกชันของระบบ ซึ่งชั้นนี้ทำให้ผู้ใช้สามารถเห็นการกระทำของระบบที่มีต่อสภาพแวดล้อมได้

สำหรับชั้นต่อไป Sensors/Actuators เป็นชั้นของอุปกรณ์ที่ทำหน้าที่ติดต่อกับสิ่งแวดล้อม

สำหรับชั้นต่อไปอินเตอร์เฟซจะเป็นอุปกรณ์ที่ใช้ในการสื่อสาร และชั้นสุดท้ายคือชั้นของระบบศูนย์กลางที่ทำหน้าที่ตัดสินใจและประมวลผลข้อมูล

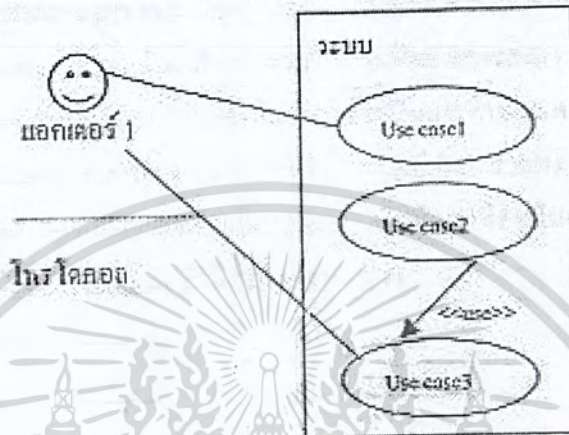
**ยูสเกส**

คอนเท็กซ์ไดอะแกรมที่ได้จากการวิเคราะห์นั้นจะบอกถึงการติดต่อระหว่างสิ่งแวดล้อมภายนอกกับระบบ สำหรับการวิเคราะห์ระบบแบบ ออบเจกต์โอเรียนเท็ดนั้นสามารถสร้างยูสเกสไดอะแกรมที่เพิ่มกรณีการติดต่อระหว่างระบบกับแอกเตอร์ภายนอก นอกจากนี้ยังใช้ในการตรวจสอบความถูกต้องของ

**คอนเท็กซ์ไดอะแกรม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยูสเคสไดอะแกรมจะแสดงมุมมองต่อฟังก์ชันการทำงานหลักๆของระบบ ซึ่งผู้พัฒนาระบบสามารถใช้ในการสื่อสารกับผู้ใช้ระบบได้ด้วยยูสเคสไดอะแกรม ในระดับบนสุดนั้นจะแสดงขอบเขตของระบบทั้งหมดซึ่งเป็นมุมมองจากฟังก์ชัน การทำงานไม่ใช่มุมมองต่อคลาส ออบเจกต์ที่จะมีในระบบโดยฟังก์ชันการทำงานของระบบนั้นจะแทนที่ด้วยวงรีและอยู่ในระบบใหญ่ ซึ่งแสดง โดยกรอบสี่เหลี่ยม โดยมีออบเจกต์กระจายอยู่รอบนอกระบบ ออบเจกต์ภายนอกเหล่านี้จะสัมพันธ์กับยูสเคสหนึ่ง ๆ อย่างจะมีความสัมพันธ์กับยูสเคสอื่น ๆ ซึ่งเราสามารถแทนได้โดยการลากเส้นแสดงความสัมพันธ์ระหว่างยูสเคสดังรูปที่ ก-2



รูปที่ ก-2 ยูสเคสไดอะแกรม

#### การกำหนดยูสเคส

ในการวิเคราะห์ระบบส่วนใหญ่จะใช้วิธีการกำหนดซีนารีโอ (Scenario) ขึ้นมาก่อนซึ่งซีนารีโอที่กำหนดขึ้นมานี้อาจจะมีมากกว่ายูสเคสจากนั้นนักพัฒนาจะเลือกซีนารีโอที่สำคัญต่อระบบขึ้นมาเพื่อทำเป็นยูสเคส

#### ซีนารีโอ

ซีนารีโอคือตัวตนของยูสเคสในโมเดล ซีนารีโอจะกำหนดลำดับของเมสเสจของระบบ รวมถึงโครงสร้างของออบเจกต์ที่ทำงานร่วมกันเพื่อให้ระบบสามารถทำหน้าที่ที่รับผิดชอบได้ ข้อดีของซีนารีโอคือทำให้ผู้พัฒนาระบบและผู้ใช้สามารถเข้าใจการทำงานของระบบได้ง่ายขึ้น นอกจากนี้ยังใช้เป็นเครื่องมือในการตรวจสอบความต้องการของระบบ

ในขั้นตอนของการวิเคราะห์ระบบนั้น เมื่อเราได้คอนเท็กซ์ไดอะแกรมและยูสเคสไดอะแกรมแล้วเราก็วิเคราะห์ต่อโดยแยกระบบออกเป็นออบเจกต์ย่อยๆจากนั้นก็กำหนดซีนารีโอขึ้นสำหรับแต่ละยูสเคส โดยซีนารีโอนี้จะแสดงการรับเมสเสจระหว่างออบเจกต์ที่เรากำหนดขึ้นมาเพื่อทำงานที่รับผิดชอบตามยูสเคสได้สำเร็จ จะเห็นว่าในขั้นตอนนี้จะเป็นการตรวจสอบยูสเคสและออบเจกต์ที่เรากำหนดขึ้นว่าครบหรือไม่

