

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนากระดานข่าวและโปรแกรมเสริมสำหรับผู้ใช้หลายคน

IMPLEMENTATION OF ONLINE MULTIUSER WHITEBOARD AND ITS PLUGINS



เลขหมู่.....
เลขทะเบียน..... 62421
วัน,เดือน,ปี 17 ส.ค. 2548

| |
|--------|
| b..... |
| i..... |

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนากระดานขาวสำหรับผู้ใช้หลายคนและโปรแกรมเสริม
IMPLEMENTATION OF MULTIUSER WHITEBOARD AND ITS PLUGINS

โดย
นายธนา ทศนาเสถียรกิจ
นายนนทสิทธิ์ นครชัย



ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

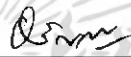
เรื่อง การพัฒนากระดานข่าวสำหรับผู้ใช้งานหลายคนและโปรแกรมเสริม

TITLE IMPLEMENTATION OF MULTUSER WHITEBOARD AND ITS PLUGINS

ผู้จัดทำ

1. นายธนา ทัศนเสถียรกิจ รหัสนักศึกษา 45010324
2. นายนนทสิทธิ์ นครชัย รหัสนักศึกษา 45010367





อาจารย์ที่ปรึกษา

(ดร.อรัญญา วลัยรัชต์)



อาจารย์ที่ปรึกษาร่วม

(ดร.สมศักดิ์ วลัยรัชต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนากระดานข่าวอะโปรแกรมเสริมสำหรับผู้ใช้งานหลายคน

| | |
|-----------------------|----------------------|
| นายธนา ทัศนาศาสตร์กิจ | 45010324 |
| นายนนาสิทธิ์ นครชัย | 45010367 |
| ดร.อริยญา วลัยรัชต์ | อาจารย์ที่ปรึกษา |
| ดร.สมศักดิ์ วลัยรัชต์ | อาจารย์ที่ปรึกษาร่วม |
| ปีการศึกษา 2548 | |

บทคัดย่อ

ในปัจจุบันระบบเครือข่ายอินเทอร์เน็ต มีความสามารถในการส่งข้อมูลด้วยความเร็วที่สูงขึ้นมาก และมีแนวโน้มว่าจะสูงขึ้นเรื่อยๆในอนาคต ทำให้สามารถตอบสนองความต้องการของโลกธุรกิจ ในการสื่อสารแบบสมจริง เช่น Voice over IP , Realtime Conference และอื่นๆ ได้อย่างมีประสิทธิภาพและเชื่อถือได้ จึงทำให้เกิดพัฒนาแอปพลิเคชันจำพวก Voice over IP หรือ Realtime Conference ขึ้นมาเป็นจำนวนมาก แต่งานบางประเภทที่ใช้ แอปพลิเคชันเหล่านั้นผู้ใช้งานยังมีความต้องการ การสื่อสารอีกรูปแบบหนึ่งที่ แสดง ภาพหรือเอกสารที่ตัวเองต้องการจะนำเสนอในระหว่างการสนทนา และสามารถ สร้าง แก้ไข ตกแต่งสิ่งที่ต้องการนำเสนอ ได้พร้อมกันหลายๆ คน ในระหว่างการสนทนา ในลักษณะที่เหมือนการใช้ Multi-user Whiteboard ดังนั้นจึงเป็นที่มาในส่วนของผู้ใช้สำหรับโครงการนี้ สำหรับที่มาในส่วนของผู้พัฒนาแอปพลิเคชันนั้น หากต้องการให้แอปพลิเคชัน ที่พัฒนานั้นมีความสามารถเช่น Multi-user Whiteboard เพิ่มขึ้นมากก็ต้องเสียเวลาในการพัฒนาซอฟต์แวร์ในส่วนนี้ขึ้นมา ซึ่งในโครงการนี้ได้มีการพัฒนา Multi-user Whiteboard Component ซึ่งทำงานบนเทคโนโลยี .NET ของ Microsoft ซึ่งผู้พัฒนาแอปพลิเคชันสามารถนำเอา Multi-user Whiteboard Component ที่พัฒนาขึ้นนี้ไปเป็นส่วนประกอบ ของ แอปพลิเคชันที่ต้องการพัฒนา ได้อย่างง่ายดาย ผ่านทางการเขียนโปรแกรมติดต่อด้วยภาษาที่ .NET Framework สนับสนุน เช่น C++,C#,J#,Visual Basic และอื่นๆ นอกจากนั้น ยังสามารถเรียกใช้งานด้วยภาษา เช่น HTML,JavaScript โดย ผ่านทาง Web Browser ของ Microsoft เช่น Internet Explorer ได้ด้วย

IMPLEMENTATION OF MULTIUSER WHITEBOARD AND ITS PLUGIN

| | |
|---------------------------|------------|
| Mr.Thiana Tassanastainkit | 45010324 |
| Mr.Nontasit Nakornchai | 45010367 |
| Dr.Aranya Walairatch | Advisor |
| Dr.Somsak Walairatch | Co-Advisor |

Academic Year 2004

ABSTRACT

Today Internet has more bandwidth than past. In future Internet have trend to send more data then application such as Voice over IP, realtime conference will reliable and will by used spaciosusly. But some business requirement more these application. The requirement want another type of communication . Requirement that user can show a document and edit or mark a document. User can drawing a picture and send the picture to another person when conference time. These requirements are cause of this project . For another cause of this project in when application developers want to develop application that have ability that abovementioned application developers must spend more time to develop that part of software. In this project Implement a Multi-user Whiteboard Component that others developer can use it to be a component of their software easily. This Component is built on .Net Technology from Microsoft this component can write code to control by many computer language than .Net support such as Visual Basic,C#,J#,C++ . Moreover this component can use in both Window Application and Web Application.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยความช่วยเหลือ และคำปรึกษาจาก ดร.อรัญญา วลัย
รัชต์ และ ดร.สมศักดิ์ วลัยรัชต์ ข้าพเจ้ารู้สึกทราบบ้างซึ่งในความอนุเคราะห์จากท่านอาจารย์ทั้งสอง
ท่าน และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง ทุก ๆ ท่าน ที่ได้ทุ่มเทประสิทธิภาพวิชาความรู้ต่างๆ ให้ข้าพเจ้า ตลอด
ระยะเวลาที่ผ่านมาทั้ง 4 ปี

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยี
พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา
สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ
และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี
คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบอบแต่ผู้มีพระคุณทุกท่าน

นายธนา ทัศนาศติบริก

นายนนทสิทธิ์ นทรชัย

สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อภาษาไทย..... | I |
| บทคัดย่อภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญตาราง..... | VI |
| สารบัญรูป..... | VII |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ความสำคัญและที่มาของโครงการ..... | 1 |
| 1.2 วัตถุประสงค์ของโครงการ..... | 1 |
| 1.3 ขอบเขตของโครงการ..... | 2 |
| 1.4 วิธีการดำเนินการ..... | 2 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ..... | 2 |
| 1.6 ส่วนประกอบของปริญญานิพนธ์..... | 3 |
| บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในโครงการ..... | 4 |
| 2.1 คอมพิวเตอร์..... | 4 |
| 2.1.1 คุณสมบัติของคอมพิวเตอร์..... | 4 |
| 2.1.2 คอมพิวเตอร์โมเดล..... | 5 |
| 2.1.3 การพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์สำหรับการนำกลับมาใช้ใหม่..... | 5 |
| 2.1.4 ปัญหาสำหรับการพัฒนาสำหรับการพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์..... | 6 |
| 2.2 .NET Framework..... | 6 |
| 2.2.1 ส่วนประกอบของ .Net Framework..... | 6 |
| 2.2.1.1 Common Language Runtime (CLR)..... | 7 |
| 2.2.1.2 NET Framework Class Library..... | 7 |
| 2.2.1.3 Programming Languages..... | 7 |
| 2.2.1.4 ASP.Net..... | 7 |
| 2.2.2 ข้อดีของการพัฒนาซอฟต์แวร์บน .Net Framework..... | 8 |
| 2.2.2.1 Executable code สามารถเรียกใช้เกือบทุก OS และเป็น..... | 8 |

| | | |
|---------|--|--|
| | อิสระต่อ Hardware | |
| | 2.2.2.2 Library เป็นชุดเดียวกัน ไม่ว่าจะ เป็นภาษาคอมพิวเตอร์ภาษา.....8 | |
| | ได้ก็ตาม และมีความสมบูรณ์มาก | |
| | 2.2.2.3 มีการควบคุมสิ่งแวดล้อมในการ Run โปรแกรมเป็นอย่างดี.....8 | |
| | 2.2.2.4 สามารถใช้ภาษาใหม่คือ C# ที่รวมเอาข้อดีของ C, C++, Java,.....9 | |
| | Delphi และ Ruby เข้าด้วยกัน | |
| | 2.3 .Net Assembly.....9 | |
| | 2.3.1 ประวัติ ความเป็นมาและความหมายของ .Net Assembly.....9 | |
| | 2.3.2 โครงสร้างของ .Net Assembly13 | |
| | 2.4 IP Multicasting.....14 | |
| บทที่ 3 | การออกแบบและพัฒนา.....17 | |
| | 3.1. Multi-user Whiteboard Component.....18 | |
| | 3.2. Multi-user Whiteboard Plug-ins.....22 | |
| | 3.2.1 Web browser Plug-in.....22 | |
| | 3.2.2 Questionnaire Plug-in.....23 | |
| | 3.3 Multi-user Whiteboard Application.....25 | |
| บทที่ 4 | การทดลองและผลการทดลอง.....27 | |
| | 4.1 การทดลองการนำ Multi-user Whiteboard Component ไปใช้งาน.....27 | |
| | 4.1.1 การนำ Multi-user Whiteboard Component ไปใช้ในส่วนของวินโดว | |
| | แอปพลิเคชัน | |
| | 4.1.2 การนำ Multi-user Whiteboard Component ไปใช้ในส่วนของเว็บ | |
| | แอปพลิเคชัน | |
| | 4.2 ทดสอบความสามารถต่างๆ ของ Multi-user Whiteboard Plug-in.....30 | |
| | 4.2.1 การทดสอบการทำงานของ Web browser Plug-in.....30 | |
| | 4.2.2 การทดสอบการทำงานของ Questionnaire Plug-in.....31 | |
| | 4.3 ทดสอบความการทำงาน ของ Multi-user Whiteboard Application.....37 | |
| บทที่ 5 | บทวิจารณ์และสรุป.....38 | |
| | 5.1 บทสรุป.....38 | |
| | 5.2 วิจารณ์สิ่งที่ได้จากโครงงาน.....38 | |
| | 5.3 ปัญหาอุปสรรคและแนวทางแก้ไข.....38 | |
| | 5.4 แนวทางการพัฒนาต่อ.....39 | |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ตาราง)

ตารางที่

2.1 ช่วง IP สำหรับ Multicast

หน้า

15



สารบัญ(รูป)

| รูปที่ | หน้า |
|---|------|
| 2.1 แสดงส่วนประกอบของคอมโพเนนต์..... | 5 |
| 2.2 แสดง ส่วนประกอบของ .NET Framework | 6 |
| 2.3 ส่วนประกอบของ Assembly | 13 |
| 2.4 แสดงการส่งข้อมูลแบบ multicast | 15 |
| 3.1 การออกแบบและพัฒนาโครงการนี้ได้แบ่งออก 3 ส่วน คือ Multi-user | 17 |
| Whiteboard Component, Multi-user Whiteboard Plug-ins, Multi-user Whiteboard Application | |
| 3.2 แสดงอินเตอร์เฟซมาตรฐานที่ Multi-user Whiteboard Plug-in ต้อง อิมพลิเมนต์..... | 18 |
| 3.3 แสดง class diagram ที่ generate จาก Microsoft Visual Studio 2005..... | 19 |
| 3.4 แสดงการ join multicast group เพื่อเริ่มต้นการรับส่งข้อมูล..... | 20 |
| 3.5 แสดงรูปแบบการส่งข้อมูลของ Multi-user Whiteboard Component..... | 20 |
| 3.6 แสดงข้อมูลของ INetwork อินเตอร์เฟซ..... | 21 |
| 3.7 แสดง class web browser plugin..... | 23 |
| 3.8 แสดง class client questionnaire plugin.. | |
| 3.9 แสดง class server questionnaire plugin..... | 24 |
| 3.10 แสดง Multiuser Whiteboard Windows Application..... | 25 |
| 3.11 แสดง Multiuser Whiteboard Web Application..... | 26 |
| 4.1 แสดงการเพิ่มคอมโพเนนต์ใน Toolbox..... | 27 |
| 4.2 แสดงการ browse หา .dll File..... | 28 |
| 4.3 แสดงการเลือกคอมโพเนนต์จาก Toolbox..... | 28 |
| 4.4 แสดงการเลือกคอมโพเนนต์จาก Toolbox ไปวางบนชิ้นงานที่กำลังพัฒนา..... | 29 |
| 4.5 แสดงการทำงานของ Multiuser Whiteboard Component ร่วมกับ Web Browser Plugin..... | 30 |
| 4.6 User Interface ของ Admin Plug-in..... | 31 |
| 4.7 แสดงข้อมูลของระบบสมาชิก..... | 32 |
| 4.8 แสดง User Interface ของการเลือกประเภทของคำถาม..... | 32 |
| 4.9 แสดง user interface ของข้อมูลแบบถูกผิด..... | 33 |
| 4.10 แสดง user interface ของข้อมูลแบบ multiple choice..... | 34 |
| 4.11 แสดง user interface ของ client Plug in..... | 34 |
| 4.12 แสดงการตอบรับการเข้าใช้งานของ client..... | 34 |

| | |
|--|----|
| 4.13 แสดงความผิดพลาดที่เกิดขึ้นจากการที่ client ป้อน username หรือ password ผิด..... | 35 |
| 4.14 แสดงตัวอย่างของข้อสอบที่ client ได้รับจาก Admin..... | 35 |
| 4.15 แสดงการรายงานผลคะแนนให้ client ทราบ..... | 35 |
| 4.16 แสดงกราฟคะแนนของสมาชิกแต่ละคน..... | 36 |
| 4.17 แสดงรายงานของผลคะแนนของสมาชิกแต่ละคน..... | 36 |
| 4.18 แสดงการทำงานของ Multi-user Whiteboard Application..... | 37 |



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

ในปัจจุบันระบบเครือข่ายอินเทอร์เน็ต มีความสามารถในการส่งข้อมูลด้วยความเร็วที่สูงขึ้นมาก และมีแนวโน้มว่าจะสูงขึ้นเรื่อยๆ ในอนาคต ทำให้สามารถตอบสนองความต้องการของโลกธุรกิจ ในการสื่อสารแบบสมจริง เช่น Voice over IP , Realtime Conference และอื่นๆ ได้อย่างมีประสิทธิภาพและเชื่อถือได้ จึงทำให้เกิดพัฒนาแอปพลิเคชันจำพวก Voice over IP หรือ Realtime Conference ขึ้นมาเป็นจำนวนมาก แต่งานบางประเภทที่ใช้ แอปพลิเคชันเหล่านั้นผู้ใช้มักยังมีต้องการ การสื่อสารอีกรูปแบบหนึ่งที่ แสดง ภาพหรือเอกสารที่ตัวเองต้องการจะนำเสนอ ในระหว่างการสนทนา และสามารถ สร้าง แก้ไข ตกแต่งเอกสารที่ต้องการนำเสนอ ได้พร้อมกันหลายๆ คน ในระหว่างการสนทนา ในลักษณะที่เหมือนการใช้ Multi-user Whiteboard ดังนั้นจึงเป็นที่มาในส่วนของผู้ใช้สำหรับโครงการนี้ สำหรับที่มาในส่วนของผู้พัฒนาแอปพลิเคชันนั้น หากต้องการให้ แอปพลิเคชันที่พัฒนานั้นมีความสามารถเช่น Multi-user Whiteboard เพิ่มขึ้นมาก ก็ต้องเสียเวลาในการพัฒนาซอฟต์แวร์ในส่วนนี้ขึ้นมา ซึ่งในโครงการนี้ได้มีการพัฒนา Multi-user Whiteboard Component ซึ่งทำงานบนเทคโนโลยี .NET ของ Microsoft ซึ่งผู้พัฒนาแอปพลิเคชันสามารถนำเอา Multiuser Whiteboard Component ที่พัฒนาขึ้นนี้ไปเป็นส่วนประกอบ ของ แอปพลิเคชันที่ต้องการพัฒนา ได้อย่างง่ายดาย ผ่านทางการเขียนโปรแกรมติดต่อกับภาษาที่ .NET Framework สนับสนุน เช่น C++, C#, J#, Visual Basic และอื่นๆ นอกจากนั้น ยังสามารถเรียกใช้งานด้วยภาษา HTML, JavaScript โดย ผ่านทาง Web Browser ของ Microsoft คือ Internet Explorer ยิ่งไปกว่านั้น Multi-user Whiteboard Component ยังรองรับให้ผู้พัฒนาแอปพลิเคชันที่ใช้ Multi-user Whiteboard Component เป็นส่วนประกอบสามารถ พัฒนาโปรแกรมเสริม (Plug-in) เพื่อเพิ่มความสามารถให้ตรงกับความต้องการได้อีกด้วย

