

โปรแกรมช่วยควบคุมการเรียนการสอนโดยใช้ COM (Component Object Model)

A Program for Assisting Instruction Using COM (Component Object Model)

จักรกริช พฤษการ *

ความเป็นมาของปัญหา

ปัจจุบันได้มีการนำระบบคอมพิวเตอร์เข้ามาประยุกต์ใช้กับการเรียนการสอนเป็นจำนวนมาก ไม่ว่าจะเป็น E-Learning, Virtual Classroom เป็นต้น เพื่อเป็นการตอบสนองต่อการระบบการศึกษาที่ต้องการเน้นให้ผู้เรียนสามารถเรียนรู้ได้ตลอดเวลา ไม่มีข้อจำกัดในเรื่องของสถานที่ หรือเวลา เป็นการขยายโอกาสทางการศึกษา และก่อให้เกิดการเรียนรู้ตลอดชีวิต ดังนั้นมหาวิทยาลัยทั่วโลกจึงได้ให้ความสนใจในการพัฒนาระบบการเรียนการสอนแบบออนไลน์กันเป็นจำนวนมาก แต่ในวิธีการของการสร้างบทเรียนแบบออนไลน์นี้ส่วนใหญ่จะใช้กับรายวิชาที่มีการบรรยายในห้องเรียน จึงทำให้ยังมีข้อจำกัดอยู่บางประการ เช่น

- ผู้เรียนไม่สามารถโต้ตอบ (interactive) กับผู้สอนได้ในทันทีที่มีปัญหา หรือข้อสงสัยต่าง ๆ
- ในบางรายวิชาที่ต้องมีการใช้ห้องปฏิบัติการคอมพิวเตอร์ การสร้างบทเรียนทำได้ค่อนข้างยาก และผู้เรียนต้องใช้เวลาในการเรียนรู้ค่อนข้างมาก
- ผู้สอนไม่สามารถประเมินความสามารถของผู้เรียนได้ในทันที

ดังนั้นในงานวิจัยชิ้นนี้จึงมุ่งเน้นในเรื่องของการควบคุมการเรียนการสอนภายในห้องปฏิบัติการคอมพิวเตอร์ ซึ่งจะต้องมีการอภิปราย สาธิตการใช้งาน และต้องการตอบโต้กันระหว่างผู้เรียนและผู้สอนแบบทันทีทันใด (real time) โดยที่ผู้สอนสามารถควบคุมหน้าจอของผู้เรียนทุกคนให้แสดงไปพร้อม ๆ กับหน้าจอของผู้สอนที่กำลังสาธิตการใช้งานในขณะนั้น ทำให้ผู้เรียนสามารถติดตามการสอนของผู้สอนได้อย่างต่อเนื่อง นอกจากนี้แล้วการสอนด้วยวิธีนี้ไม่จำเป็นที่จะต้องให้ผู้เรียนอยู่ในห้องเดียวกันกับผู้สอน เนื่องจากจะมีการส่งข้อมูลเสียงออกไปพร้อมข้อมูลภาพ ทำให้ผู้เรียนสามารถอยู่ที่ใดก็ได้ ก็สามารถเรียนได้เหมือนกัน

* อาจารย์ สาขาการจัดการสารสนเทศและคอมพิวเตอร์ คณะพาณิชยศาสตร์และการจัดการ
มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตตรัง

วัตถุประสงค์ของการวิจัย

เพื่อศึกษาและวิจัยถึงการนำ COM Object มาประยุกต์ใช้งานกับการเรียนการสอนแบบ Real Time ผ่านระบบเครือข่าย

ขอบเขตของการวิจัย

ในการวิจัยนี้จะมีการออกแบบและพัฒนาโปรแกรมซึ่งจะประกอบไปด้วย 2 ส่วนด้วยกันคือ

1. ออกแบบและพัฒนาโปรแกรมในส่วนของเซิร์ฟเวอร์ (Server) โดยโปรแกรมจะต้องมีความสามารถดังต่อไปนี้

- สามารถทำงานได้คล้ายคลึงกับ DNS (Domain Name System) คือ สามารถที่จะแปลงชื่อของผู้ใช้งาน (user) ให้เป็น IP Address ได้

- จะต้องสามารถรองรับการติดต่อ (connection) จากไคลเอนต์หลายเครื่องพร้อมกันได้

2. ออกแบบและพัฒนาโปรแกรมในส่วนของไคลเอนต์ (Client) โดยโปรแกรมจะต้องมีความสามารถดังต่อไปนี้

- สามารถติดต่อกับเซิร์ฟเวอร์ เพื่อทำการขอ IP Address ของไคลเอนต์เครื่องอื่น ๆ เพื่อให้ไคลเอนต์สามารถติดต่อสื่อสารถึงกันและกันได้