การวิเคราะห์ซีนารีโอไม่มีกฎเกณฑ์ที่แน่นอน แต่อาศัยความชำนาญของผู้วิเคราะห์และความเข้าใจในระบบ

ยูเอ็มแอลมีโมเดลที่รองรับการวิเคราะห์ซีนารีโออยู่ 2 โมเดลคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

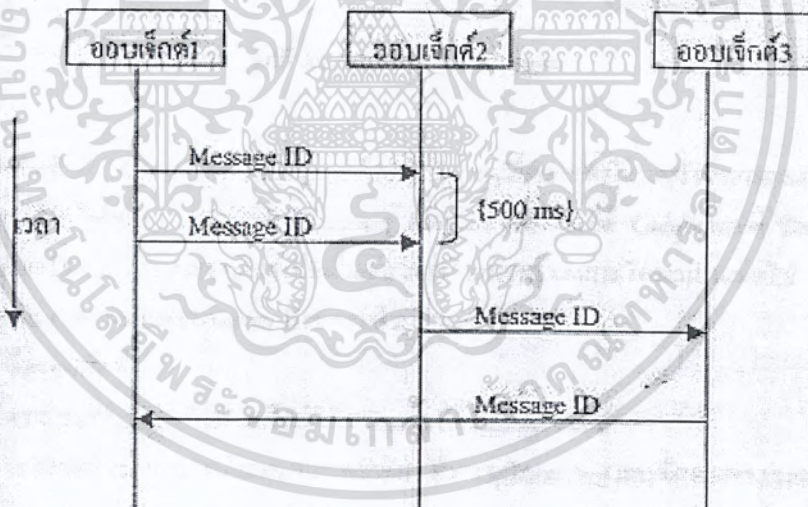
ซีเควนไดอะแกรม (Sequence Diagram) ที่แสดงถึงลำดับของเมสเสจที่ออบเจกต์ในระบบส่งและรับซึ่งกันและกัน

คอลแลบบอเรชันไดอะแกรม (Collaboration Diagram) จะแสดงโครงสร้างของออบเจกต์ที่ทำงานร่วมกันเป็นหลัก

### ซีเควนไดอะแกรม

ส่วนประกอบของซีเควนไดอะแกรมจะมีเส้นในแนวตั้งหมายถึงออบเจกต์โดยมีชื่อของออบเจกต์เขียนอยู่ด้านบนหรือคั่นล่างของเส้น ส่วนเส้นในแนวนอนนั้นแสดงถึงเมสเสจ เส้นตรงที่แทนเมสเสจทุกเส้นจะต้องมีออบเจกต์ต้นกำเนิดของเมสเสจ ส่วนจุดสิ้นสุดของเมสเสจก็จะเป็นออบเจกต์ปลายทาง โดยแต่ละเส้นของเมสเสจจะมีชื่อกำกับอยู่ โดยชื่อนี้อาจมีการเพิ่มรายการของพารามิเตอร์และค่าคงที่คืนกลับมาด้วย สำหรับเวลานั้นจะนับจากบนลงล่าง โดยสเกลของแกนตั้งจะบอกเพียงลำดับเมสเสจเท่านั้นไม่สามารถใช้เป็นอัตราส่วนในการคำนวณหาเวลาระหว่างเมสเสจได้

ซีเควนไดอะแกรมส่วนใหญ่จะประกอบไปด้วยองค์ประกอบต่าง ๆ คือ ออบเจกต์ เมสเสจ คาบเวลา และสัญลักษณ์ของชีนารีโอ นอกจากนี้ยังสามารถเพิ่มข้อมูลที่จำเป็นเข้าไปในซีเควนไดอะแกรมได้ สำหรับตัวอย่างซีเควนไดอะแกรมที่เขียนขึ้นดังรูปที่ ก-3



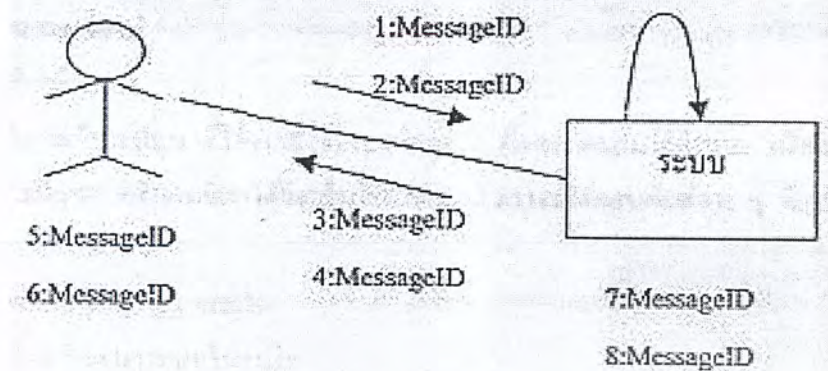
รูปที่ ก- 3 ซีเควนไดอะแกรม

สำหรับซีเควนไดอะแกรมที่ได้กล่าวไปจะมีเมสเสจในรูปแบบเพียร์ทูเพียร์ (Peer-to-Peer) คือมีออบเจกต์ตัวหนึ่งส่งเมสเสจให้ออบเจกต์อีกตัวหนึ่ง แต่ก็สามารถมีเมสเสจแบบกระจายได้ ซึ่งเมสเสจที่จะส่งออกจากออบเจกต์หนึ่งไปให้กับออบเจกต์อีกหลาย ๆ ตัว