1.2 วัตถุประสงค์ของโครงการ

โครงการนี้มุ่งหวังเพื่อศึกษาการพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์ ซึ่งเป็นการพัฒนาซอฟต์แวร์ที่มีความยืดหยุ่นสูงสามารถรองรับการเพิ่มความสามารถได้อย่างดี มีความสามารถในการนำกลับมาใช้ใหม่ได้เป็นอย่างดี มีความน่าเชื่อถือสูง นำไปใช้ได้สะดวก และศึกษาเทคนิคและวิธีการพัฒนาซอฟต์แวร์ บน Microsoft .Net Framework แพลตฟอร์ม เพื่อให้เกิดทักษะและความชำนาญในการประกอบอาชีพในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงการ

ในโครงการนี้จะทำการศึกษาแนวคิดและวิธีการในการพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์ โดยใช้ .Net เทคโนโลยี ซึ่งผลงานที่ได้จะมี 3 ส่วนดังนี้

1. Multi-user Whiteboard Component
2. Multi-user Whiteboard Plug-in
3. Multi-user Whiteboard Application

Multi-user Whiteboard Component เป็นซอฟต์แวร์คอมพิวเตอร์ ที่ผู้ใช้หลายๆ คนงานสามารถนำเสนอข้อมูลภาพ, วาดภาพและแสดงข้อความต่างๆ ได้พร้อมๆ กัน และสามารถพัฒนา Plug-in มาเสริมความสามารถอย่างไม่จำกัด

Multi-user Whiteboard Plug-in เป็นซอฟต์แวร์คอมพิวเตอร์ ที่ต้องอิมพลิเมนต์ตามซอฟต์แวร์ อินเทอร์เน็ตเฟรมเวิร์กที่กำหนดโดย Multi-user Whiteboard Component

Multi-user Whiteboard Application เป็นแอปพลิเคชันที่บรรจุ Multi-user Whiteboard Component และใช้ ทดสอบ Multi-user Whiteboard Component

1.4 วิธีการดำเนินการ

1. ศึกษาเกี่ยวกับกระบวนการพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์
2. ศึกษาเกี่ยวกับเทคโนโลยี Microsoft .Net
3. ศึกษาเกี่ยวกับการเขียนโปรแกรมส่งข้อมูลผ่านเครือข่าย
4. วิเคราะห์และออกแบบระบบ
5. พัฒนา Multi-user Whiteboard Component
6. พัฒนา Multi-user Whiteboard Plug-ins
7. สร้าง Multi-user Whiteboard Application สำหรับทดสอบ Multi-user Whiteboard Component ที่พัฒนาขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์
2. ได้รับความรู้ความเข้าใจเกี่ยวกับการพัฒนาซอฟต์แวร์บน .Net Framework
3. ได้รับความรู้ความเข้าใจเกี่ยวกับการเขียนโปรแกรมผ่านเครือข่าย
4. ซอฟต์แวร์ Multi-user Whiteboard Component ที่นำไปใช้ได้สะดวก และสามารถเพิ่มความสามารรถได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ส่วนประกอบของปฏิญญานิพนธ์

ปฏิญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ซึ่งประกอบด้วยทฤษฎีเกี่ยวกับ คอมพิวเตอร์, .Net Framework, .Net Assembly และการส่งข้อมูลผ่านเครือข่าย

บทที่ 3 กล่าวถึงชิ้นงานของโครงการนี้ รายละเอียดของส่วนที่ได้ออกแบบพัฒนาขึ้นทั้ง 3 ส่วน ในโครงการนี้ ได้แก่ Multi-user Whiteboard Component, Multi-user Whiteboard Plugins, Multi-user Whiteboard Application

บทที่ 4 กล่าวถึงการทดลองและผลการทดลอง

บทที่ 5 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้รับจากโครงการ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้ในโครงการ

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานที่สำคัญ ในส่วนที่เกี่ยวข้องกับ โครงการนี้ ได้แก่ ทฤษฎีเกี่ยวกับคอมโพเนนต์ , ทฤษฎีเกี่ยวกับ .Net Framework , ทฤษฎีเกี่ยวกับ .Net Assembly และ ทฤษฎีการส่งข้อมูลผ่านเครือข่ายแบบ Multicast

2.1 คอมโพเนนต์

มีความตกลงทั่วไปในสังคมของผู้พัฒนาซอฟต์แวร์ โดยให้ความหมายของคอมโพเนนต์ คือ คอมโพเนนต์คือหน่วยอิสระที่สามารถประกอบขึ้นจากคอมโพเนนต์อื่นๆ เพื่อที่จะสร้างระบบซอฟต์แวร์ แต่ก็ต้องมีบุคคลอื่นๆนำเสนอนิยามอื่นๆ ของคอมโพเนนต์

Council and Heineman ให้นิยามความหมายของ คอมโพเนนต์ ดังนี้

“A software element that conforms to a component model and can be independently deployed and composed without modification according to a composition standard.”

Ian Sommerville ได้ให้ความหมายของคอมโพเนนต์ไว้ดังนี้

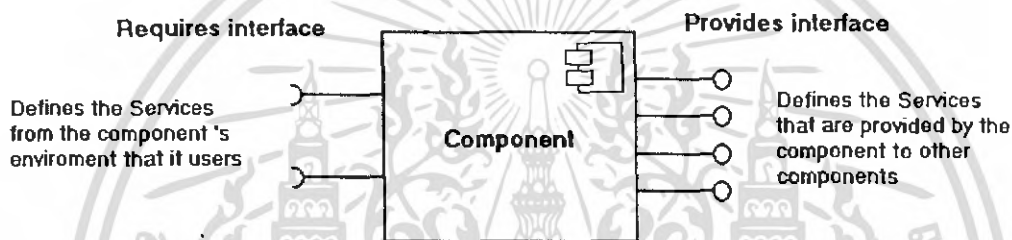
“A component is a software unit whose functionality and dependencies are completely defined by a set of public interfaces. Component can be combined with other components without reference to their implementation and can be deployed as an executable unit.”

2.1.1 คุณสมบัติของคอมโพเนนต์

คุณสมบัติโดยทั่วไปที่สำคัญ ของคอมโพเนนต์มี 5 ประการ ดังนี้

1. Standardised หมายความว่าคอมโพเนนต์ที่นำมาใช้ในกระบวนการพัฒนาซอฟต์แวร์จะต้อง ถูกต้อง ตามข้อกำหนดของคอมโพเนนต์โมเดล โดยคอมโพเนนต์โมเดลนี้ จะกำหนด ข้อกำหนดต่างๆ เช่น คอมโพเนนต์อินเตอร์เฟซ,คอมโพเนนต์เมตาเดต้า,คอมโพเนนต์คอมโพสิชัน,คอมโพเนนต์คอกิวเมนเดชั่น ,คอมโพเนนต์ดีพลอยเมนต์
2. Independent หมายความว่าคอมโพเนนต์ควรเป็นอิสระ และเป็นไปได้ที่จะประกอบและดีพลอย มันโดยปราศจากการใช้คอมโพเนนต์อื่นที่ระบุเฉพาะเจาะจง ในสถานะการที่คอมโพเนนต์ต้องการบริการจากคอมโพเนนต์อื่น จะต้องกำหนด “requires interface” อย่างชัดเจน

3. Composable สำหรับคอมโพเนนต์ที่ composable การติดต่อกับภายนอกทั้งหมดจะต้องผ่าน “publicly defined interface” ซึ่งอินเตอร์เฟซเหล่านั้นจะภายนอกสามารถเข้าถึงข้อมูลภายในของคอมโพเนนต์เช่น method และ attribute ต่างๆ ได้
4. Deployable คอมโพเนนต์เป็นไบนารีสามารถนำไปใช้งานในระบบนั้นได้โดยไม่ต้องคอมไพล์ ระบบนั้นและ คอมโพเนนต์นั้นใหม่
5. Documented คอมโพเนนต์ควรจะต้องมีเอกสารอธิบายเป็นอย่างดีเนื่องจากคอมโพเนนต์เป็น blackbox ซึ่งผู้ใช้จะไม่สามารถรู้รายละเอียดภายในได้ โดยสิ่งที่สำคัญที่เอกสารจะต้องอธิบายคือข้อกำหนดของทุกๆ public interface ทั้งหมด ยิ่งไปกว่านั้น ผู้ใช้ที่จะตัดสินใจว่าจะเลือกใช้คอมโพเนนต์นี้หรือไม่สามารถตัดสินใจได้ว่าคอมโพเนนต์นี้ตรงกับความต้องการหรือไม่



รูปที่ 2.1 แสดงส่วนประกอบของคอมโพเนนต์

2.1.2 คอมโพเนนต์โมเดล

คอมโพเนนต์โมเดลเป็นการนิยามมาตรฐานสำหรับการพัฒนาคอมโพเนนต์, คอกคิวมเตชัน, ดีพลอยเมนต์ มาตรฐานเหล่านี้เป็นมาตรฐานสำหรับผู้พัฒนาคอมโพเนนต์ เพื่อจะได้แน่ใจได้ว่าคอมโพเนนต์ทั้งหลายสามารถทำงานร่วมกันได้ ยิ่งไปกว่านั้นมาตรฐานเหล่านี้ยังได้ให้โครงสร้างพื้นฐานในการ เอกซิคิวชัน และได้ให้ มิดเคิลแวร์ ที่สนับสนุน โอเปอร์เรชันต่างๆของ คอมโพเนนต์

2.1.3 การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์สำหรับการนำกลับมาใช้ใหม่

ข้อดีที่สำคัญสำหรับการพัฒนาที่ดีที่สุดสำหรับการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ คือความสามารถในการนำกลับมาใช้ใหม่(Reuse) โดยปกติแล้วคอมโพเนนต์มีความสามารถในการนำกลับมาใช้ใหม่ได้ในตัวเองอยู่แล้ว แต่ก็มีหลักการที่สามารถเพิ่มความสามารถในการนำกลับมาใช้ใหม่ให้กับคอมโพเนนต์ ดังนี้

1. ไม่ออกแบบเมทอด ให้เจาะจงสำหรับแอปพลิเคชันใดแอปพลิเคชันหนึ่ง
2. ต้องมีเมทอดให้ครบสมบูรณ์ ตามฟังก์ชันการทำงานของ คอมโพเนนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 เปลี่ยนชื่อต่างๆให้เหมาะสมและ เจอเนอรัลมากขึ้น

4. เพิ่ม “configuration interface” ทำให้คอมโพเนนต์ สามารถปรับเปลี่ยนได้ ตามสถานการณ์ เมื่อนำคอมโพเนนต์ไปใช้งาน

2.1.4 ปัญหาสำหรับการพัฒนาสำหรับการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์

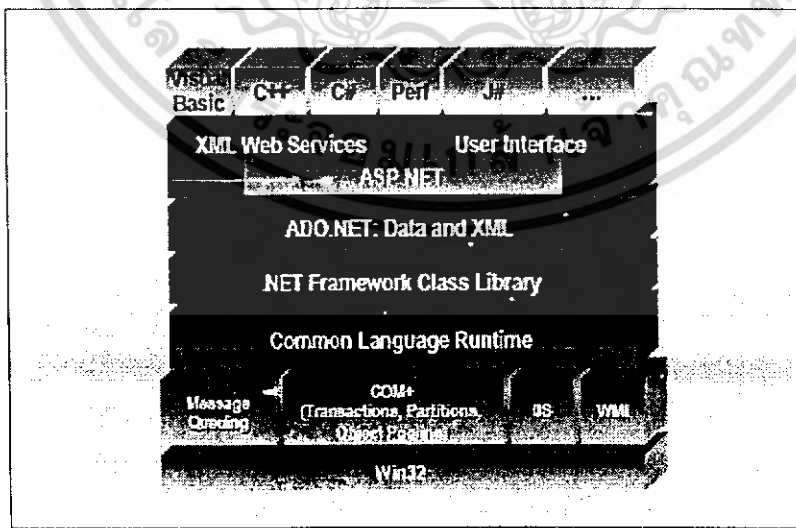
เนื่องจากคอมโพเนนต์เป็นเสมือน blackbox ผู้พัฒนาที่นำคอมโพเนนต์ไปใช้งานนั้น ผู้ใช้ จะไม่สามารถดูรายละเอียดภายในได้เลยว่าคอมโพเนนต์ทำงานอะไรบาง ดังนั้น จึงมีคำถามว่าผู้ใช้ จะรู้ได้อย่างไรว่าคอมโพเนนต์ที่นำมาใช้นั้นมีความน่าเชื่อถือมากน้อยแค่ไหน ซึ่งส่งผลกระทบต่อซีเคียวริตี้คอมโพเนนต์ ที่ไม่น่าเชื่อถืออาจนำมาซึ่งปัญหาต่างๆ เช่น โค้ดที่ผิดพลาดภายในคอมโพเนนต์ อาจทำให้ระบบล่มและ โค้ดที่ไม่หวังทำให้ ซีเคียวริตี้ในระบบมีปัญหา เช่น สามารถเกิด Trojan horse ได้

2.2 .NET Framework

.NET Framework คือ โครงสร้างพื้นฐานในการพัฒนาโปรแกรมคอมพิวเตอร์ที่ ไมโครซอฟต์ พยายามรวมระบบปฏิบัติการ และอินเทอร์เน็ตให้เป็น แพลตฟอร์มเดียวกัน ถูก ออกแบบมา เพื่อให้อำนวยความสะดวกในการพัฒนาโปรแกรมสมัยใหม่ ที่ใช้งานในระบบ เครือข่าย

2.2.1 ส่วนประกอบของ .Net Framework

.Net Framework มีส่วนประกอบ หลัก ๆ ดังนี้



รูปที่ 2.2 แสดง ส่วนประกอบของ .NET Framework

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1 Common Language Runtime (CLR)

เป็นส่วนพื้นฐานที่ติดต่อกับระบบปฏิบัติการ Windows ทำหน้าที่เป็น run-time environment ให้กับโปรแกรมที่เขียนขึ้นสำหรับใช้บน .Net CLR มีส่วนของ compiler ทั้งที่เป็นแบบปกติ (compile ก่อนที่จะนำโปรแกรมไปใช้) และแบบ Just-In-Time (compile เมื่อจะใช้โปรแกรมนั้น ๆ) มีส่วนของ Memory Management ที่เอาไว้สำหรับจัดสรรหน่วยความจำของเครื่องให้กับ โปรแกรม รวมไปถึงการทำ Garbage Collection (การเรียกคืนหน่วยความจำที่ไม่ได้ใช้อีกต่อไป) ส่วนของ Common Type Systems (CTS) ทำให้ภาษาต่าง ๆ ที่เขียนขึ้นบน .Net สามารถทำงานร่วมกันได้ เพราะขนาด และรูปแบบของข้อมูลที่เก็บไว้นั้นเป็นรูปแบบเดียวกัน

2.2.1.2 NET Framework Class Library

เป็น class library พื้นฐาน ที่โปรแกรมต่าง ๆ ไม่ว่าจะเขียนด้วยภาษาใดบน .Net ก็สามารถใช้ class library นี้ร่วมกันได้ เช่น การติดต่อบริการฐานข้อมูล (ADO.Net), การติดต่อกับ file system ของ server (IO), ฯลฯ

2.2.1.3 Programming Languages

เป็นเซตของ ภาษาคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อการเขียนโปรแกรมบน .Net Framework ไมโครซอฟท์นั้นเน้นไปที่ 3-4 ภาษาหลัก ๆ ได้แก่ VB.Net ซึ่งเป็นตัวที่พัฒนาต่อมาจาก VB, C# ซึ่งเป็นภาษาใหม่ที่มี syntax ใกล้เคียงกับ Java และ C++, Visual C++, และ JScript.Net ส่วนภาษาอื่น ๆ นั้น มีบริษัท หรือหน่วยงานอื่น ๆ เป็นผู้พัฒนาขึ้น ซึ่งคาดว่าจะมีเป็นสิบ ๆ ภาษา สำหรับ .Net Framework นั้น ไม่ว่าจะเขียนโปรแกรมด้วยภาษาใดก็ตาม Compiler ใน CLR ก็จะ compile โปรแกรมนั้นให้อยู่ในรูปของ Intermediate Language (IL) ซึ่งจะถูกนำไปแปลเป็นภาษาเครื่อง (Native Code) อีกทีเมื่อตอนที่นำไปใช้

2.2.1.4 ASP.Net

เป็น server side script ที่พัฒนาต่อมาจาก ASP ตัวเก่า เพื่อให้โปรแกรมเมอร์สามารถพัฒนา web application ให้ใช้ .Net ได้สะดวกขึ้น ASP.Net นี้ถึงแม้จะอ้างอิงมาจาก ASP ตัวเก่า แต่ก็มี syntax หลายส่วนที่เปลี่ยนแปลงไป