- ไคลเอนต์จะต้องสามารถที่จะรับ-ส่ง ข้อมูลเสียงระหว่างกันได้

- ไคลเอนต์จะต้องสามารถที่จะทำการแชร์ (Sharing) หน้าจอของ Application กันได้

วิธีการดำเนินงานวิจัย

การวิจัยนี้เป็นการพัฒนาโปรแกรม มีจุดมุ่งหมายเพื่อสร้างและทดสอบการใช้งาน ซึ่งขั้นตอนในการทำมี 3 ขั้นตอน ดังนี้

ขั้นตอนที่ 1 การศึกษา COM Object ที่จะต้องนำมาใช้งาน

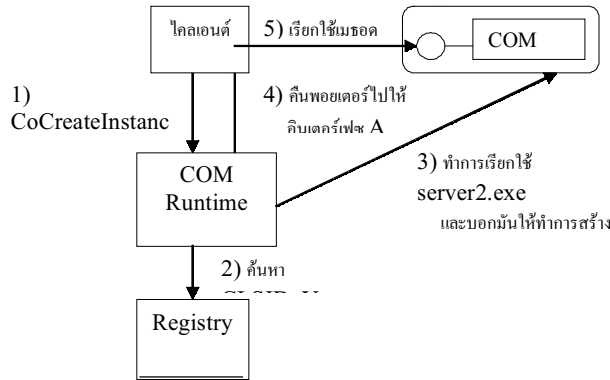
ขั้นตอนที่ 2 การออกแบบโปรแกรม

ขั้นตอนที่ 3 การทดสอบโปรแกรมที่ได้พัฒนาขึ้นมา

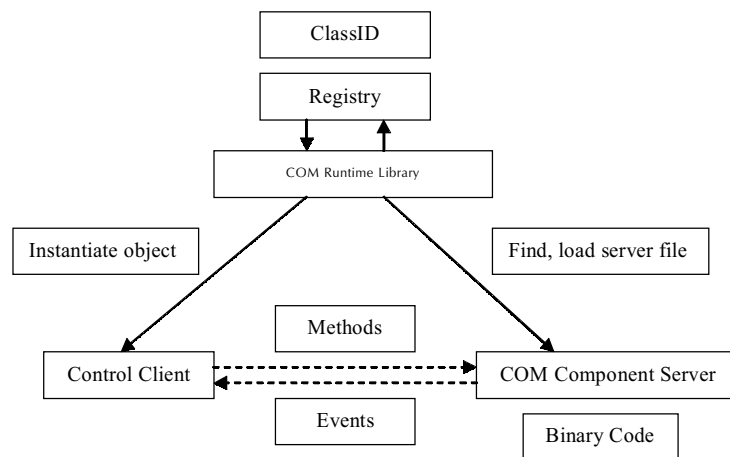
ขั้นตอนที่ 1 การศึกษา COM Object ที่จะต้องนำมาใช้งาน

การพัฒนาซอฟต์แวร์ในปัจจุบัน จะเน้นไปถึงการนำเทคโนโลยีทางด้านออบเจกต์ (object) มาใช้กันเป็นจำนวนมาก แต่ปัญหาของแต่ละภาษาที่ใช้ในการเขียนโปรแกรมแบบออบเจกต์นั้น มีมุมมองในเรื่องของออบเจกต์แตกต่างกันไปบ้าง ถึงแม้ว่าจะไม่ได้มากมายนัก แต่ก็ยังไม่เป็นมาตรฐานเดียวกัน ดังนั้นจึงมีความคิดที่จะทำให้การเข้าถึงออบเจกต์นั้นเป็นมาตรฐานเดียวกัน ซึ่งนั่นหมายความว่าไม่ว่าเราจะใช้ภาษาอะไรในการเขียนโปรแกรมก็สามารถที่จะใช้งานออบเจกต์ได้เหมือนกัน จึงได้มีการคิดสร้าง COM (Component Object Model) ขึ้นมา เพื่อเป็นมาตรฐานสำหรับกลุ่มของซอฟต์แวร์ซึ่งต้องการเข้าถึงบริการที่ให้โดยซอฟต์แวร์อื่น

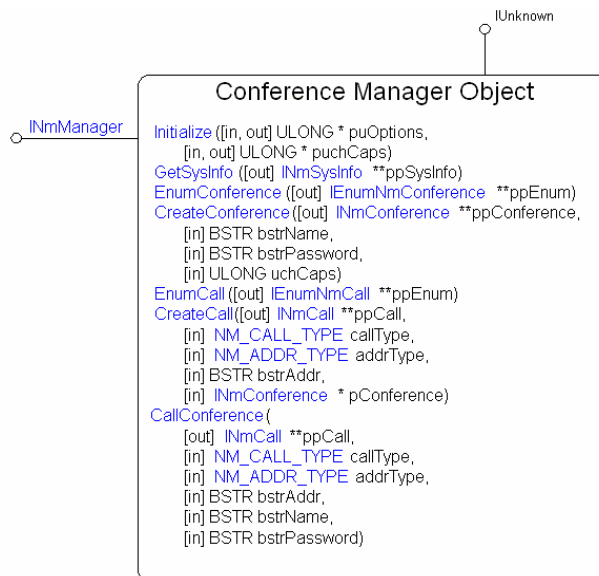
ก่อนที่เราสามารถเข้าถึงบริการของ COM Object ได้นั้น จะต้องมีทำการรับพอยน์เตอร์ของอินเตอร์เฟซของ COM นั้นๆ มาก่อน เมื่อได้พอยน์เตอร์มาแล้วก็สามารถที่จะเรียกใช้เมธอด (method) ต่างๆ ในอินเตอร์เฟซที่มันอ้างอิงได้