### คอลแลบบอเรชันไดอะแกรม

คอลแลบบอเรชันไดอะแกรมเป็นเครื่องมือสำหรับใช้แสดงชีนารีโออีกตัวหนึ่งที่มีประโยชน์ คอลแลบบอเรชันไดอะแกรมนั้นจะแสดงข้อมูลพื้นฐานเช่นเดียวกับซีเควนไดอะแกรม แต่ความแตกต่างคือ ซีเควนไดอะแกรมนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนูญเตเห็นาเบเซบระเขชนตานิกริวค้ำไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แควนไดอะแกรมจะแสดงลำดับของเมสเสจเป็นหลัก ส่วนคอลเลบบอเรนซ์ไดอะแกรมนั้นจะแสดงโครงสร้างของออบเจ็กต์ที่ทำงานร่วมกันเป็นหลัก ตัวอย่างของคอลเลบบอเรนซ์ไดอะแกรมดังรูปที่ ก-4



รูปที่ ก-4 คอลเลบบอเรนซ์ไดอะแกรม

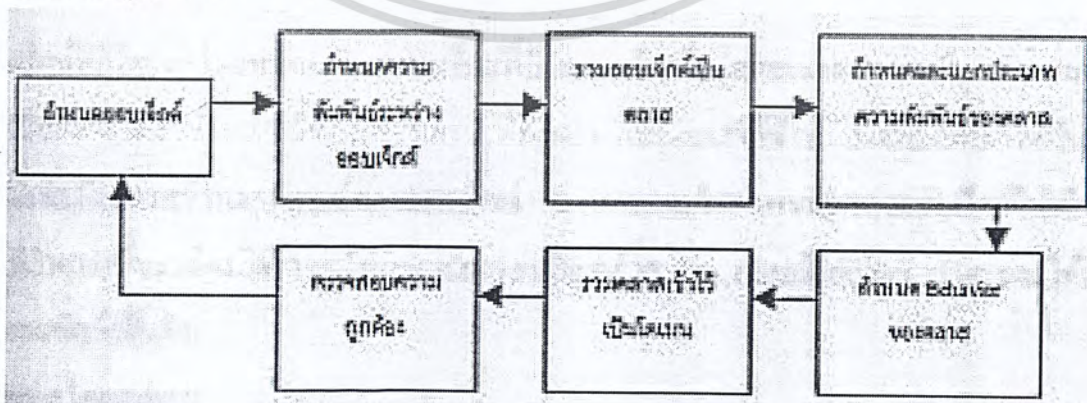
ข้อเสียคอลเลบบอเรนซ์ไดอะแกรมคือ เมื่อมีการเปลี่ยนแปลงโครงสร้างของออบเจ็กต์แล้ว จำเป็นต้องมีการแก้ไขไดอะแกรมมากกว่าซีแควนไดอะแกรม อย่างไรก็ตามคือคอลเลบบอเรนซ์ไดอะแกรมก็สามารถใช้แสดงโครงสร้างของออบเจ็กต์เพื่อเป็นแนวทางในการสร้างคลาสไดอะแกรมต่อไป

สำหรับคอลเลบบอเรนซ์ไดอะแกรมประกอบด้วยส่วนต่าง ๆ ดังนี้ คือ

- ชื่อออบเจ็กต์
- ความสัมพันธ์ระหว่างออบเจ็กต์ที่มีการแลกเปลี่ยนเมสเสจกัน
- เมสเสจที่ประกอบไปด้วยหมายเลขลำดับ ตัวระบุ (Identifier) และทิศทางของเมสเสจ

**กระบวนการทางออบเจ็กต์**

ยูสเคสไดอะแกรมและคอนเท็กซ์ไดอะแกรมที่เราสร้างขึ้นมานั้นเป็นจุดเริ่มต้นในการวิเคราะห์ระบบแบบออบเจ็กต์โอเรียนเท็ดซึ่งเมื่อวิเคราะห์ระบบเสร็จเรียบร้อยแล้วจะได้เป็นโมเดล ซึ่งประกอบด้วยออบเจ็กต์ คลาส ความสัมพันธ์ระหว่างออบเจ็กต์และความสัมพันธ์ระหว่างคลาส ลำดับขั้นตอนการสืบทอดคุณสมบัติซึ่งถือว่าเป็นคุณสมบัติที่สำคัญที่สุดของออบเจ็กต์โอเรียนเท็ด กระบวนการทางออบเจ็กต์แสดงได้ดังรูปที่ ก-5