2.2.2 ข้อดีของการพัฒนาซอฟต์แวร์บน .Net Framework

การพัฒนาซอฟต์แวร์บน .Net Framework มีข้อดีต่างๆดังนี้

2.2.2.1 Executable code สามารถเรียกใช้เกือบทุก OS และเป็นอิสระต่อ Hardware

Microsoft ทำได้อย่างไร เป็นเรื่องที่น่าสนใจ เรื่องนี้ Microsoft ต้อง Install Program ลงไปชุดหนึ่งครับ ที่เรียกว่า Microsoft .Net Framework โปรแกรมตัวนี้จะจำลองตัวเองเป็น OS เล็กๆ ที่หน้าตาเหมือนกับ DOS, และเราสามารถ Run โปรแกรมอะไรก็ได้ที่เขียนขึ้นมาสำหรับ .Net Framework ที่ผมพูดว่า Linux และ Solaris สามารถใช้ Microsoft Office.Net ได้นั้น มันยังไม่จริงนะครับ เพราะวันนี้ทาง Microsoft ยังไม่ Implement .Net Platform บน Linux หรือ Solaris แต่ก็มีความบริษัท ประกาศอาจจะ Implement ส่วนนี้ให้ ขอเพียงแก่ Microsoft ว่าจ้างเท่านั้น ส่วนตัว program ที่ทำงานบน .Net Framework นั้น เป็น .exe ที่สามารถ run ได้โดยตรงจาก OS แต่ code นั้นไม่ใช่เป็น code ของ Intel Pentium หรือ CPU ใดๆ แต่หากเป็น Machine code ที่เป็นภาษาที่ Microsoft คิดค้นขึ้นครับ เมื่อเวลา Run Code นี้จะถูก Compile ให้เป็น Machine Code แท้ๆ ของเครื่องที่ใช้งานอยู่ ด้วยวิธีนี้ เราสามารถ run .exe เดียวกันในบนทุกเครื่อง

2.2.2.2 Library เป็นชุดเดียวกัน ไม่ว่าจะเป็ภาษาคอมพิวเตอรืภาษาใดก็ตาม และมีความสมบูรณ์มาก

ภาษาต่างๆ มี Library ไม่เหมือนกัน ทำให้การเรียนรู้แต่ละภาษานั้นกินเวลามาก และถ้าเขียน module ที่ใช้ภาษาที่มีความสามารถสูง เช่นภาษา C ที่สามารถรองรับ จำนวน parameters ที่ผ่านเข้าหา function ไม่อั้นเช่น printf("%d%d%d", 1, 2, 3); พอใช้ภาษา Pascal ไปอ่าน library ตัวนี้จะไม่รองรับครับ ทำให้การเชื่อมระหว่างภาษาเป็นไปได้โดยยาก Microsoft เลยตัดสินใจที่ยกเลิก Library ของทุกภาษา และออกแบบ Library ตัวใหม่ให้ทุกภาษาใช้ร่วมกัน ต่อไปทุกภาษาจะมี Library ชุดที่สมบูรณ์ที่สุดใช้ ใน Library ชุดนี้มีเกือบทุกสิ่งที่ Programmer อยากรจะมี ADO, Threading ,Networking เป็นต้น

2.2.2.3 มีการควบคุมสิ่งแวดล้อมในการ Run โปรแกรมเป็นอย่างดี

จุดขายของ Java อยู่ที่ โปรแกรมที่เป็น .class นั้น เมื่อนำมาทำงานจะถูกควบคุมไม่ให้ทำงานออกนอกกรอบนอกทางไปวนระบบ ซึ่งคุณสมบัตินี้ถูก Microsoft นำมาเลียนแบบเป็นที่ มีการพิสูจน์ที่มาของ Module โดยใช้การเข้ารหัสมาช่วย ผู้ใช้จะมีสิทธิเลือกเองว่าเชื่อถือ Module นั้นหรือไม่ ใน Platform นี้ยังสามารถกำหนดสิทธิผู้ใช้ตาม OS ได้ มีการจัดการ Garbage Collection ได้ ทำให้ไม่เกิด Memory Leak อีกด้วย

2.2.2.4 สามารถใช้ภาษาใหม่คือ C# ที่รวมเอาข้อดีของ C, C++, Java, Delphi และ Ruby เข้าด้วยกัน

Microsoft เคยเป็นผู้สนับสนุน Java แต่ Microsoft ก็ไม่สามารถยอมรับความซ้ำของ Java ได้ ดังนั้น ใน Visual J++ จึงทำการปรับปรุง Java โดยเพิ่ม keyword ที่ชื่อว่า Delegate ด้วย keyword ตัวนี้ทำให้ Java สามารถเข้าถึง Windows API และ COM components ได้โดยตรง ทำให้เร็วกว่า Java ของ Sun 5 เท่า แต่เสียความ Compatible กับ Java เจ้าอื่น ทำให้ Sun ตัดสินใจฟ้อง Microsoft และ Microsoft ก็แพ้คดี หลังจากนั้น Microsoft ประกาศเลิก สนับสนุน Java Web ของ Microsoft ทั้งหมดตัดเอา Java ทิ้ง เมื่อเวลาผ่านไป Microsoft Visual Basic ก็ได้รับความนิยมสูงขึ้น แต่ก็ยังมีโปรแกรมเมอร์อยู่เป็นจำนวนมากที่ยอมรับความง่ายเกินไปของ Visual Basic ไม่ได้ เมื่อเรือแตกจากภาษา C และ C++ จึงหันไปหา Java ทำให้สังคม Java ก็กึกก้องมาก ทาง Microsoft ก็รู้ถึงปัญหานี้ จึงหาทางออกแบบภาษาใหม่ ให้ออกแนวภาษา C และ C++ ทำให้ programmer ต่างๆ ผู้ใช้ Java อยู่ สามารถเรียนรู้ภาษานี้ได้โดยไม่ยาก เพื่อการนี้ Microsoft ทำการซื้อตัว Anders Hejlsberg ผู้เป็นผู้ออกแบบ Turbo Pascal และ Delphi จาก Borland มาออกแบบภาษาใหม่โดยมีรหัสว่า Cool และในที่สุดได้ชื่อเป็นทางการว่า C#

ภาษานี้เป็นภาษาที่รวมเอาความสามารถๆ จาก C, C++, Java, และ Delphi เข้าด้วยกันใน C# version 1.0 และรวมความสามารถบางอย่างของ ภาษา Ruby เข้ามาอีกใน C# version 2.0 C# มีความคล้าย Java อยู่ถึง 80% มันเป็นภาษาที่ใช้แนวคิดของ OOP อย่างเต็มตัว และใช้ความสามารถของ .Net ได้เกือบ 100%

2.3 .Net Assembly

Assembly เป็นศัพท์เฉพาะที่ Microsoft สร้าง Assembly ตามความหมายของ .NET นั่นก็คือพวก file .dll รุ่นใหม่นั้นเอง ในอดีตจนถึงปัจจุบันการใช้ file .dll นั้นมีปัญหาหลายอย่าง จึงเป็นเหตุผลว่าทำไมต้องมี Assembly

2.3.1 ประวัติ ความเป็นมาและความหมายของ .Net Assembly

ในยุคของ DOS นั้นโปรแกรมยังไม่ใหญ่มาก โปรแกรมส่วนใหญ่ก็จะไม่มี Graphic มีแต่ตัวหนังสือ (ยกเว้นเกมส์) การแสดงตัวอักษรต่างๆ บนหน้าจอหรือการติดต่อกับอุปกรณ์ต่างๆ เช่น Harddisk, floppy disk นั้นบน DOS และ BIOS ได้เตรียมฟังก์ชันการใช้งานให้เราสามารถเรียกใช้งานได้อยู่แล้ว และถึงแม้ว่าเราจะไม่ใช่ฟังก์ชันดังกล่าว เราเขียนติดต่อกับ I/O โดยตรงก็ทำได้โดยง่าย และขนาดของโปรแกรมไม่ใหญ่ ถ้าเรามองมุมมองของการเขียน โปรแกรม โปรแกรมที่เป็น .exe ส่วนมากบน DOS นั้นสร้างมาจากภาษา C ซึ่งแน่นอนครับว่าเกือบทุกโปรแกรมต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียกใช้ฟังก์ชันที่ชื่อว่า printf() และถ้าเรามีโปรแกรม .exe อยู่ 1,000 ตัวนั้นก็หมายความว่าเรา
มี code ของ printf() แปรกอยู่ถึง 1,000 โปรแกรม

มาถึง Windows 16 bits Windows รุ่นเปลี่ยนแปลงโลก นั่นก็คือ Windows 3.1 ซึ่ง
Windows นั้นเป็น OS ที่ใช้ Graphic User Interface ซึ่งเลียนแบบมาจาก Apple McIntosh สิ่ง
เปลี่ยนไปก็คือ โปรแกรมต่างๆ นั้นแสดงผลเป็น Graphic ทั้งหมด เรามามองถึงมุมมองคน
ออกแบบ Windows รุ่นแรกๆ กันหน่อยดีกว่า ในยุคนั้น Harddisk มีขนาดเล็กมาก Memory ก็น้อย
ความเร็วของ CPU ก็จำกัด ทุกสิ่งทุกอย่างดูเหมือนว่ายังไม่พร้อมสำหรับ Windows ครบ ดังนั้น
การออกแบบ Windows ในยุคนั้นจึงถือเอาหลักประหยัด Resource ให้มากที่สุดเท่าที่จะเป็นไปได้
ดังนั้นสิ่งที่ Microsoft นำเอามาใช้ในเวลานั้นก็คือสิ่งที่เรียกว่า Shared Library คือการที่เอา
Library ที่มีการใช้ร่วมกัน (เช่น printf()) เอาไว้ในแฟ้มข้อมูลแยกต่างหาก จะไม่รวมเข้าไปอยู่ใน
.exe ดังนั้นถ้าโปรแกรม 1,000 ตัว มีการเรียกใช้ printf() จะไม่มีโปรแกรมตัวไหนเลยที่เก็บ code
ของ printf() เวลารันโปรแกรม จะมีการดึงเอา code ของ printf() มาใช้ทำให้โปรแกรมแต่ละตัวมี
ขนาดเล็กลงมากจนน่าแปลกใจ โปรแกรมที่พิมพ์คำว่า Hello, World รุ่นบน DOS และบน
Windows นั้นมีขนาดแทบไม่แตกต่างกัน ถ้าไม่มีเรื่อง Shared Library มาช่วย โปรแกรม Hello,
World บน Windows อาจจะมีใหญ่กว่ารุ่นบน DOS เป็นหลายร้อยเท่าครับ

และจุดที่มีการประชาสัมพันธ์กันก่อนข้างจะมากสำหรับ Shared Library ก็คือความคิดที่
สามารถแก้บั๊กได้ เช่นถ้า printf() มีบั๊ก เราไม่จำเป็นต้องหาทั้ง 1,000 โปรแกรมมาใหม่ ในเมื่อ
โปรแกรมแต่ละตัวเรียกใช้ printf() ข้างนอก เราเพียงปรับปรุง Shared Library ของ printf() เท่านั้น
โปรแกรมทุกตัวก็ถือได้ว่ามีการแก้บั๊กแล้วทั้งหมด และ Shared Library ที่กล่าวมา จะอยู่ใน
แฟ้มข้อมูลที่เรารู้จักกันดี นั่นก็คือ .dll นั่นเอง (Dynamic Linked Library)

มายุค Windows 95 .dll ซึ่งเคยเป็นพระเอกนั้น ถูกมองเป็นผู้ร้ายไป จนกระทั่งทุกวันนี้ มี
ผู้คนเรียก .dll กันติดปอกว่า dll Hell เนื่องจาก ปัญหาหลักที่เกิดขึ้นกับ .dll นั่นก็คือมันเป็นสิ่งที่
ใช้ร่วมกันกันหลายโปรแกรม ทำให้เกิดปัญหาดังนี้ครับ

การตั้งชื่อ .dll เป็นได้เนอะครับว่าบริษัทต่างๆ ที่สร้าง .dll ขึ้นมานั้นอาจจะสร้างชื่อเดียวกัน
ดังนั้นเมื่อเวลาเราลงโปรแกรม จะทำให้มีการทับกันเกิดขึ้น

ถึงแม้ว่า .dll จะมาจากบริษัทเดียวกัน แต่ก็อาจจะเป็นคนละรุ่น และบ่อยครั้งครับที่เมื่อเรา
เรา .dll รุ่นใหม่ไปทับ .dll รุ่นเก่า มันจะทำให้โปรแกรมที่ install ไว้ก่อนหน้านี้ ทำงานไม่ได้ ซึ่งทุก
วันก็ยังมีปัญหานี้อยู่

ทาง Microsoft จึงได้ออกแบบ .dll แบบใหม่ขึ้น รู้จักกันในชื่อ COM Component มันเป็น
ความพยายามที่จะแก้ปัญหา dll Hell โดยที่ COM Component นั้นสามารถรองรับ Interface
Inheritance ทำให้ .dll ตัวเดียวกันสามารถมีหลาย Version แต่ใช่ว่ามันจะเป็นยาวิเศษที่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รักษาโครข่ายตัวนี้ได้ 100% มีคำถามเกิดขึ้นครับว่า มีซักกี่บริษัทที่เวลาออก COM Component รุ่นใหม่ จะกลับไปทดสอบดูว่า COM Component รุ่นนี้เข้ากันได้กับ โปรแกรมรุ่นเก่าโดยสมบูรณ์ เชื่อว่าแทบจะไม่มีบริษัทไหนเลขที่ลงทุนลงแรงขนาดนั้น แม้แต่ Microsoft เองก็ตาม ดังนั้นปัญหาการลงโปรแกรมตัวหนึ่ง ทำให้โปรแกรมตัวอื่นที่เคยทำงานได้กลับทำงานไม่ได้มัน ยังคงเกิดขึ้นอยู่เสมอในยุคปัจจุบันครับ

ปัญหา DLL Hell เริ่มเลวร้ายขึ้น เพราะจำนวน โปรแกรมที่มีการพัฒนาขึ้นสำหรับ Windows นั้นมีปริมาณเพิ่มขึ้นแบบทวีคูณ ทำให้ Windows นั้นถือว่าเป็น OS ที่ไม่ Stable เมื่อเทียบกับค่าย UNIX เช่น Linux เป็นต้น แต่ถ้าไล่กันเข้าไปจริงๆ ปัญหาที่เกิดจาก DLL Hell นั้นเอง

Microsoft ไม่นิ่งนอนใจรับอุดหนุนลงทุนลงแรงมากที่สุดเท่าที่เคยมีมา แต่กลับไปขายน้ำดีแบบนี้ ทาง Microsoft จึงประกาศโครงการที่ชื่อว่า Fusion Technology เพื่อแก้ไขปัญหา DLL Hell โดยเฉพาะครับ และ Windows 98SE นั้นถือว่าเป็นรุ่นแรกของ Technology นี้ครับ Microsoft ทำดังนี้ครับ

เมื่อเวลา Install Programming นั้น ถ้าพบว่า .DLL ที่กำลัง Install นั้นชนกับ DLL ที่มีอยู่แล้ว ทาง Windows จะแจ้งโปรแกรมที่กำลัง Install ว่าได้ย้าย .DLL ที่กำลัง Install นั้น ไปอยู่ที่อื่น ทำให้ไม่ต้องทับกัน ซึ่งการย้ายแบบนี้เราเรียกว่า Side-by-Side และ Technology ที่ใช้ในรุ่นนี้มี Code Name ว่า Hydrogen

แต่มันก็แทบไม่ได้ช่วยอะไรครับ เพราะ โปรแกรมที่กำลัง Install นั้นต้องเรียกใช้ API พิเศษ ซึ่งมีเพียงโปรแกรมไม่กี่ตัวครับที่ใช้ประโยชน์จากมัน

Fusion Technology รุ่นต่อมาคือ Windows ME / 2000 ซึ่งมีสิ่งที่เรียกว่า Windows File Protection (WFP) หรือ System File Protection (SFP) ซึ่งทำการป้องกันไม่ให้ System Files ที่สำคัญถูกโปรแกรมใดๆ ทับ ถ้าถูกทับจริงๆ ก็จะมีกรรมวิธีในการ Restore เพื่อรักษาให้ระบบเสถียร แต่นั่นก็สามารถป้องกันได้เฉพาะ System Files ไม่รวมถึง Application ทั่วไป

เมื่อมาถึง .NET ซึ่งเป็นยุคที่กำลังเครื่องของคอมพิวเตอร์และ Resource ต่างๆ เริ่มมีมากขึ้น Harddisk ขนาด 20 GB ที่ใช้ตามบ้านถือว่ามิมขนาดเล็ก ซึ่งถ้าเปรียบเทียบกับก่อนหน้านั้นประมาณ 10 ปี ขนาด Harddisk 20GB นั้นถือว่าเป็นศูนย์คอมพิวเตอร์ขนาดยักษ์ทีเดียว

เมื่อกำลังเครื่องของคอมพิวเตอร์ไม่ใช่เป็นข้อจำกัด ดังนั้นแนวทางของ Fusion Technology นั้นจึงออกไปในแนวทางที่ไม่มีการทับกัน รุ่นต่างๆ ให้ถือว่าเป็นคนละแผ่นข้อมูลกัน เริ่มต้น .NET นั้นทำการออกแบบ .DLL เสีย ซึ่งในรุ่นนี้จะมีสิ่งที่เก็บนอกจากตัว Code แล้วนั่นก็คือสิ่งที่เรียกว่า MetaData ซึ่งเป็นการอธิบายตัว .DLL ว่าภายในนั้นประกอบด้วยอะไรบ้าง เราถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่า MetaData นั้นเป็นข้อมูลที่สำคัญ ที่สามารถนำไปใช้สร้าง IntelliSense ให้กับ Visual Studio ได้ และยิ่งไปกว่านั้น MetaData ยังเก็บ Version ของ .IDL, .IDL, .IDL, .IDL, .IDL รุ่นใหม่มีเรื่องที่ Microsoft เรียกว่า Assembly

Assembly มี 2 ประเภท แบบแรกคือที่เป็น Private มันจะอยู่เฉพาะกับโปรแกรมของเรา จะไม่ไปรบกวนกับโปรแกรมอื่น และอีกแบบคือ Shared ซึ่งมันต้องไปอยู่ใน Directory ที่ใช้ร่วมกันระหว่างโปรแกรม