รูปที่ ก-5 ขั้นตอนการวิเคราะห์แบบออบเจ็กต์โอเรียนเท็ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การกำหนดขอบเจ็ท

การกำหนดขอบเจ็ทโดยมากใช้วิธีการขีดเส้นใต้คำนาม วิธีการนี้ใช้กับคำอธิบายปัญหาที่ได้จากผู้ไ้ระบบโดยตรง คือการขีดเส้นใต้คำนาม หรือ วลี ที่เป็นคำนามอยู่ในคำอธิบายปัญหา หลังจากที่เรได้ขีดเส้นใต้คำนามแล้วเราจะได้ออบเจ็ทต่างๆ ที่แบ่งออกเป็น 3 ประเภท คือ

- ออบเจ็ทที่พัฒนาระบบสนใจ
- ออบเจ็ทที่พัฒนาระบบไม่สนใจ
- คุณสมบัติของออบเจ็ท

โดยจุดประสงค์หลักในการกำหนดออบเจ็ทคือ ต้องการหาออบเจ็ทที่ผู้พัฒนาสนใจ ส่วนออบเจ็ทที่ไม่ได้สนใจนั้นก็อาจจะมียุทธศาสตร์การวิเคราะห์ระบบในภายหลัง

กลยุทธ์ในการขีดเส้นใต้คำนามเพื่อกำหนดออบเจ็ทนั้นเป็นวิธีที่ง่ายและรวดเร็วในการค้นหาออบเจ็ทรวมทั้งคุณสมบัติของออบเจ็ทจากคำอธิบายปัญหา แต่ว่าการใช้กลยุทธ์นี้ทำให้เราได้ออบเจ็ทที่ไม่สนใจมากอีกทั้งยังมีความกำกวมซึ่งผู้วิเคราะห์ระบบจำเป็นต้องแก้ปัญหาเหล่านี้ด้วย

### การกำหนดความสัมพันธ์ระหว่างออบเจ็ท

จะดูจากการส่งเมสเสจเป็นหลักเพราะว่าเมสเสจทุกเมสเสจจะหมายถึงความสัมพันธ์ระหว่างออบเจ็ทซึ่งเป็นผู้ส่งเมสเสจกับออบเจ็ทซึ่งเป็นผู้รับเมสเสจ

### การกำหนดคุณสมบัติของออบเจ็ท

คุณสมบัติของออบเจ็ทเป็นส่วนข้อมูลของออบเจ็ทที่ไม่สามารถแบ่งแยกย่อยลงไปได้อีก ถ้าในระหว่างการวิเคราะห์พบว่าคุณสมบัติที่กำหนดยังสามารถแยกย่อยลงไปได้ แสดงว่าควรกำหนดคุณสมบัตินั้นเป็นออบเจ็ทที่อยู่ภายใน

ในระบบงานทั่วไปบางครั้งพบว่าบางออบเจ็ทไม่จำเป็นต้องมีคุณสมบัติ แต่มีเพียงเมธอดก็เพียงพอแล้ว

### การกำหนดคลาส

ในการกำหนดออบเจ็ทโดยใช้คำอธิบายของปัญหานั้น เราจะได้ออบเจ็ทออกมามากมาย โดยมีหลายออบเจ็ทที่มีโครงสร้างเหมือนกัน ออบเจ็ทที่มีโครงสร้างข้อมูลและเมธอดเหมือนกัน เราสามารถโมเดลให้เป็นคลาสได้ ในขั้นตอนนี้ของการวิเคราะห์คือการกำหนดคลาสสำหรับแต่ละออบเจ็ทในระบบ

คลาสเป็นการสร้างเอกลักษณ์ของออบเจ็ท เราสามารถกำหนดชนิดของออบเจ็ทให้เป็นคลาสได้ คลาสที่กำหนดขึ้นจะต้องมีตัวระบุให้กับแต่ละออบเจ็ทด้วย เช่น ออบเจ็ทบัญชี เราอาจให้ตัวระบุเป็นหมายเลขบัญชีเป็นต้น

### คลาสไดอะแกรม

คลาสไดอะแกรมเป็นไดอะแกรมที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบโดยใช้วิธีเอกสารทั้งออบเจ็ทโอเรียนเต็ท เนื่องจากจะแสดงโครงสร้างของออบเจ็ทและคลาสที่มีในระบบรวมทั้งแสดงว่าความสัมพันธ์ใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ด้วย คลาสไดอะแกรมจะเป็นโครงสร้างหลักของระบบและใช้ในการแยกย่อยรายละเอียดของระบบด้วยวิธีทางออบเจกต์โอเรียนเท็ด อย่างไรก็ตามก็ดีคลาสไดอะแกรมนั้นอาจจะมีลักษณะคล้ายออบเจกต์ไดอะแกรมแต่ว่าคลาสไดอะแกรมจะแสดงเฉพาะคลาส โดยไม่แสดงอินสแตนซ์หรือออบเจกต์ทั้งหมด