Assembly ที่เป็นแบบ Shared นั้น จะเก็บอยู่ในส่วนที่เรียกว่า Global Assembly Cache (GAC) ซึ่งโดยปกติแล้วจะอยู่ใน WINNT\Assembly\GAC โดยที่การที่ไม่ให้โปรแกรมทับกันนั้น ทาง Microsoft ใช้หลักการเข้ารหัสแบบ RSA โดยที่กำหนดให้บริษัททุกบริษัทที่จะนำเอา Assembly ไปเก็บไว้ใน GAC ได้นั้นต้องสร้าง Key 2 ตัว นั่นก็คือ Public Key และ Private Key ทาง Microsoft จะนำเอา 8 bytes หลังของ Public key ของบริษัทนั้นมาสร้างเป็นชื่อ Directory เช่น 1.0.2411.0_b77a5c561934e089 ส่วนแรก 1.0.2411.0 คือ Version และส่วนที่สอง b77a5c561934e089 นั้นเป็น 8 bytes สุดท้ายของ Public key ของ Microsoft ด้วยวิธีนี้จะมี การทับกันเกิดขึ้น ชื่อ Directory แบบพิเศษนี้ Microsoft เรียกว่า Strong Name (โปรแกรมที่ใช้ในการจัดการ GAC ชื่อว่า GACUtil.exe มามาให้กับ .NET Framework)

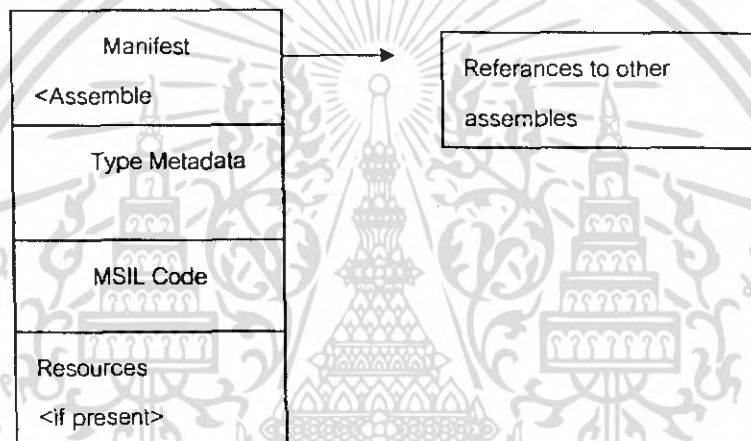
การนำ RSA มาใช้นั้นยังส่งผลถึงเรื่อง Security อีกด้วยครับกล่าวคือมันสามารถพิสูจน์ทราบได้ว่า Assembly ตัวที่กำลังจะลงนั้น มาจากบริษัทนั้นๆ จริงหรือไม่ จะไม่มี Hacker คนไหนสวมรอยแอบเอาโทรจันใส่เข้ามา ในอดีตนั้นเราไม่มีทางรู้ได้เลย แต่ .NET ใช้ Technology ที่ชื่อว่า Digital Signature ทุก Assembly จะมีลายเซ็นติดอยู่ ถ้ามีการเปลี่ยนแปลงใดๆ แม้แต่นิดเดียว มันจะทำให้ลายเซ็นตัวนั้นพัง เราจะรู้ได้เลยว่า Assembly ดังกล่าวมีปัญหา ลายเซ็นตัวนี้ปลอม ได้ยากมาก

เราจะเห็นได้ว่าโปรแกรมในอนาคตที่สร้างจาก .NET นั้นจะ ไม่มีปัญหาเรื่อง DLL Hell อีกต่อไป ทั้งนี้ก็เนื่องจากการใช้งานของ Fusion Technology 2.0 แต่ปัญหา DLL Hell นั้นใช้จะจบสิ้นเลยทีเดียว เรายังมี .DLL version 2 หรือ COM Component ที่ Fusion Technology 2.0 ยังไม่รองรับ เนื่องจากใน COM Component นั้นไม่มี Meta Data ที่ Fusion Technology จะนำเอาไปสร้าง Strong Name ได้ (ส่วน DLL version 1 ที่เป็น 16 bits นั้นคงไม่ต้องกังวลมาก เพราะจำนวนโปรแกรมที่น้อยอยู่แล้ว และมันไม่สมควรที่จะรันโปรแกรมเก่าแบบนั้นแล้ว)

ทาง Microsoft มีโครงการ Fusion Technology 2.5 ที่จะมากับ Windows XP พร้อมกับ คุยไว้ว่า version 2.5 นี้คือจุดสิ้นสุดของ DLL Hell โดยถาวร โดยความสามารถที่จะจัดการกับ COM Component ง่ายๆ ได้ด้วย

2.3.2 โครงสร้างของ .Net Assembly

ส่วนต่างๆของแอสเซมบลีได้ให้ประโยชน์ต่อโปรแกรม .NET ที่จะค้นหาเกี่ยวกับแอสเซมบลีอีกตัวหนึ่ง และแจกแจงการอ้างอิงระหว่างโปรแกรมและคอมโพเนนต์ทั้งหลาย แอสเซมบลีหนึ่งๆจะบรรจุโค้ดที่เอ็กซีคิวต์ได้สำหรับโปรแกรมหรือคลาสไลบรารี ตามด้วยเมตาดาต้า(metadata - คือข้อมูลที่อธิบายข้อมูลอีกตัวหนึ่ง) ซึ่งเปิดโอกาสให้โปรแกรมอื่นๆค้นหาคลาส เมทอด และพรอพเพอร์ตี้ทั้งหลายของออบเจกต์ที่ถูกกำหนดอยู่ในแอสเซมบลีนี้ได้ เมตาดาต้าทำงานได้สองทางคือ เป็นสารบัญที่อธิบายสิ่งต่างๆที่บรรจุอยู่ในแอสเซมบลี และเป็นบรรณานุกรมที่อธิบายการอ้างอิงข้อมูลภายนอกแอสเซมบลี โดย .NET แอสเซมบลีที่เป็นไฟล์เดี่ยวๆจะมีรูปแบบทั่วไปดังต่อไปนี้



รูปที่ 2.3 ส่วนประกอบของ Assembly

แต่ละแอสเซมบลีจะบรรจุ แมนิเฟสท์ (manifest) ซึ่งอธิบายเนื้อหาของแอสเซมบลี (เหมือนกับใบตราส่งสินค้าสำหรับการขนส่งสินค้า) แมนิเฟสท์นี้ยังเรียกได้อีกอย่างหนึ่งว่า เมตาดาต้าของแอสเซมบลี <assembly metadata> เนื่องจากมันอธิบายตัวแอสเซมบลีเองไว้มีโมดูลที่มันบรรจุอยู่บ้าง แอสเซมบลีอื่นใดที่มันอ้างอิง เหล่านี้เป็นต้น เราจะตรวจสอบสิ่งนี้ในรายละเอียดทีหลังของบทเมื่อเราไปดูเนื้อหาของแอสเซมบลีที่เราได้สร้างขึ้นมาแล้ว เทคโนโลยีคอมโพเนนต์ก่อนหน้านี้อาทิ COM เป็นต้น ไม่มีคอนเซปต์ที่สร้างเอาไว้ภายในอย่างเช่นแมนิเฟสท์ นี้เป็นหัวใจของการอธิบายตัวเองที่ถูกรวบรวมอยู่ใน .NET แอสเซมบลี .NET รุ่นใหม่จะใช้แมนิเฟสท์ในแอสเซมบลีของโปรแกรมในขณะเอ็กซีคิวต์โปรแกรมสำหรับรองรับการอ้างอิงถึงแอสเซมบลีอื่นๆ ในระบบ อาทิ ไลบรารี System.Console ที่บรรจุเมธอด WriteLine<> สำหรับการพิมพ์ข้อความ “Hello ,World” ออกมาเป็นต้น แมนิเฟสท์จะต้องตามด้วย เมตาดาต้าของชนิดข้อมูล <type metadata> ซึ่งก็คือคำอธิบายของคลาสหรือพหุวัตต์ เมธอด และอื่นๆ ที่บรรจุใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอสเซมบลีอยู่รวมกับชนิดข้อมูลสำหรับพารามิเตอร์ของมันและค่าส่งคืน แล้วตามท้ายด้วยไบนารีโค้ดจริงๆสำหรับข้อมูลแต่ละชนิดที่ถูกจัดเก็บอยู่ในรูปของโค้ด Microsoft Intermediate Language <MSIL> ที่ไม่ขึ้นอยู่กับเครื่อง สุดท้ายยังมีรีสอร์ซ <resource> ใดๆที่ประกอบกันขึ้นมาเป็นแอสเซมบลีอีกด้วย รีสอร์ซถือเป็นส่วนที่เอ็กซ์คิวต์ไม่ได้ของโปรแกรมของคุณ (ระบุเอาไว้ในไฟล์ .Resources) เช่น อิมเมจ ไอคอน หรือเมสเสจไฟล์ เป็นต้น

ในขณะที่แอสเซมบลีหนึ่งมักจะประกอบขึ้นมาจากไฟล์เพียงไฟล์เดียว มันก็ยังสามารถถูกสร้างขึ้นมาจากไฟล์หลายๆไฟล์ก็ได้ จากมุมมองของ .NET รันไทม์ แอสเซมบลีหลายไฟล์ (multi-file) นั่นก็คือไฟล์ DLL หรือ EXE แต่ละไฟล์ที่เกิดขึ้นมาเพื่อประกอบรวมกันเป็นหลายๆไฟล์ มีเพียงแค่ไฟล์เดียวที่บรรจุแมนิเฟสต์เอาไว้ แมนิเฟสต์จะชี้ไปยังไฟล์อื่นๆที่ทำให้เกิดแอสเซมบลีหลายไฟล์ขึ้นมา ไฟล์ที่บรรจุโค้ดที่เอ็กซ์คิวต์ได้จะถูก เรียกว่าโมดูล ซึ่งไฟล์เหล่านี้บรรจุเมตาดาต้า ของชนิดข้อมูลและ MSIL โค้ด มันยังอาจเป็นรีสอร์ซไฟล์ที่ไม่มีโค้ดที่เอ็กซ์คิวต์ได้บรรจุอยู่ก็ได้

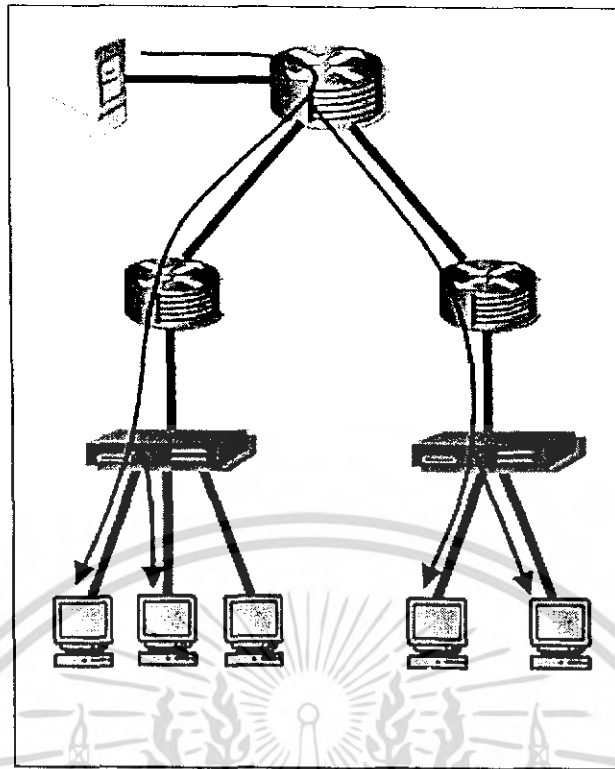
แอสเซมบลีหลายไฟล์นั้นมักจำเป็นต่อแอปพลิเคชันขั้นสูงที่แน่นอนบางตัวเท่านั้น โมดูลหรือรีสอร์ซจะถูกโหลดก็ต่อเมื่อมันเอ็กซ์คิวต์หรือถูกนำไปใช้จริงแล้วเท่านั้น ดังนั้นจึงสามารถประหยัดเวลาการดาวน์โหลด และหน่วยความจำได้เมื่อโมดูลและรีสอร์ซที่แทบไม่ค่อยได้ใช้ถูกจัดเก็บเอาไว้เป็นไฟล์แยกจากกัน ยกตัวอย่างเช่น แอปพลิเคชันหนึ่งที่ส่งมอบกันระหว่างประเทศ อาจมีโมดูลหรือรีสอร์ซที่เขียนขึ้นในภาษาที่แตกต่างกัน จึงควรที่จะแยกสิ่งเหล่านี้ออกจากกัน ดังนั้นจะมีเพียงแค่โมดูลสำหรับภาษาที่ใช้งานจริงเท่านั้นที่จะถูกโหลดเข้าไปในหน่วยความจำ

2.4 IP Multicasting

ในการส่งข้อมูลบน Internet โดยทั่วไปจะเป็นการส่งแบบ 1 ต่อ 1 กล่าวคือ มีผู้ส่ง 1 คน และผู้รับ 1 คน หรือที่เราเรียกว่า Unicast ซึ่งเมื่อจำนวนผู้รับเพิ่มขึ้นมากกว่า 1 จึงจำเป็น ต้องส่งข้อมูลซ้ำตามจำนวนของผู้รับ เช่น เมื่อต้องการส่งภาพ Video Conference ไปให้เพื่อนที่อยู่ต่าง Campus อีก 3 คน จำเป็นต้อง ส่งภาพซ้ำถึง 3 ครั้ง ซึ่งเป็นการ สูญเสียเวลาและทรัพยากร รวมทั้งทำให้ข้อมูลใน Network คับคั่งได้

เทคโนโลยี Multicast เป็นเทคโนโลยีช่วยแก้ปัญหานี้ได้ โดยในการส่งแบบ 1 ต่อ N (N คือจำนวนผู้รับ) ผู้ส่งสามารถส่ง ข้อมูลเพียงครั้งเดียวให้กับผู้รับหลายคนได้ในเวลาเดียวกัน Multicast แบ่งออกเป็น 2 ประเภทคือ IP Multicast ที่ต้องใช้ ความช่วยเหลือจาก Router และ Application Layer Multicast ซึ่งเป็นเทคโนโลยีใหม่ที่ถูกพัฒนาเมื่อเร็ว ๆ นี้ ดังนั้นโดยปกติเวลาที่พูดถึง Multicast จึงหมายถึง IP Multicast

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงการส่งข้อมูลแบบ multicast

โดยในการทำ Multicast นั้นจะทำการกำหนด IP ซึ่งเป็น Multicast Address ต่างๆเป็นช่วงๆ และแต่ละช่วงจะมีการกำหนดวัตถุประสงค์เอาไว้เฉพาะ ดังแสดงในตาราง

| Range | Assignment |
|-----------------------------|--------------------------------|
| 224.0.0.0-224.0.0.255 | Local Network control block |
| 224.0.1.0-224.0.1.255 | Inter-network control block |
| 224.0.2.0-224.0.255.0 | AD-Hoc block |
| 224.1.0.0-224.1.255.255 | ST Multicast groups |
| 224.2.0.0-224.2.255.255 | SDP/SAP block |
| 224.252.0.0-224.255.255.255 | DIS Transient block |
| 225.0.0.0-231.255.255.255 | Reserved |
| 232.0.0.0-232.255.255.255 | Sourcespecific multicast block |
| 233.0.0.0-233.255.255.255 | GLOP block |
| 234.0.0.0-238.255.255.255 | Reserved |
| 239.0.0.0-239.255.255.255 | Administratively scoped block |

ตารางที่ 2.1 ช่วง IP สำหรับ Multicast

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในช่วง IP ต่างๆที่กำหนดขึ้นมาตั้งในตารางข้างต้นนี้ ยังได้กำหนดลงไปอีกด้วยว่าแต่ละหมายเลข IP นั้นใช้อะไร เช่น IP Address 224.0.0.1 และ 224.0.0.2 .ในช่วงของ Local Network control block(224.0.0.0-224.0.0.255) นั้นจะเป็น IP Address ที่สงวนไว้สำหรับให้อุปกรณ์ Routers ใช้สำหรับส่งข้อมูลเกี่ยวกับ Multicast Group ระหว่างกันเท่านั้น เป็นต้น

ในปัจจุบันเทคนิคในการควบคุมการสื่อสารข้อมูลผ่านระบบเน็ตเวิร์คแบบ Multicast มีอยู่ด้วยกัน 2 วิธีคือ

1. Peer to peer เป็นวิธีซึ่งเครื่องทุกเครื่องที่ต้องการสื่อสารแบบ Multicast มีสิทธิ์ในการรับและส่งข้อมูลเท่าๆกัน และโดยปกติแล้วด้วยเทคนิคนี้ จะไม่สามารถป้องกันเครื่องภายนอกทำการลงทะเบียนเพื่อเป็นสมาชิกของ Multicast Group ได้ กล่าวคือเครื่องคอมพิวเตอร์ใดๆในซึ่งเชื่อมต่ออยู่กับ Internet สามารถสมัครเป็นสมาชิกของ Multicast Group ได้ และทุกเครื่องที่เป็นสมาชิกมีสิทธิ์รับข้อมูลที่ Multicast ได้โดยไม่มีการยกเว้นนั่นเอง

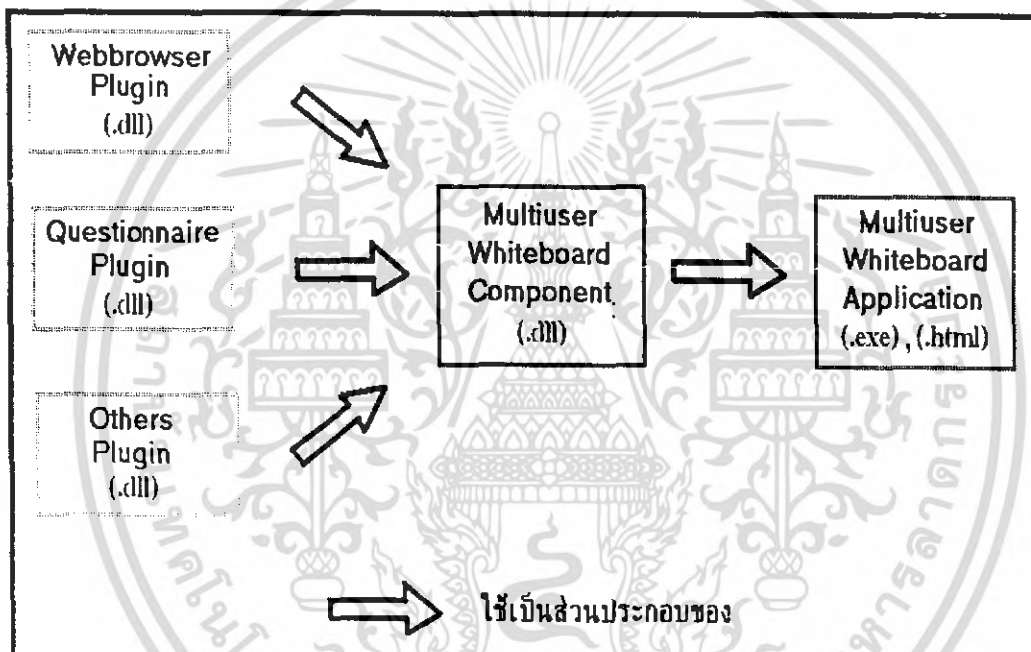
2. Central server เป็นวิธีการซึ่งใช้เครื่อง server เครื่องหนึ่งในการควบคุม Multicast Group โดยเครื่องคอมพิวเตอร์ใดที่ต้องการจะสมัครเป็นสมาชิกของ Multicast Group ต้องร้องขอสถานภาพสมาชิกจากเครื่อง server ที่ควบคุมกลุ่มก่อน หากการร้องขอสำเร็จก็จะได้รับข้อมูล หากไม่สำเร็จก็จะไม่สามารถได้รับ Multicast packet สำหรับเทคนิคการ Multicast แบบ Central server นี้จะไม่สามารถใช้ได้บน IP Network หากแต่ในปัจจุบันจะมีเพียงโครงข่าย ATM เท่านั้นที่สนับสนุนเทคนิคการ Multicast ในรูปแบบนี้

โดยปกติอุปกรณ์ Router จะไม่ส่ง Multicast Packet ข้าม subnet เพราะฉะนั้นเพื่อให้สามารถรับ Multicast Packets ได้ต้องมีกรรมวิธีพิเศษเพิ่มเติมเข้ามา นั่นคือก่อนที่จะทำการรับ Multicast Packets เครื่องที่ต้องการรับ Multicast Packets นั้นจะต้องเข้าเป็นสมาชิกของ Multicast group ก่อน โดยในการสมัครเข้าเป็นสมาชิกของ Multicast group นั้นจะต้องมีการส่ง IGMP<Internet Group Management Protocol> ไปยังอุปกรณ์ Router ที่รับผิดชอบ subnet นั้น <local router> เพื่อทำการสมัครเข้าเป็นสมาชิกของ Multicast group ก่อน โดยในการส่ง IGMP ไปยัง Local Router นั้นก็เพื่อลงทะเบียนเครื่องคอมพิวเตอร์เครื่องนั้นไว้กับ Local Router ว่าต้องการจะรับ Multicast Packets จาก Address ใด รวมทั้งเมื่อเครื่องคอมพิวเตอร์นั้นส่ง Multicast Packets ออกไป อุปกรณ์ Router จะได้ทำการ Router Packets นั้นๆข้าม subnet เมื่อต้องการจะหยุดการรับ Multicast Packets จะต้องส่ง IGMP ไปยัง Local Router เพื่อยกเลิกการลงทะเบียนเข้าใช้บริการ Multicast group

การออกแบบและพัฒนา

การออกแบบและพัฒนาโครงงานนี้ได้แบ่งออก 3 ส่วน คือ

1. Multi-user Whiteboard Component
2. Multi-user Whiteboard Plug-ins
3. Multi-user Whiteboard Application



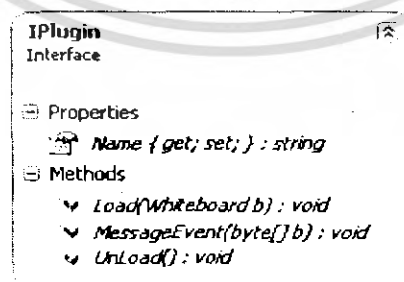
รูปที่ 3.1 การออกแบบและพัฒนาโครงงานนี้ได้แบ่งออก 3 ส่วน คือ Multi-user Whiteboard Component, Multi-user Whiteboard Plug-ins, Multi-user Whiteboard Application

โดย Multi-user Whiteboard plug-ins ต่างๆ ถูกนำไปใช้งานกับ Multi-user Whiteboard Component สามารถนำ Multi-user Whiteboard Component ที่มีส่วนประกอบของ Multi-user Whiteboard Plug-ins ไปใช้ในการพัฒนา แอปพลิเคชันต่างๆ ตามความเหมาะสม

3.1. Multi-user Whiteboard Component

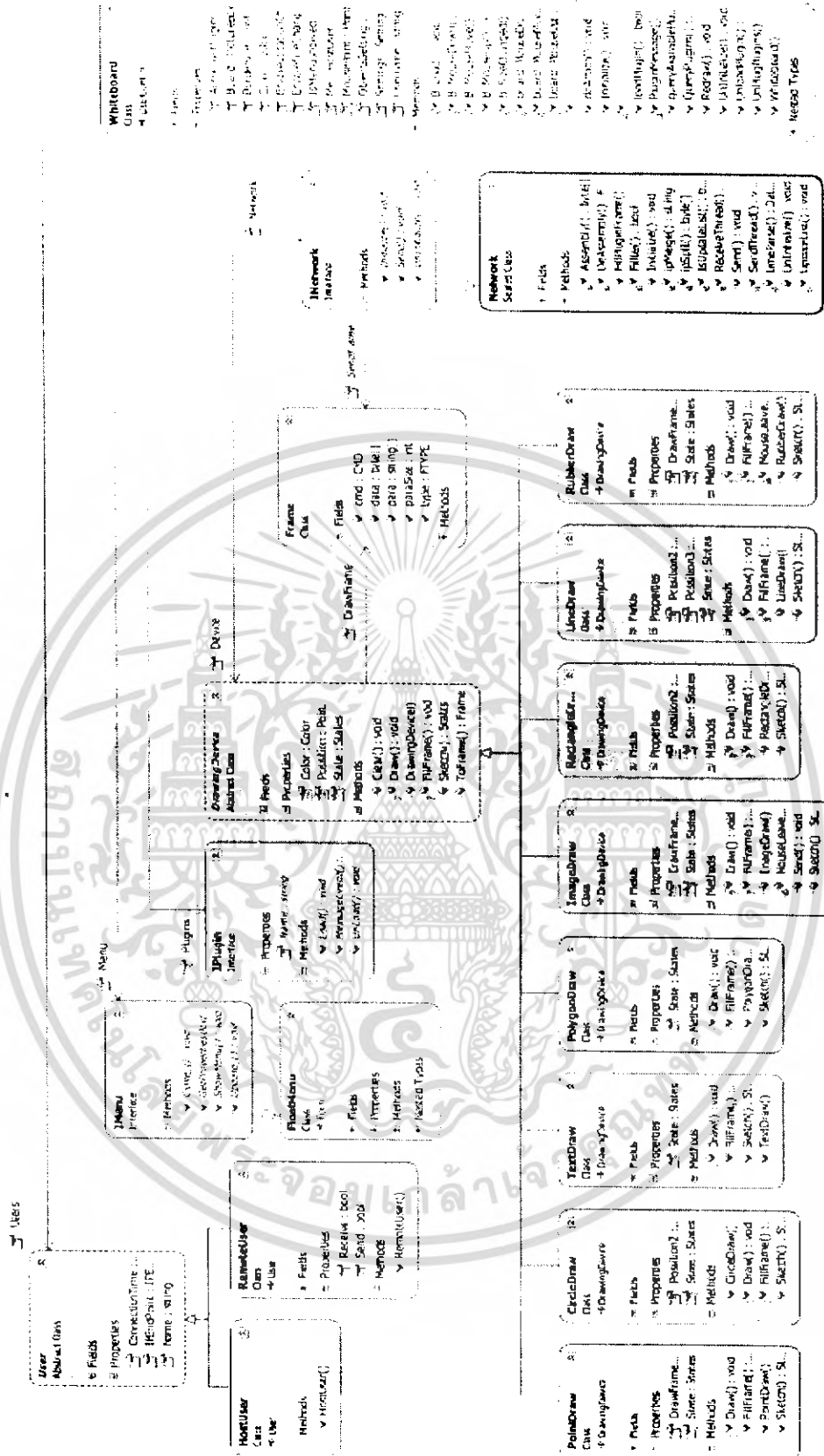
Multi-user Whiteboard Component นี้เป็นซอฟต์แวร์คอมพิวเตอร์ ที่มีความสามารถต่างๆ คือ ผู้ใช้สามารถวาดภาพ, รูปทรงต่างๆ , ข้อความ, และสามารถส่งข้อมูลแบบ multicast ผ่านเครือข่ายไปแสดงยังหน้าจอของผู้ใช้ต่างๆ ได้พร้อมกัน โดยผู้ใช้แต่ละคนสามารถเลือกได้ว่า จะรับหรือส่งข้อมูล(เช่น รูปภาพต่างๆ) ของ ผู้ใช้คนใดก็ได้

Multi-user Whiteboard Component นี้เมื่อพัฒนาเสร็จจะ คอมไพล์ได้เป็นไฟล์ นามสกุล .dll ซึ่งก็คือ .Net Assembly นั่นเอง ซึ่งสามารถไฟล์นี้นำไปใช้เป็นส่วนประกอบ ได้ทั้งใน วินโดว์ แอปพลิเคชันซึ่งพัฒนาบน .Net Framework และ เว็บแอปพลิเคชัน ซึ่งใช้ เว็บเบราว์เซอร์ของ ไมโครซอฟต์ในการแสดงผล โดยที่ผู้พัฒนาวินโดว์แอปพลิเคชันหรือ เว็บแอปพลิเคชัน สามารถนำคอมโพเนนต์นี้ไปใช้เป็นส่วนประกอบได้อย่างง่ายดาย และถ้าหาก ตัวคอมโพเนนต์นี้มีการแก้ไขหรือเปลี่ยนแปลง ผู้พัฒนาวินโดว์แอปพลิเคชันหรือ เว็บแอปพลิเคชันก็ไม่จำเป็นต้องทำการคอมไพล์ใหม่ Multi-user Whiteboard Component นี้สามารถรองรับการเพิ่มความสามารถได้อย่างไม่จำกัดตามความต้องการของ ผู้พัฒนาแอปพลิเคชันแต่ละราย ผู้พัฒนาแอปพลิเคชันสามารถพัฒนา Multi-user Whiteboard Plug-ins ต่างๆ ขึ้นมาตามที่คุณพัฒนาแอปพลิเคชันต้องการ โดย plug-in ที่พัฒนานี้จะต้องใช้ภาษาใดก็ได้ที่ .Net Framework สนับสนุน และอิมพลิเมนต์ตามอินเตอร์เฟซ มาตรฐานที่กำหนดโดย Multi-user Whiteboard Component เมื่อ plug-in พัฒนาเสร็จก็เป็น ไฟล์ .dll ก็คือ .Net Assembly เช่นกัน ซึ่งสามารถนำ plug-in ที่ผู้พัฒนา plug-in แต่ละรายมาใช้เพิ่มความสามารถให้กับ Multi-user Whiteboard Component ได้โดย Multi-user Whiteboard Component ไม่จำเป็นต้องคอมไพล์ใหม่แต่อย่างใด และง่ายดายเพียงแค่นำไฟล์ .dll ของ Multi-user Whiteboard Plug-in ที่พัฒนาเสร็จไปไว้ที่ ไดเรกทอรีเดียวกับ ไฟล์ .dll Multi-user Whiteboard Component Multi-user Whiteboard Component ก็จะได้รับความสามารถของ plug in นั้นเพิ่มเติม



รูปที่ 3.2 แสดงอินเตอร์เฟซมาตรฐานที่ Multi-user Whiteboard Plug-in ต้อง อิมพลิเมนต์

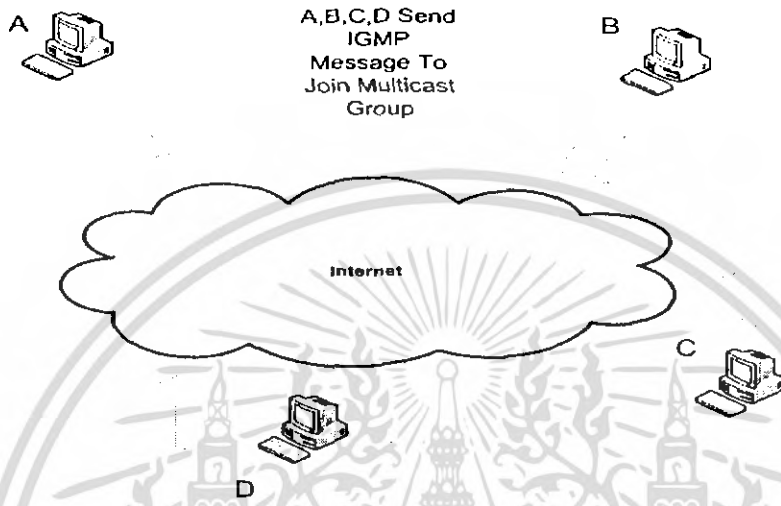
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดง class diagram ที่ generate จาก Microsoft Visual Studio 2005

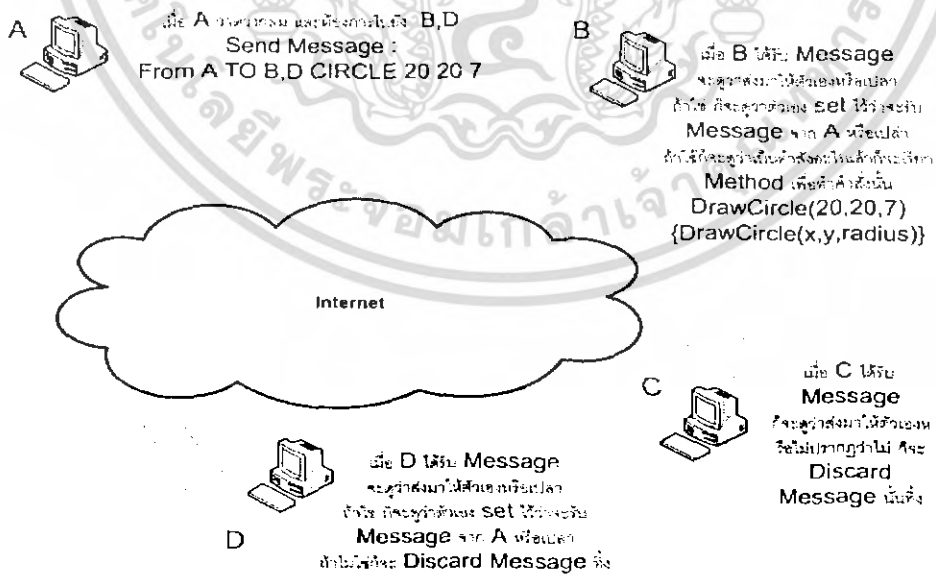
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับวิธีการส่งข้อมูลของ Multi-user Whiteboard Component จะเป็นแบบ UDP Multicast ซึ่งจะมีขั้นตอนโดย Multi-user Whiteboard Component ที่ต้องการจะรับหรือส่งข้อมูลกับ Multi-user Whiteboard Component ตัวอื่นๆ ก็จะใช้ IGMP(Internet Group Management Protocols) ให้แก่ไปยัง Router เพื่อ join multicast group จากนั้นก็สามารถส่งและรับข้อมูลของ Multicast Group ที่ทำการ Join ได้



รูปที่ 3.4 แสดงการ join multicast group เพื่อเริ่มต้นการรับส่งข้อมูล

เมื่อ join multicast group แล้ว Multi-user Whiteboard Component จะสามารถทำการรับส่งข้อมูลแบบ Multicast ได้ และมีรูปแบบการรับส่งข้อมูล ตามที่แสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงรูปแบบการส่งข้อมูลของ Multi-user Whiteboard Component