ในบางกรณีอาจจำเป็นต้องเพิ่มคลาสที่แสดงความสัมพันธ์ (Association Class) เนื่องจากความสัมพันธ์นี้สามารถมีคุณสมบัติและเมธอดเป็นของตัวเอง

### ความสัมพันธ์ระหว่างออบเจกต์

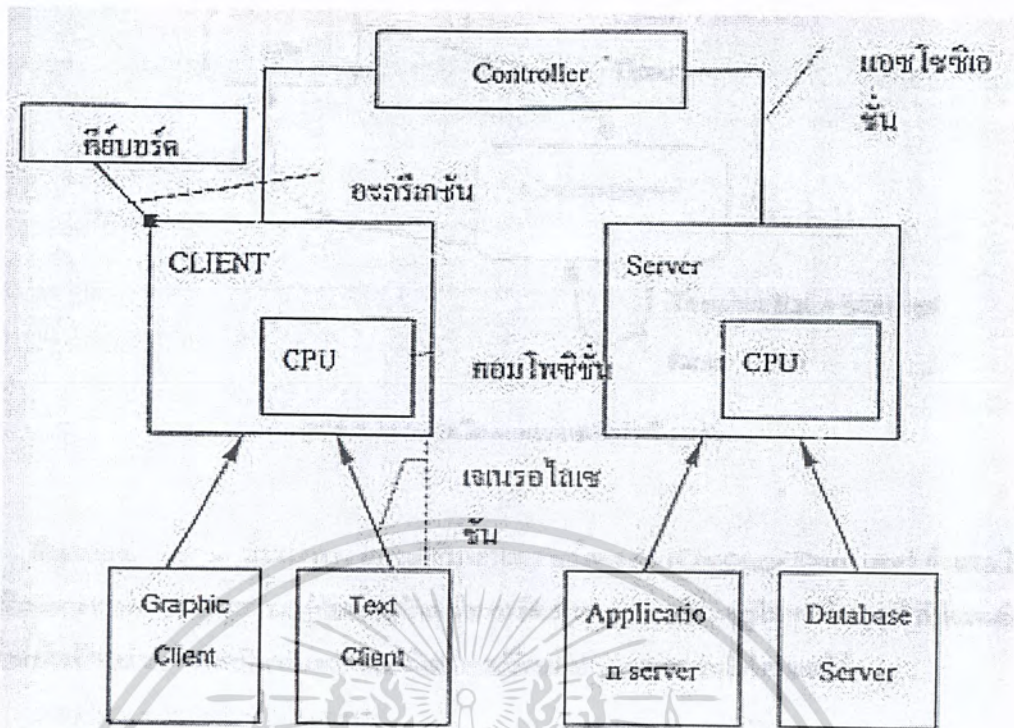
ในระบบออบเจกต์โอเรียนเท็ดแบ่งความสัมพันธ์ระหว่างออบเจกต์ออกเป็น 4 ประเภทดังนี้

1. แอชโซซิเอชัน (Association)
2. อะกรีเกชัน (Aggregation)
3. คอมโพสิชัน (Composition)
4. เจเนอราไลเซชัน (Generalization)

ความสัมพันธ์แอชโซซิเอชันเป็นความสัมพันธ์ระหว่างคลาส เมื่อออบเจกต์ของคลาสมีการส่งเมสเสจถึงกัน ในการสร้างความสัมพันธ์แบบแอชโซซิเอชันสามารถทำได้โดยประกาศอินสแตนซ์ไว้เป็นตัวอ้างอิงไปยังออบเจกต์ที่เป็นผู้รับเมสเสจ และเมื่อออบเจกต์ต้องการส่งเมสเสจก็เรียกฟังก์ชันจากอินสแตนซ์ออบเจกต์ผู้รับเมสเสจ

ความสัมพันธ์แบบอะกรีเกชันและคอมโพสิชันเป็นความสัมพันธ์ระหว่างคลาสที่มีลักษณะคล้ายกัน กล่าวคือการที่มีออบเจกต์ของคลาสหนึ่งเป็นส่วนหนึ่งของอีกออบเจกต์ของคลาสหนึ่งในการสร้างความสัมพันธ์แบบอะกรีเกชัน และคอมโพสิชันสามารถทำได้โดยการประกาศอินสแตนซ์ของออบเจกต์ไว้ภายในคลาสที่โคสร้างเป็นออบเจกต์เจ้าของ โดยออบเจกต์เจ้าของจะเป็ยผู้สร้างและทำลายออบเจกต์ที่อยู่ใน

ความสัมพันธ์แบบเจเนอราไลเซชัน เป็นความสัมพันธ์เมื่อมีออบเจกต์หนึ่งสืบทอดคุณสมบัติมาจากอีกออบเจกต์หนึ่งโดยในความสัมพันธ์เจเนอราไลเซชันนั้น คลาสที่ได้รับการสืบทอดคุณสมบัติมาหรือสืบทอดนั้นได้รับการสร้างเป็นออบเจกต์ที่มีคุณสมบัติและเมธอดที่เหมือนกับออบเจกต์ที่สร้างจากซูเปอร์คลาสและยังมีคุณสมบัติและเมธอดที่เพิ่มเข้ามาด้วย