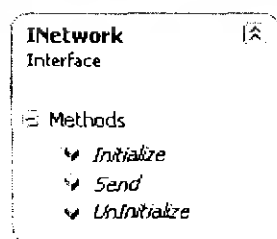
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 จะเห็นได้ว่า Multi-user Whiteboard Component แต่ละตัวจำเป็นต้องรู้ว่า มีใครในระบบ ออนไลน์อยู่บ้างจึงจะสามารถเลือกได้ว่า จะส่งข้อมูลให้ใครหรือจะรับข้อมูลจากใคร เพื่อให้ขอบเขตของโครงการนี้ไม่กว้างเกินไป ผู้พัฒนาโครงการนี้ได้คิดโปรโตคอลอย่างง่ายทำหน้าที่ ให้ข้อมูลแก่ Multi-user Whiteboard Component ที่ออนไลน์อยู่ในระบบว่า ขณะนี้มีใครออนไลน์อยู่ในระบบบ้าง

โดยโปรโตคอลที่ทำหน้าที่ให้ข้อมูลผู้ที่ออนไลน์ นี้ที่มีวิธีการทำงานคือ เริ่มต้นจะส่ง hello message ที่มีชื่อและ timestamp ซึ่งเป็นเวลาที่เริ่มทำงานออกไป หากไม่มีใครตอบกลับมาแสดงว่ามีตัวเองอยู่ในระบบคนเดียว หากมีผู้อื่นอยู่ในระบบ แล้วเมื่อได้รับ hello message แล้ว จะเก็บชื่อของผู้ที่ส่ง hello packet ไว้ใน online list และดูว่าเวลาเริ่มต้นทำงานของตัวเองน้อย(ก่อน)สุดใน online list หรือไม่ หากใช่ก็จะถือว่าตัวเองเป็น root ซึ่ง root ก็จะทำการ multicast online list message ซึ่งมี timestamp ของ root นี้ไปให้ผู้อื่น เมื่อตัวอื่นๆ ได้ multicast online list message ก็ จะดูว่า timestamp น้อย(ก่อน)กว่าของตัวเองหรือไม่ ถ้าใช่ ทำการ update ออนไลน์ list ของตัวเอง ก็จะทำให้รู้ได้ว่ามีใครบ้างกำลังออนไลน์อยู่ในระบบ และหากผู้ใดต้องการออกจากระบบ ก็ จะ multicast goodbye message ออกไป เมื่อตัวอื่นๆ ได้รับ goodbye message แล้วก็จะ ลบชื่อของผู้ที่ส่ง goodbye message ออกจาก online list ของตัวเอง

ข้อจำกัดของโปรโตคอลนี้คือเครื่องคอมพิวเตอร์ของผู้ใช้ทุกคน ต้องตั้งเวลาให้เท่ากัน ระบบจึงสามารถทำงานได้อย่างถูกต้อง

อย่างไรก็ดี โครงการนี้ได้อิมพลีเมนต์ ในส่วนของ network ใน network class ซึ่งเป็นอิสระจากส่วนอื่นๆ โดย network class นี้ถ่ายทอดจาก INetwork อินเตอร์เฟส ในอนาคตหากต้องการปรับปรุงโปรโตคอลหรือวิธีการต่างๆ ในการรับส่งข้อมูลให้เหมาะสมและดีขึ้น ก็ยังสามารถทำได้โดยไม่ต้องยุ่ง กับ code ในส่วนอื่น เพียงแค่ network class ที่อิมพลีเมนต์มาใหม่จะต้อง อิมพลีเมนต์ตาม INetwork อินเตอร์เฟส ก็สามารถเปลี่ยนรูปแบบการส่งข้อมูล ของ Multi-user Whiteboard Component ได้ตามที่นั้น network class Implement



รูปที่ 3.6 แสดงข้อมูลของ INetwork อินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2. Multi-user Whiteboard Plug-ins

เพื่อให้ Multi-user Whiteboard Component สามารถใช้งานได้อย่างยืดหยุ่นและสามารถเพิ่มความสามารถได้อย่างไม่จำกัด Multi-user Whiteboard Component จึงเปิดโอกาสให้สามารถพัฒนา plug-in ไปเพิ่มความสามารถได้อย่างไม่จำกัดจำนวนตัวของ plug-in และ Multi-user Whiteboard Component สามารถเลือก load ไฟล์ .dll ของ plug-in ไปรันได้อย่างใดนามิกตามใจชอบขณะ runtime โดยมีข้อแม้ว่า Multi-user Whiteboard Plug-ins ต้องพัฒนาตามข้อกำหนดต่อไปนี้

1. ต้องพัฒนาโดยภายใต้เทคโนโลยี .Net Framework
2. ต้องถ่ายทอดอินเตอร์เฟส IPlugin ที่กำหนดโดย Multi-user Whiteboard Component ตามรูปที่ 3.2
3. ต้องตั้งชื่อไฟล์ .dll ของ plug-in ให้ขึ้นต้นด้วย "Whiteboard_" แล้วตามด้วย ชื่อ plug-in นั้นๆ
4. plug-in ต้องไม่ใช่ critical resource ที่ Multi-user Whiteboard Component ใช้ เช่น multicast ip และ port ที่ Multi-user Whiteboard Component ใช้

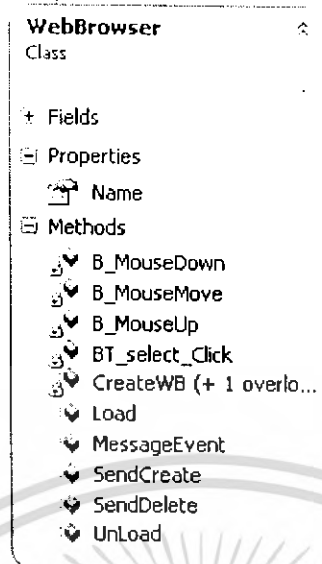
ในโครงการนี้ได้ทดลองพัฒนา plug-in มาเพื่อทดสอบความสามารถในการรองรับ plug-in ของ Multi-user Whiteboard Component โดย plug-in ที่พัฒนามีสองตัวได้แก่ 1. Web browser Plug-in 2.Questionnaire Plug-in

3.2.1 Web browser Plug-in

เป็น plug-in ที่เสริมความสามารถให้ Multi-user Whiteboard Component สามารถแสดงผลเว็บเพจที่ต้องการจะนำเสนอไปแสดงยัง ผู้ใช้คนอื่นๆ ที่ออนไลน์อยู่ในระบบ โดยที่ผู้ที่ต้องการจะแสดงสามารถระบุ URL ของ เว็บเพจที่ต้องการจะแสดง และยังสามารถระบุเพิ่มได้อีกว่าจะอนุญาตให้สามารถ navigate ไป เว็บเพจอื่นได้อีกหรือไม่ โดยผู้ที่ต้องการจะนำเสนอ เว็บเพจสามารถ เลือกปุ่มเครื่องมือจาก web browser plug-in เมนู แล้วก็วาด เว็บเพจได้ในลักษณะเดียวกันกับการวาดรูปอื่นๆ เว็บเพจ ที่ต้องการก็จะไปปรากฏ ยัง Multi-user Whiteboard Component อื่นๆ ที่ ต้องการจะแสดงผล

Web browser มีข้อจำกัดในการใช้งานดังนี้คือไม่สามารถใช้งานได้ในกรณีที่ Multi-user Whiteboard Component เป็นส่วนประกอบ ของ Web Application

Web browser Plug-in นี้ถ่ายทอดอินเตอร์เฟส IPlugin ที่กำหนดโดย Multi-user Whiteboard Component และ plug-in ตัวนี้เป็นไฟล์ที่มีชื่อว่า "Whiteboard_Webbrowser.dll" โดย plug-in ตัวนี้มีรายละเอียดดังรูปที่ 3.7



รูปที่ 3.7 แสดง class web browser plug-in

3.2.2 Questionnaire Plug-in

เป็น plug-in ที่เสริมความสามารถให้ Multi-user Whiteboard Component สามารถส่งหาคำถามไปยังผู้ที่ออนไลน์อยู่ในระบบ โดย plug-in ตัวนี้จะแบ่งเป็น 2 ส่วน ส่วนแรก ทำหน้าที่ติดต่อกับฐานข้อมูล ทำหน้าที่ดึงคำถามมาจากฐานข้อมูลและส่งไปยังผู้ทำแบบสอบถาม ส่วนที่สอง ทำหน้าที่รับข้อมูลจากแบบสอบถามไปแสดงผล และจับเวลาการทำแบบสอบถามและสรุปคะแนนจากแบบสอบถามจากนั้นส่งผลลัพธ์กลับไปแสดงผลยังส่วนแรก

คุณสมบัติของ Plug-in ตัวนี้ คือ เป็นระบบข้อสอบ คือ

- สามารถทำระบบสมาชิก
- สามารถออกข้อสอบ
- สามารถทำข้อสอบ
- สามารถรายงานผลคะแนน

โดยคุณสมบัติข้างต้นที่กล่าวมาทำให้ Plug-ins ตัวนี้แบ่งการทำงานออกเป็น 2 ส่วน คือ

- Admin Plug-ins

เป็น Plug-in ที่ทำหน้าที่เป็น Admin โดยมีหน้าที่ต่างๆดังนี้ คือ

- จัดทำระบบสมาชิก
- จัดทำข้อสอบ

-Client Plug-ins

- ทำข้อสอบ

โดย Plug-in ทั้ง 2 ตัวต้องมีการใช้งานร่วมกับ Whiteboard Component

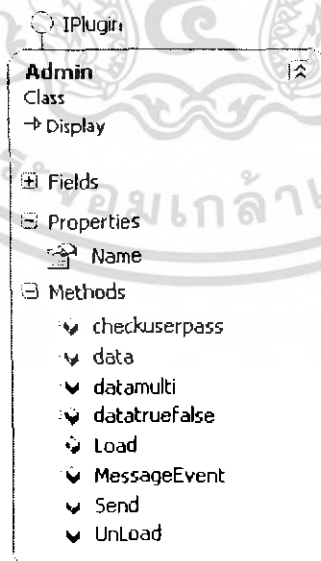
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของการทำงาน Questionnaire Plug-in

1. user ที่จะ เป็น Admin ต้องทำการสร้างระบบ member กับ ตัวข้อสอบเป็นดังรูป
2. user ที่จะ เป็น Admin ต้องทำการเลือกประเภทของข้อสอบว่าต้องการแบบใดระหว่าง แบบถูกผิดและแบบ multiple choice
3. user ที่ เป็น client จะทำการยืนยันการมีสิทธิในการเข้าใช้งาน โดยป้อน username และ password
4. ถ้า user ที่ เป็น client มีสิทธิการใช้งาน Admin จะส่งข้อสอบไปให้ตัว Client
5. เมื่อ user ทำข้อสอบเสร็จเรียบร้อยแล้ว Client จะทำการส่งคะแนนที่ได้กลับไปสู่ Admin



รูปที่ 3.8 แสดง class questionnaire client plug-in



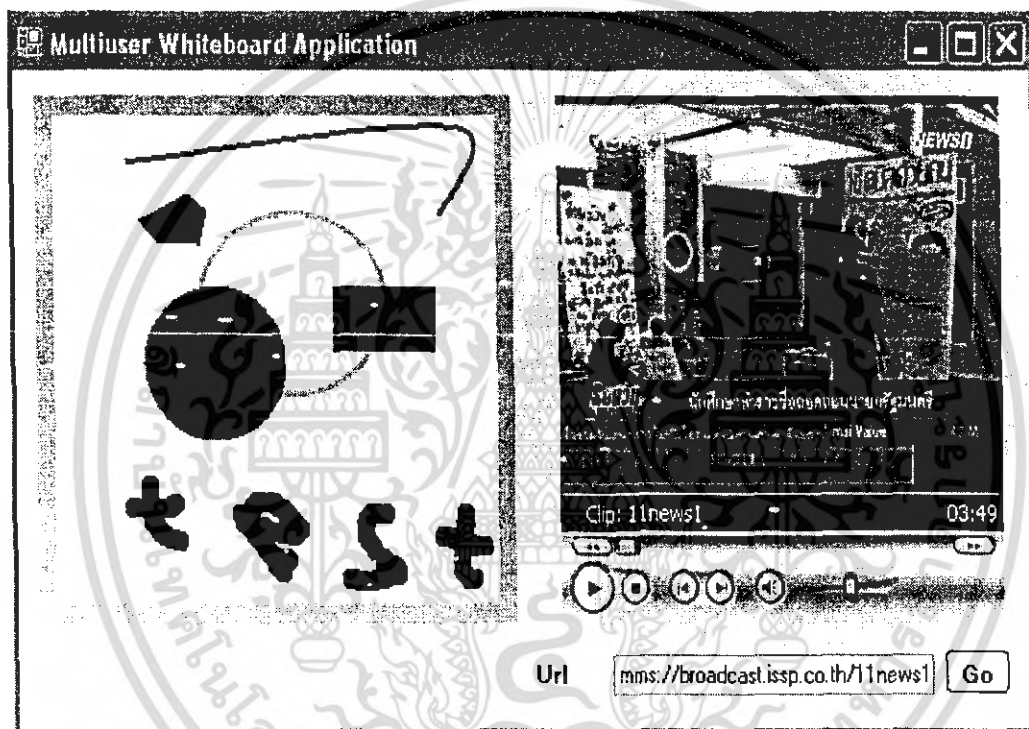
รูปที่ 3.9 แสดง class questionnaire server plug-in

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Multi-user Whiteboard Application

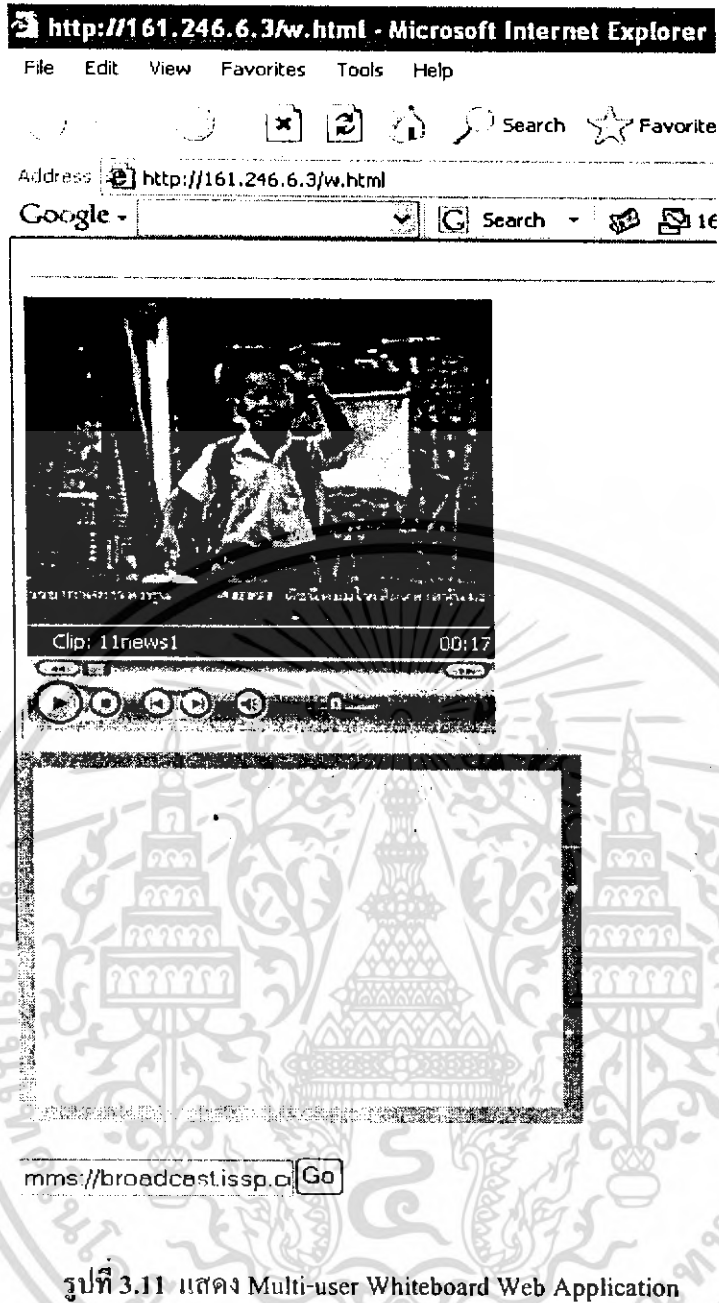
ส่วนนี้ประกอบวินโดว์แอปพลิเคชัน และ เว็บแอปพลิเคชัน ซึ่งจะบรรจุ Multi-user Whiteboard Component เพื่อทดสอบว่า Multi-user Whiteboard Component สามารถใช้เป็นส่วนประกอบและทำงานได้ทั้งบนเว็บแอปพลิเคชันและวินโดว์แอปพลิเคชันได้อย่างไม่มีปัญหา

รูปข้างล่างนี้เป็นการนำ Multi-user Whiteboard Component ซึ่งเป็น .Net Component และ Window Media Player Component ซึ่งเป็น COM component ที่พัฒนาโดยไมโครซอฟต์ มาสร้าง Multi-user Whiteboard Application



รูปที่ 3.10 แสดง Multi-user Whiteboard Windows Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดง Multi-user Whiteboard Web Application

โดยที่การพัฒนาโปรแกรมโดยใช้ คอมพิวเตอร์ ผู้พัฒนาสามารถสร้างแอปพลิเคชันได้อย่างรวดเร็ว โดยผู้เขียนโปรแกรมสามารถสร้างแอปพลิเคชันตามรูปข้างบนโดยใช้เวลาพัฒนาให้เสร็จเพียงไม่ถึง 10 นาที ก็จะได้แอปพลิเคชันที่สามารถดู Multimedia Stream และวาดรูปผ่านเครือข่ายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

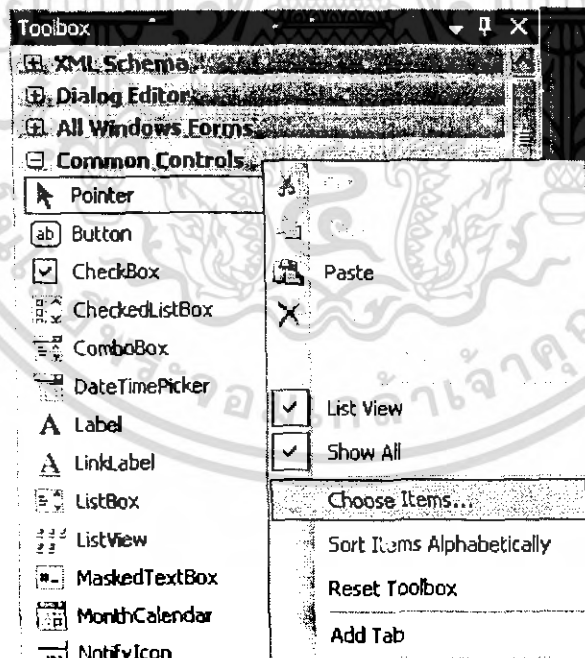
การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงขั้นตอนการทดลองการนำ Multi-user Whiteboard Component ไปเป็นส่วนประกอบของ วินโดว์แอปพลิเคชัน และเว็บแอปพลิเคชัน การทดลองการทำงานของ Multi-user Whiteboard Component เมื่อเป็นส่วนประกอบของ Multi-user Whiteboard Application ทั้ง ในส่วน ที่เป็น วินโดว์แอปพลิเคชัน และเว็บแอปพลิเคชัน อีกทั้งกล่าวถึงการนำ Multi-user Whiteboard Component มาทดลองทำงานร่วมกับ Multi-user Whiteboard Plug-in อีกด้วย

4.1 การทดลองการนำ Multi-user Whiteboard Component ไปใช้งาน

4.1.1 การนำ Multi-user Whiteboard Component ไปใช้ในส่วนของวินโดว์แอปพลิเคชัน การนำไปใช้สามารถทำได้หลายวิธี แต่เพื่อความสะดวกเราสามารถ ใช้ Visual Studio 2005 ช่วยทำงานให้เราโดยต้องทำตามวิธีการข้างล่างนี้

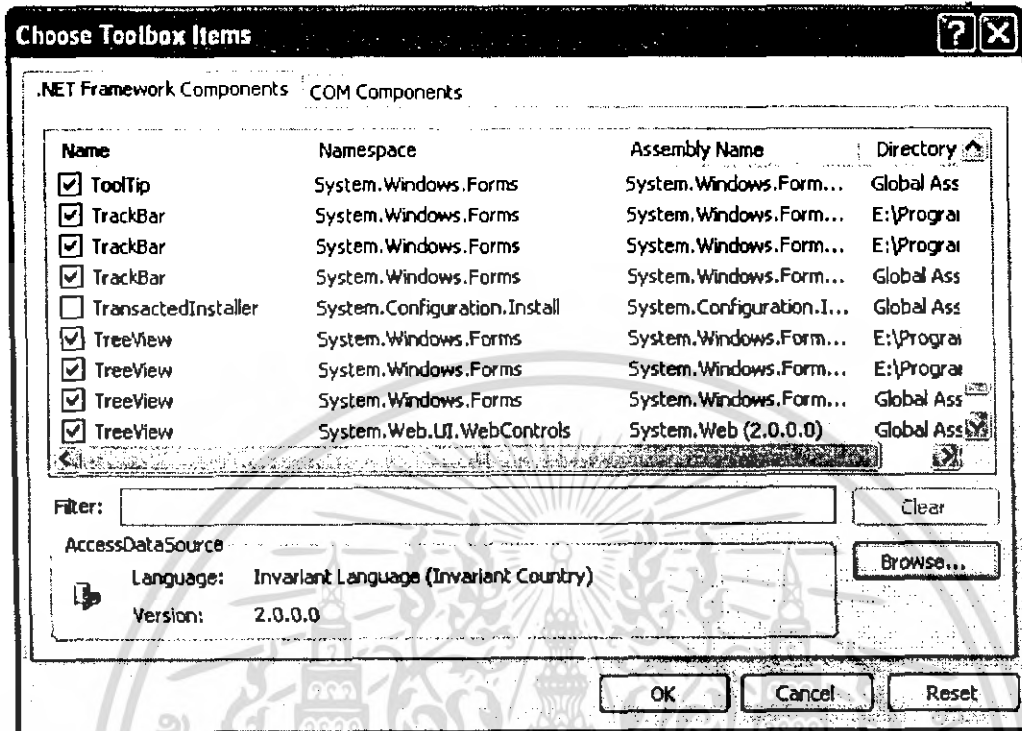
1. ก่อนอื่นต้อง Add Component ของเราให้ Toolbox ใน Visual Studio 2005 รู้จักก่อน



รูปที่ 4.1 แสดงการเพิ่มคอมโพเนนต์ใน Toolbox

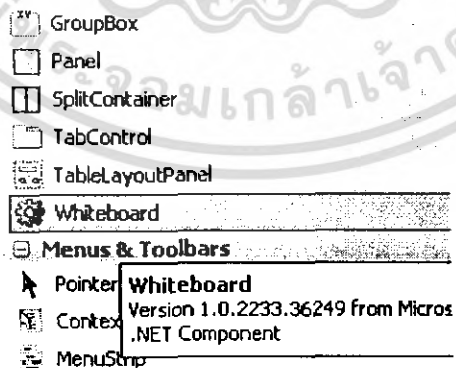
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จดบันทึกการ Browse และเลือกไฟล์ .dll ของ Multi-user Whiteboard Component



รูปที่ 4.2 แสดงการ browse หา .dll File

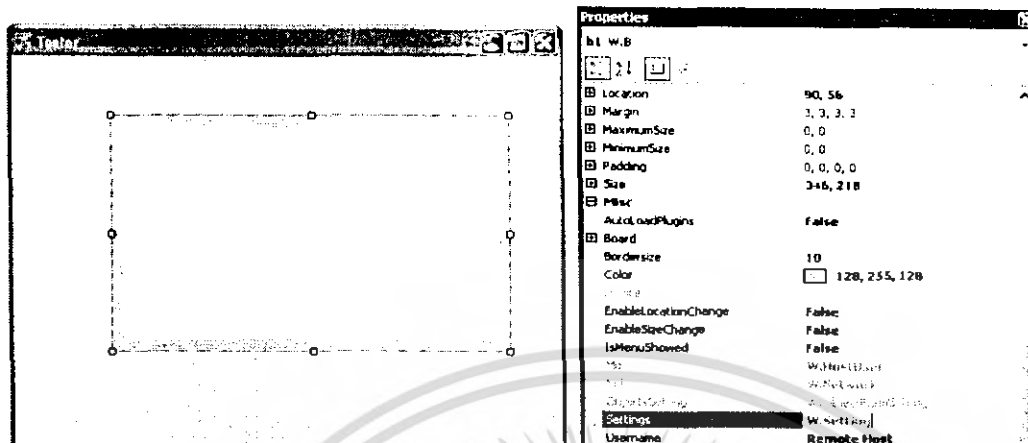
3. เมื่อ Toolbox รู้จักแล้วก็สามารถเลือก Multi-user Whiteboard Component จาก Toolbox ไปใช้งานได้เลย



รูปที่ 4.3 แสดงการเลือกคอมโพเนนต์จาก Toolbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.สามารถนำคอมโพเนนต์ไปวางที่ใดก็ได้และเลือกขนาดได้ตามต้องการ และยังสามารถกำหนดคุณสมบัติอื่นๆ ได้อีกทางหนึ่งคือขอมขนาดขอบทาง Properties box



รูปที่ 4.4 แสดงการเลือกคอมโพเนนต์จาก Toolbox ไปวางบนชิ้นงานที่กำลังพัฒนา

5 สามารถเขียน code ควบคุมการทำงานของคอมโพเนนต์ได้เช่น Code ข้างล่างนี้

```
private void Form1_Load(object sender, EventArgs e)
{
    b1.Initialize();
    Whiteboard.BorderSize = 12;
    AutoLoadPlugins = true;
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    whiteboard.UnInitialize();
}
```

4.1.2 การนำ Multi-user Whiteboard Component ไปใช้ในส่วนของเว็บแอปพลิเคชัน

การนำคอมโพเนนต์ไปใช้งานบนเว็บเบราว์เซอร์สามารถทำได้โดยเพิ่มแทรก เข้าไปใน HTML ไฟล์ดัง โค้ด ดังต่อไปนี้

```
<OBJECT id="wb" classid="W.dll#W.B" width=800 height=600> </OBJECT>
```

id ใช้เป็น Identifier สำหรับ อ้างถึง object ของ Multi-user Whiteboard Component นี้

classid คือชื่อไฟล์และ namespace ของ Multi-user Whiteboard Component นี้

width คือขนาดความกว้าง Multi-user Whiteboard Component นี้

height คือขนาดความสูง Multi-user Whiteboard Component นี้

และสามารถ เรียก Method ของ Multi-user Whiteboard Component Object ได้โดยผ่านทาง

Javascript ดังเช่น ตัวอย่างข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