รูปที่ ก-6 ความสัมพันธ์ระหว่างกลาสในแบบต่างๆ

#### การกำหนดเมธอดของออบเจกต์

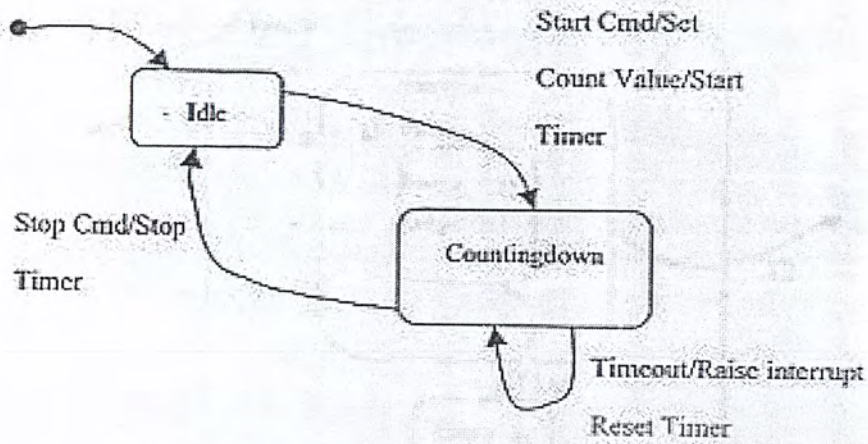
เมธอดเป็นส่วนที่บ่งบอกถึงคุณสมบัติและหน้าที่รับผิดชอบของออบเจกต์ ในการโมเดลโอเปอเรชัน และเมธอดจะใช้สเตตไดอะแกรม ด้วยการใส่สเตตไดอะแกรมจะทำให้เราสามารถตรวจสอบว่าออบเจกต์สามารถทำงานร่วมกันได้อย่างถูกต้องหรือไม่

จากที่นำเสนอไปก่อนหน้านี้เป็นการกำหนดออบเจกต์ คุณสมบัติของออบเจกต์รวมทั้งโครงสร้างของออบเจกต์และความสัมพันธ์ ความสัมพันธ์ระหว่างออบเจกต์จะเกิดขึ้นได้ก็ต่อเมื่อมีการเรียกเมธอดของออบเจกต์

#### การกำหนดสเตตของออบเจกต์

สำหรับการวิเคราะห์และออกแบบระบบแบบออบเจกต์โอเรียนเต็ลเราจะกำหนดให้ 1 คลาสมี 1 สเตตไดอะแกรม ตัวสเตตนั้นแสดงด้วยสี่เหลี่ยมที่มีขอบมน สำหรับการเปลี่ยนสเตตจะแสดงด้วยเส้นตรงที่มีหัวลูกศร โดยเริ่มต้นจากสเตตที่เป็นต้นทางไปยังสเตตที่เป็นปลายทาง สำหรับรายละเอียดการเปลี่ยนสเตตนั้นจะประกอบไปด้วย อีเวนต์ที่ทำให้เกิดการเปลี่ยนสเตตตามด้วย แอคชัน (ฟังก์ชันหรือโอเปอเรชัน) ที่จะต้องทำเมื่อมีการเปลี่ยนสเตต ส่วนเครื่องหมายวงกลมระบายสีดำเป็นจุดเริ่มต้นของสเตตไดอะแกรมดังรูปที่ ก-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-7 สเตทไดอะแกรมของตัวจับเวลา

สัญลักษณ์  $\diamond$  เรียกว่าคอนดิชันนอลคอนเน็คเตอร์(Condition Connector) ซึ่งยอมให้มีการเปลี่ยนสเตทได้หลายสเตทขึ้นอยู่กับเงื่อนไขหรือการ์ด ตัวการ์ดจะเขียนอยู่ในวงเล็บ  $\square$  ถ้ามีอีเวนต์เกิดขึ้นแล้ว การ์ดจะเป็นจริงหรือไม่เป็นจริงจะทำให้เกิดการเปลี่ยนแปลงสเตทตามที่กำหนดไว้

#### สเตทไดอะแกรมพื้นฐานในยูเอ็มแอล

ไวยากรณ์ของสเตทไดอะแกรมในยูเอ็มแอล ได้อธิบายไปแล้วในบางส่วนในรายงานส่วนนี้จะนำเสนอส่วนที่เพิ่มความสามารถต่าง ๆ ที่สำคัญสำหรับสเตทไดอะแกรม

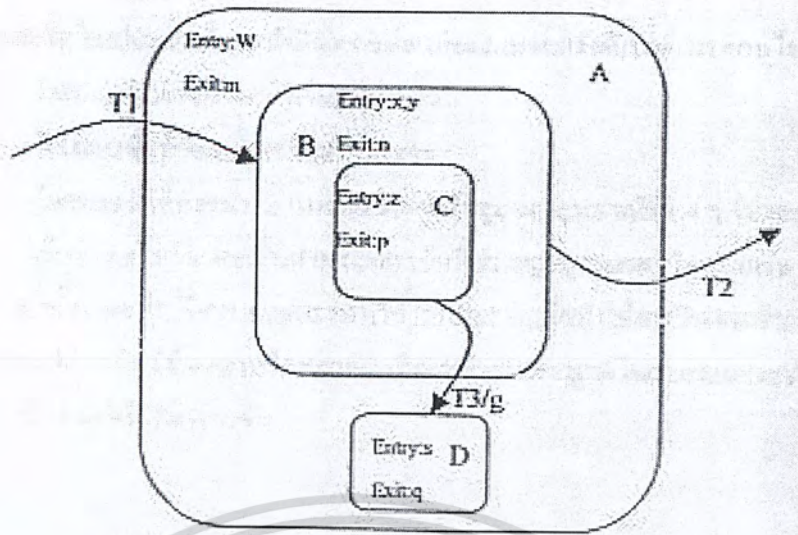
- สามารถทำสเตทซ้อนสเตทได้โดยสเตทภายในเรียกว่า สับสเตท ส่วนสเตทภายนอกเรียก ซุปเปอร์สเตท
- มีการกำหนดสเตทเริ่มต้นให้กับซูปเปอร์สเตทได้ เมื่อมีการเปลี่ยนสเตทมาที่ ซูปเปอร์ สเตทนี้ ก็ให้เริ่มที่สเตทเริ่มต้นนี้
- สามารถใช้ฮิสทอรีคอนเน็คเตอร์ (History Connector) คือ เมื่อกลับเข้ามาซูปเปอร์สเตทนี้อีก กำหนดให้เริ่มสเตทที่ค้างอยู่ก่อนที่จะมีการเปลี่ยนสเตทออกไป ถ้าไม่มีก็ให้เริ่มที่ฮิสทอรีสเตท

สเตทไดอะแกรมสามารถมีแอกชันในตัวสเตทเองหรือในช่วงการเปลี่ยนแปลงสเตทก็ได้ โดยสามารถแบ่งเป็น

- แอกชันที่ทำเมื่อมีการเข้าสู่สเตท หรือ Entry Action (ใช้สัญลักษณ์ “Entry:”)
- แอกชันที่ทำเมื่อมีการออกจากสเตท หรือ Exit Action (ใช้สัญลักษณ์ “Exit:”)
- แอกชันที่ทำเมื่อสเตทนั้นๆแอกทิฟ หรือ Activities (ใช้สัญลักษณ์ “Do:”)

จากที่กล่าวมาสามารถสรุปได้ดังรูปที่ ก-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-8 สเตทไดอะแกรมสำหรับซูปเปอร์สเตทและสับสเตท

สำหรับทรานซิชันอาจจะมีพารามิเตอร์และการ์ดรวมทั้งแอกชัน โดยไวอากรณ์ สรุปได้ดังนี้

ชื่ออีเวนต์(รายการพารามิเตอร์)[การ์ด]/รายการแอกชันสำหรับการเปลี่ยนสเตท ^ รายการอีเวนต์

ซึ่งสรุปได้ดังตารางที่ ก-1

คำ	คำอธิบาย
ชื่ออีเวนต์	เป็นชื่ออีเวนต์ที่ทำให้เกิดการเปลี่ยนสเตท
การ์ด	เป็นเงื่อนไขที่อยู่ในรูปประโยคบูลีน ซึ่งจะต้องมีค่าเป็นจริงจึงจะทำให้เกิดการเปลี่ยนสเตทได้ โดยปกติจะใช้ร่วมกับคอนดิชันอลซูปเปอร์สเตทคอนเนคเตอร์ เพื่อตัดสินใจว่าจะเปลี่ยนไปยังสเตทใด
รายการแอกชัน	เป็นรายการที่ใช้เครื่องหมายลูกน้ำแยกแต่ละส่วน โดยเป็นแอกชันที่จะต้องเกิดขึ้นเมื่อมีการเปลี่ยนสเตท
รายการอีเวนต์	เป็นรายการอีเวนต์ที่ใช้ลูกน้ำชั้นลำดับ โดยจะเป็นอีเวนต์ที่จะต้องเกิดขึ้นเมื่อมีการเปลี่ยนสเตท

ตารางที่ ก-1 ความหมายของพารามิเตอร์ต่าง ๆ ของทรานซิชัน

การกำหนดเมธอดหรือโอเปอเรชัน

การกำหนดเมธอดหรือโอเปอเรชันมีหลักการดังนี้

- กำหนดเมธอดหรือ โอเปอเรชันสำหรับออบเจกต์โดยเป็นอิสระต่อกัน
- พยายามซ่อนรายละเอียดโครงสร้างภายในออบเจกต์โดยผู้ต้องการติดต่อต้องผ่านอินเตอร์เฟซที่กำหนดไว้เท่านั้น
- ซูปเปอร์คลาสควรมีเมธอดหรือ โอเปอเรชันที่สับคลาสส่วนใหญ่ต้องการใช้งาน
- เมธอดหรือ โอเปอเรชันที่ทำหน้าที่จัดการเกี่ยวกับแมสเฮจหรืออีเวนต์ประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โอเปอเรชั่นที่จัดการการรับส่งเมสเสจ
- โอเปอเรชั่นที่ทำหน้าที่อ่านหรือเขียนค่าข้อมูลของคุณสมบัติต่างๆของออบเจกต์
- เมธอดหรือโอเปอเรชั่นสำหรับแอคชั่นที่ปรากฏอยู่บนสแตทโคอะแกรม