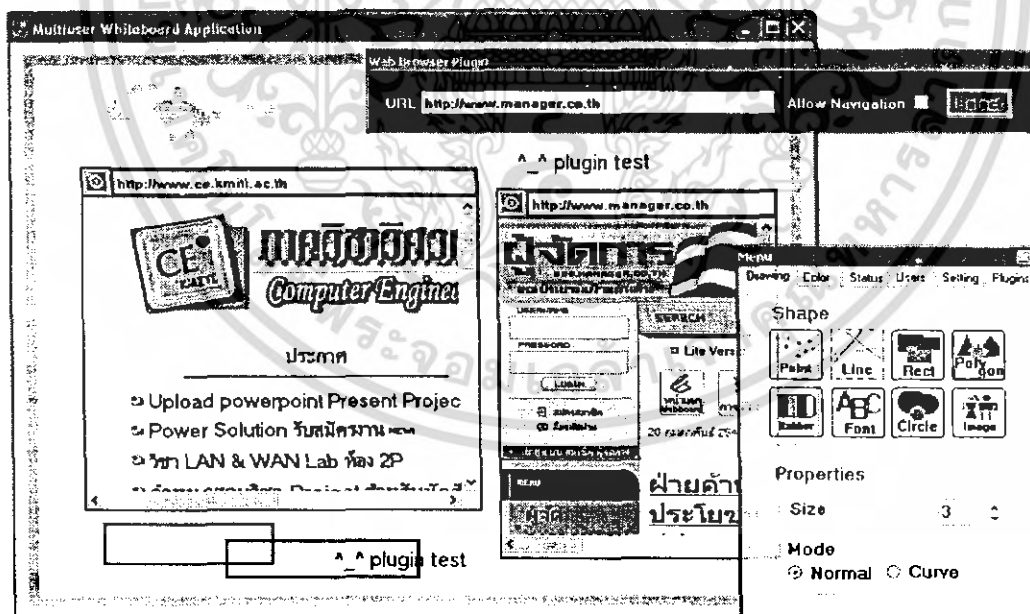
<script language="javascript">
    function start()
    {
        wb.Initialize();
        wb.BorderSize = 10;
    }
    function stop()
    {
        wb.UnInitialize();
    }
</script>

```

4.2 ทดสอบความสามารถต่างๆ ของ Multi-user Whiteboard Plug-in

4.2.1 การทดสอบการทำงานของ Web browser Plug-in

การทดสอบการทำงานของ Web browser Plug-in เมื่อนำมาใช้ร่วมกับ Multi-user Whiteboard Component



รูปที่ 4.5 แสดงการทำงานของ Multi-user Whiteboard Component ร่วมกับ Web Browser Plug-in

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

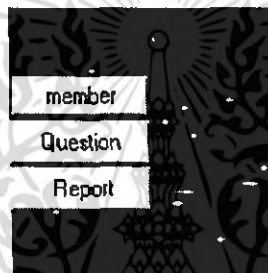
ในส่วนของวินโดวแอปพลิเคชัน Web browser Plug-in สามารถ Load มารันได้อย่างไม่มีปัญหาโดยสามารถใช้งานผ่าน Toolbox สีดำในรูปที่ 4.5 โดยสามารถกำหนดได้ว่า จะ Show URL. โค และวางลงไปยังตำแหน่งใด ใน Multi-user Whiteboard Component ก็ได้และมีขนาดเท่าไรก็ได้ อีกทั้งยังสามารถกำหนดให้ Browser ตัวนั้นสามารถ Navigation ไป ยัง URL อื่นได้หรือไม่

แต่ส่วนของเว็บแอปพลิเคชัน Web browser เช่น Internet Explorer ไม่สนับสนุนการทำงานของ Web browser Plug-in ดังนั้น plug-in ตัวนี้จึงทำงานได้เฉพาะบนวินโดวแอปพลิเคชัน

4.2.2 การทดสอบการทำงานของ Questionnaire Plug-in

การทำงานของ Plug-in ตัวนี้เป็นไปตามขั้นตอนข้างล่างนี้

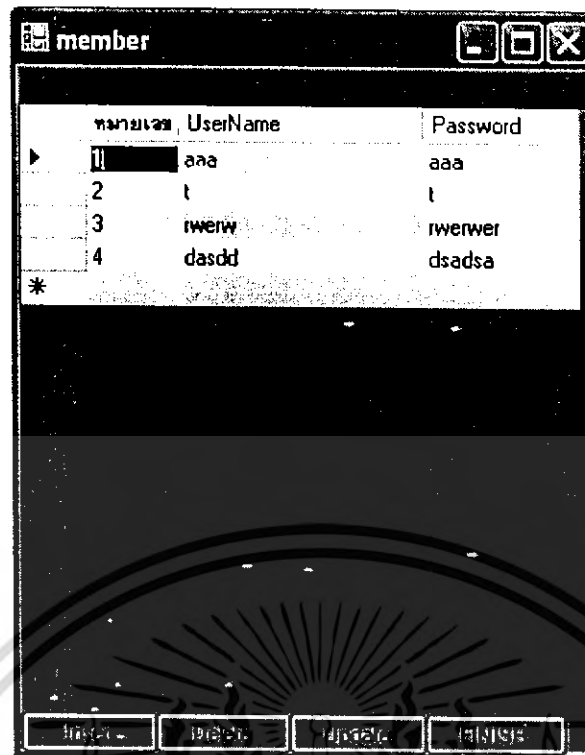
1. user ต้องเลือก Questionnaire Plug-ins อย่างใดอย่างหนึ่งระหว่าง Admin กับ Client โดยถ้า user เลือกที่จะเป็น Admin เมื่อ load Admin Plug-in ขึ้นมาก็จะมีลักษณะดังรูป



รูปที่ 4.6 แสดง User Interface ของ Admin Plug-in

2. user ที่จะเป็น Admin(Mother) ต้องทำการสร้างระบบ member กับ ตัวข้อสอบเป็นดังรูป ข้อมูลของระบบ member จะประกอบด้วยข้อมูลดังนี้
 - หมายเลขสมาชิก
 - Username
 - Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

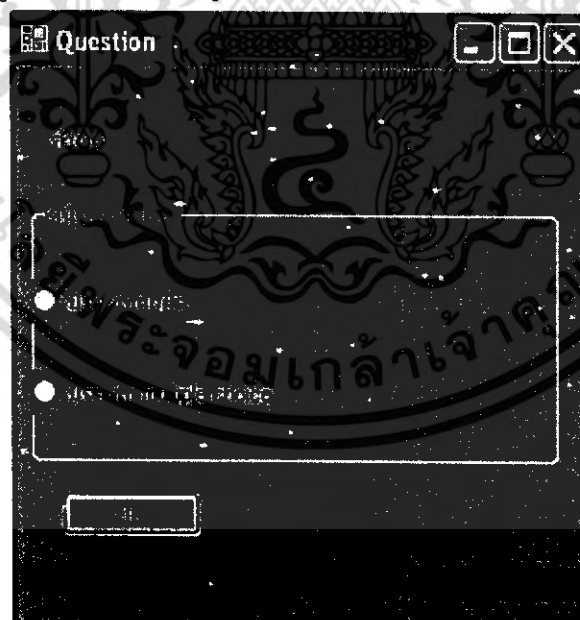


| หมายเลข | UserName | Password |
|---------|----------|----------|
| 1 | aaa | aaa |
| 2 | t | t |
| 3 | rwerw | rwerw |
| 4 | dasdd | dsadsa |

*
 [ปุ่มลบ] [ปุ่มเพิ่ม] [ปุ่มแก้ไข] [ปุ่มลบ]

รูปที่ 4.7 แสดงข้อมูลของระบบสมาชิก

3. user ที่จะเป็น Admin(Mother) ต้องทำการเลือกประเภทของข้อสอบว่าต้องการแบบใด ระหว่างแบบถูกผิดและแบบ multiple choice



Question

ชื่อคำถาม

ประเภทข้อสอบ

ถูกผิด

Multiple Choice

คำตอบ

[ปุ่มตกลง]

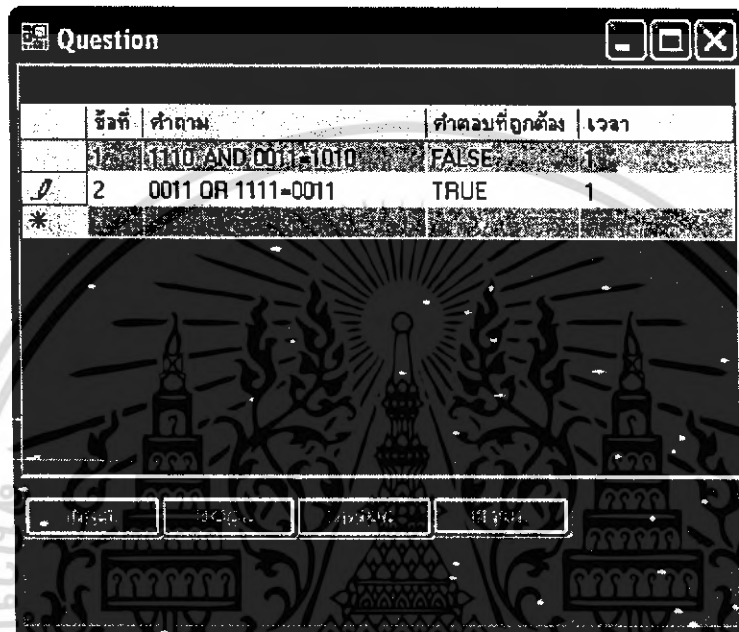
รูปที่ 4.8 แสดง User Interface ของการเลือกประเภทของคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า user ทำการเลือกประเภทถูกผิด

ข้อมูลของระบบข้อสอบแบบถูกผิด จะประกอบด้วยข้อมูลดังนี้

- หมายเลขข้อ
- คำถาม
- คำตอบที่ถูกต้อง
- เวลาที่กำหนด



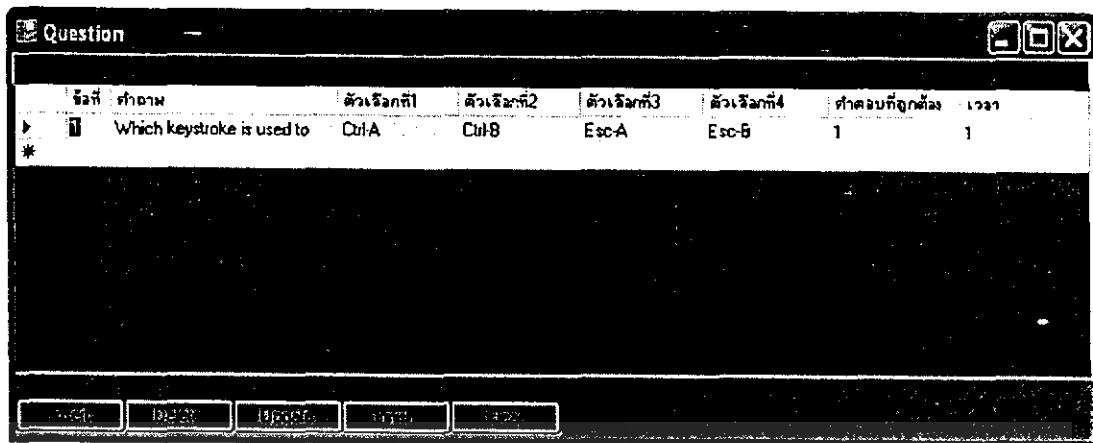
รูปที่ 4.9 แสดง user interface ของข้อมูลแบบถูกผิด

ถ้า user ทำการเลือกประเภท multiple choice

ข้อมูลของระบบข้อสอบแบบ multiple choice จะประกอบด้วยข้อมูลดังนี้

- หมายเลขข้อ
- คำถาม
- Choice 1
- Choice 2
- Choice 3
- Choice 4
- คำตอบที่ถูกต้อง
- เวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



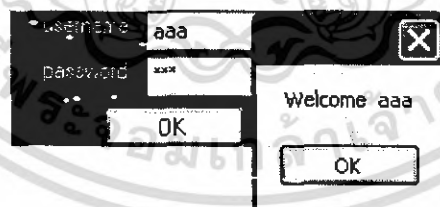
รูปที่ 4.10 แสดง user interface ของข้อมูลแบบ multiple choice

4. เมื่อ user ทำการ load Child Plug-in เข้ามาต้องป้อน username , password



รูปที่ 4.11 แสดง user interface ของ client Plug in

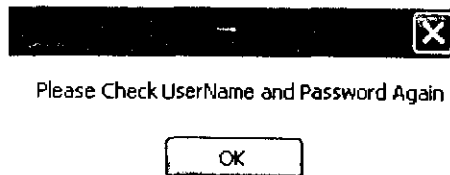
5. หลังจากนั้น Admin (Mother) จะทำการ check ว่า user มีสิทธิในการใช้งานหรือไม่
6. ถ้า user ที่เป็น client (Child) มีสิทธิการใช้งาน Admin(Mother) จะส่งข้อสอบไปที่ตัว Client (Child)



รูปที่ 4.12 แสดงการตอบรับการเข้าใช้งานของ client

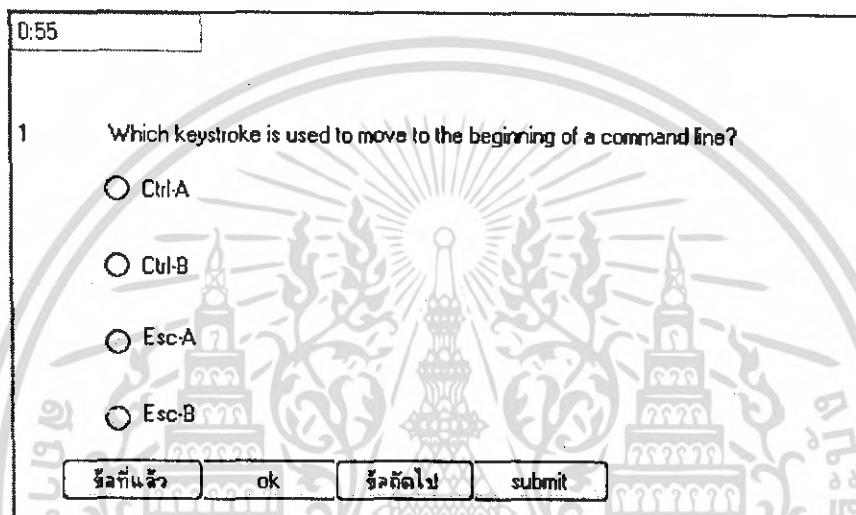
ถ้า user ที่เป็น client (Child) ไม่มีสิทธิการใช้งาน Admin(Mother) จะส่ง Message กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 แสดงความผิดพลาดที่เกิดขึ้นจากการที่ client ป้อน username หรือ password ผิด

7. เมื่อได้รับอนุญาตให้เข้าใช้งานแล้ว ข้อสอบจะถูก load ขึ้นมา



รูปที่ 4.14 แสดงตัวอย่างของข้อสอบที่ client ได้รับจาก Admin

8. เมื่อ user ทำข้อสอบเสร็จเรียบร้อยแล้ว Client (Child) จะทำการส่งคะแนนที่ได้กลับไปสู่ Admin (Mother)



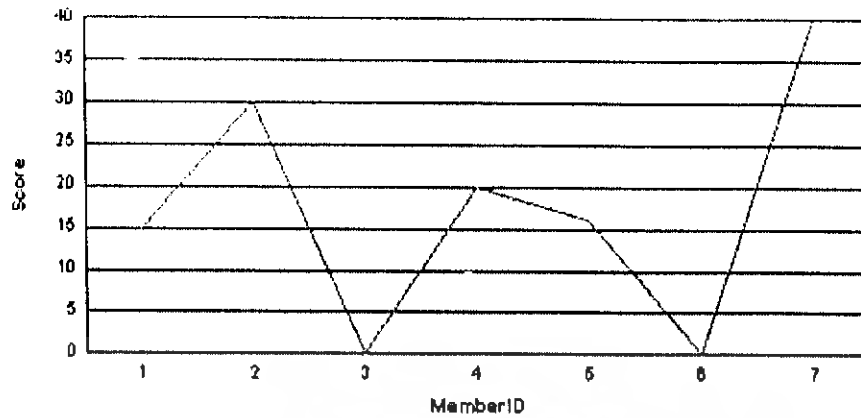
รูปที่ 4.15 แสดงการรายงานผลคะแนนให้ client ทราบ

9. หลังจากนั้น Admin ก็สามารถตรวจสอบผลคะแนนของแต่ละ user ที่ออกมาได้ดังรูปที่

4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Score



รูปที่ 4.16 แสดงกราฟคะแนนของสมาชิกแต่ละคน

REPORT SCORE

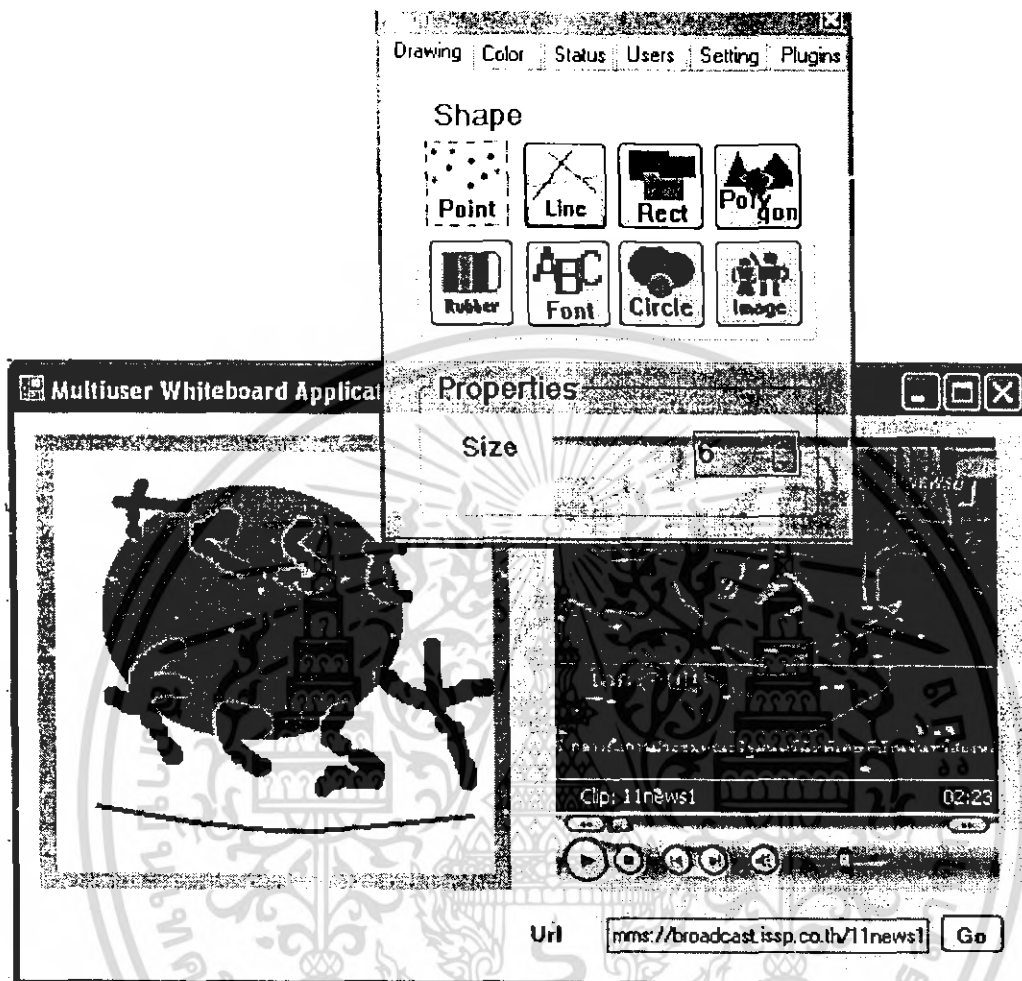
2/19/2006

| <u>MemberID</u> | <u>Score</u> |
|-----------------|--------------|
| 1 | 15 |
| 2 | 30 |
| 3 | 0 |
| 4 | 20 |
| 5 | 16 |
| 6 | 0 |
| 7 | 40 |

รูปที่ 4.17 แสดงรายงานของผลคะแนนของสมาชิกแต่ละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดสอบความการทำงาน ของ Multi-user Whiteboard Application



รูปที่ 4.18 แสดงการทำงาน ของ Multi-user Whiteboard Application

จากการทดลองนำ Multi-User Whiteboard Component มาสร้าง Multi-User Whiteboard Application ปรากฏว่าสามารถทำงานร่วมกับคอมพิวเตอร์ของผู้พัฒนารายอื่นได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทสรุป

การสร้างซอฟต์แวร์เชิงคอมโพเนนต์โดยใช้ .Net Framework สามารถทำได้สะดวก โดยทุกโปรแกรมที่สร้างโดยใช้ .Net Framework ไม่ว่าจะเป็นไฟล์ .dll หรือ .exe ซึ่งก็คือ .Net Assembly นั้นเอง ซึ่ง โครงสร้างพื้นฐานทางกายภาพของ .Net Assembly ก็มีความเป็น Component อยู่ในตัวเองอยู่แล้ว แต่ที่โปรแกรมส่วนใหญ่ไม่สามารถนำไปใช้เป็นส่วนประกอบของโปรแกรมอื่นได้นั้น เป็นเพราะว่า โปรแกรมทั้งหลายเหล่านั้นไม่ได้ออกแบบโครงสร้างทางตรรกอย่างเหมาะสมสำหรับการเป็นคอมโพเนนต์ ถ้าโปรแกรมได้รับการออกแบบมาให้เหมาะสมก็จะทำให้สามารถนำมาใช้เป็นส่วนประกอบของโปรแกรมอื่นได้สะดวก ทำให้เกิดประโยชน์อย่างมหาศาลในการวิจัย คือสามารถนำไปใช้เป็นส่วนประกอบของโปรแกรมใดๆก็ได้โดยไม่ต้อง พัฒนาส่วนของคอมโพเนนต์นั้นใหม่ ยิ่งกว่านั้นคอมโพเนนต์ที่ออกแบบมาอย่างเหมาะสมนั้นมีความยืดหยุ่นสามารถเพิ่มความสามารถให้เหมาะสมสำหรับสภาวะการใช้งานได้ โดยโปรแกรมที่ใช้คอมโพเนนต์เป็นส่วนประกอบนั้นไม่ต้องมีการแก้ไขแต่อย่างใด

5.2 วิจารณ์สิ่งที่ได้จากโครงการงาน

การพัฒนาโครงการนี้ทำให้ได้ความรู้ในการสร้างซอฟต์แวร์เชิงคอมโพเนนต์ เป็นอย่างดี อีกทั้งยังได้รับความรู้และประสบการณ์ ในการพัฒนาซอฟต์แวร์บน .Net Framework และทำให้เกิดความคุ้นเคยกับ .Net Framework Class Library ในส่วนต่างๆ เช่น System.Net , System.Reflection และ System.Threading และโครงการนี้ยังทำให้เกิดความชำนาญในภาษา C# อีกด้วย

5.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

ในส่วนของ class network ที่มีต้องมีการส่งข้อมูลรายชื่อผู้ออนไลน์อยู่ในระบบไปให้ ทุกๆ Multi-user Whiteboard Component โดยจะเลือก root ที่ทำหน้าที่ส่งข้อมูลดังกล่าว โดยจะเลือก root ตามเวลาเริ่มทำงาน ของ Multi-user Whiteboard Component แต่ละตัว ซึ่งจะมี ข้อจำกัดคือ ทุกๆ เครื่องคอมพิวเตอร์ ที่ อยู่ในระบบ ต้องตั้งเวลาเท่ากัน แนวทางการแก้ไขคืออาจจะใช้วิธีการพัฒนา server โปรแกรมทำหน้าที่ส่งข้อมูลรายชื่อผู้ที่ออนไลน์อยู่ในปัจจุบันไปยัง ทุกตัวที่ออนไลน์อยู่ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางการพัฒนาต่อ

1. พัฒนาในส่วนของผู้สเซอร์อินเตอร์เฟสให้มีความสวยงาม และให้ผู้ใช้สามารถใช้งานได้สะดวกมากขึ้น
2. พัฒนาในส่วนของ Multi-user Whiteboard Component ให้สามารถวาดรูปทรงอื่นๆ ได้เพิ่มขึ้น เช่นรูปทรงสามมิติต่างๆ
3. พัฒนา plugin เพื่อเพิ่มความสามารถต่างๆให้กับ Multi-user Whiteboard Component เช่น สำหรับแสดง Microsoft Windows Media Player เพื่อให้สามารถแสดง multimedia ชนิดต่างๆ ได้
4. พัฒนาในส่วนของ class ที่ถ่ายทอดคุณสมบัติ INetwork ให้ไม่มีข้อจำกัดตามที่กล่าวไว้ในบทที่ 3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Ian Sommerville., 2004, “Software Engineering”, England, Addison Wesley Press.
- [2] Karli Watson.David Espinosa. Zach Greenvoss., 2003.”Beginning Visual C#”,USA,Wrox Press
- [3] ศุภชัย สมพานิช.,2003,“คู่มือการเขียนโปรแกรมและใช้งาน Visual C#”,นนทบุรี InfoPress



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้