จากที่กล่าวมาทั้งหมดเป็นวิธีการหาเมธอดหรือโอเปอเรชั่น เพื่อทำให้การวิเคราะห์ระบบมีความถูกต้องมากที่สุด โดยจะได้ผลลัพธ์เป็นคลาสโคอะแกรมที่ค่อนข้างจะสมบูรณ์ โดยแต่ละคลาสประกอบไปด้วยคุณสมบัติ และเมธอดหรือโอเปอเรชั่น

### 3 การออกแบบ

ในกระบวนการการออกแบบนั้นจะเป็นการปรับเปลี่ยนโครงสร้างและการทำงานของโมเดลที่ได้จากการวิเคราะห์ระบบ โดยสามารถแบ่งออกเป็น 3 ระดับ คือ

- การออกแบบในระดับสถาปัตยกรรม (Architectural Design) เป็นการออกแบบในระดับโครงสร้างของระบบในระดับใหญ่คือ ระบบย่อย (Subsystem) แพคเกจ (Package) และ ทาสก์ (Task)
- การออกแบบในระบบโครงสร้างการทำงาน (Mechanistic Design) เป็นการออกแบบที่เน้นเกี่ยวกับโครงสร้างการทำงานระหว่างคลาสและออบเจกต์ที่มีอยู่ในระบบเพื่อให้บรรลุหน้าที่รับผิดชอบ
- การออกแบบในระดับรายละเอียด (Detail Design) จะกล่าวถึงโครงสร้างข้อมูลและอัลกอริทึมในการทำงานของคลาสต่างๆ

จากที่กล่าวมาทั้งหมดแนวทางการออกแบบระบบเพื่อปรับปรุงสิ่งที่ได้จากการวิเคราะห์ให้ถูกต้องมากที่สุดและสามารถนำกลับมาใช้ใหม่ได้มากที่สุด ซึ่งเป็นจุดประสงค์ที่สำคัญที่สุดในการวิเคราะห์และออกแบบระบบแบบออบเจกต์โอเรียนเท็ด ดังนั้นในปัจจุบันได้มีการเก็บรูปแบบการออกแบบในระบบเก่าเพื่อนำกลับมาประยุกต์ใช้กับระบบใหม่โดยอาจจะมีการปรับเปลี่ยนในบางส่วนบ้างก็ตาม รายละเอียดที่เก็บเกี่ยวกับ ข้อกำหนดของคลาสและออบเจกต์รวมทั้งความสัมพันธ์ระหว่างกันในระบบ ดังนั้นผู้ที่ออกแบบระบบมือใหม่ก็สามารถออกแบบระบบได้อย่างมีประสิทธิภาพ ด้วยการศึกษารูปแบบการออกแบบที่มีผู้เสนอไว้ ไปประยุกต์ใช้ทำให้ไม่ต้องเสียเวลาในการศึกษาและสร้างวิธีในการออกแบบระบบของตนเองขึ้นมา รูปแบบการออกแบบนี้เราเรียกว่า ดีไซน์แพทเทิร์น (Design Pattern)

## บรรณานุกรม

- [1] Richard C. Leinecker (2000) : “COM+ Unleashed”, Sams Publishing
  - [2] Alan Gordon (2000) : “The COM and COM+ Programming Primer”, Prentice Hall PTR
  - [3] Eric Harmon (2000) : “Delphi COM Programming”, Macmillan Technical Publishing
  - [4] Inprise Coperation (1999) : “Borland Delphi’s 5 Developer Guide”, Inprise Coperation
  - [5] Steve Teixeira and Xavier Pacheco (2000) : “Borland Delphi’s 5 Developer Guide”, Sams Publishing
  - [6] Marco Cantú (1999) : “Mastering Delphi 5”, Sybex Inc.
  - [7] Alex Fedorov and Natalia Elmanova, Ph. D. (2000) : “Advance Delphi Developer’s Guid to ADO”, World Ware Publishing Inc.
  - [8] Alan W. Brown (2000) : “Large-Scale, Component – Based Development”, Prentice Hall PTR
  - [9] Guy Eddon and Henry Eddon (1998) : “Prgramming Components With Microsoft Visual Basic 6.0”, Microsoft Press
  - [10] Noel Jerke, George Szabo, David Jung and Don Kiely (1999) : “The Waite Group’s Visual Basic 6 Client/Server How-To”, Techmedia
- เว็บไซต์อ้างอิง
- [11] <http://www.msdn.microsoft.com>