COM Object ที่มีการนำมาใช้งานนั้นจะถูกอิมพลีเมนต์ (implement) อยู่ในรูปของไฟล์นามสกุล DLL หรือ EXE (binary object) ซึ่ง COM Object ทุกตัวจะต้อง register เข้ากับ window ก่อน และข้อมูลต่างๆ ของ COM Object จะถูกเก็บอยู่ใน register ของ window เมื่อไคลเอนต์ ต้องการจะติดต่อกับ COM object ตัวโปรแกรมไคลเอนต์จะใช้ ClassID ในการอ้างอิงถึง COM Object นั้น ในการหา ClassID จะใช้ COM Runtime Library ทำหน้าที่ในการค้นหา เมื่อเจอแล้วก็จะสร้าง instance ของ COM Object จากนั้นจะรีเทิร์น (return) อินเตอร์เฟซพอยน์เตอร์ของออบเจกต์มาให้ ซึ่งไคลเอนต์จะใช้พอยน์เตอร์นี้ในการ call method ต่างๆ ที่มีอยู่ในออบเจกต์ได้



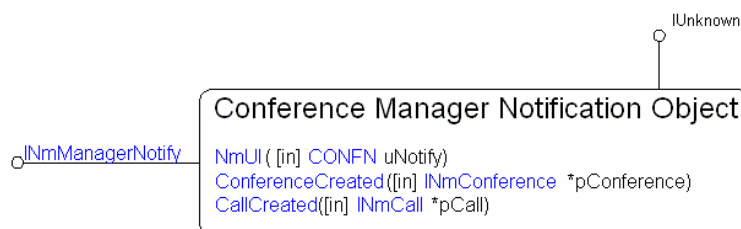
ตัวอย่างของ COM Object ที่มีการนำมาใช้งานนั้นมีดังต่อไปนี้

Conference Manager Object

ออบเจกต์นี้จะเป็นตัวหลักที่จะใช้ในการ

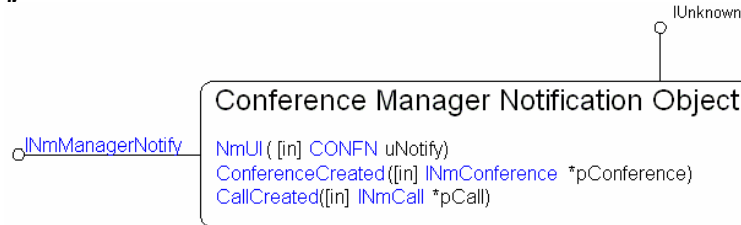
- สร้าง และคอยจัดการกับการเรียกเข้ามาของเครื่องอื่น ๆ (incoming call) และเรียกออกไปยังเครื่องอื่น (outgoing call)
- สร้าง และจัดการกับการใช้งานร่วมกันของผู้ใช้หลายๆ คน (conference)

Conference Manager Object นี้จะมี interface มาให้คือ INmManager ซึ่งจะเป็นตัวที่จะทำหน้าที่ในการเข้าถึงเมธอด (method) ต่างๆที่อยู่ในออบเจกต์

Conference Manager Notification Object

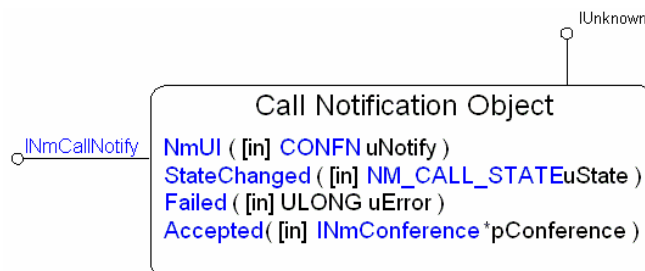
ออบเจกต์นี้จะเชื่อมต่อกับ Conference manager object เพื่อที่จะทำหน้าที่ในการรับอีเวนต์ (event) ต่างๆ ที่ Conference manager object ส่งมาให้ ซึ่งเราจะต้องมาทำการอิมพลีเมนต์ต่อไปว่าอีเวนต์ที่เกิดขึ้นนั้นเราจะจัดการกับอีเวนต์เหล่านั้นอย่างไร

Call Object



ออกแบบเจ็ทนี้จะมี interface มาให้คือ INmCall ซึ่งจะเป็นตัวที่จะใช้ในการเข้าถึงเมธอดต่างๆ ที่อยู่ในออกแบบเจ็ท เช่น คอยจัดการกับ incoming และ outgoing call โดยการบอกว่าจะ Accept หรือ Reject การ call ในแต่ละครั้ง

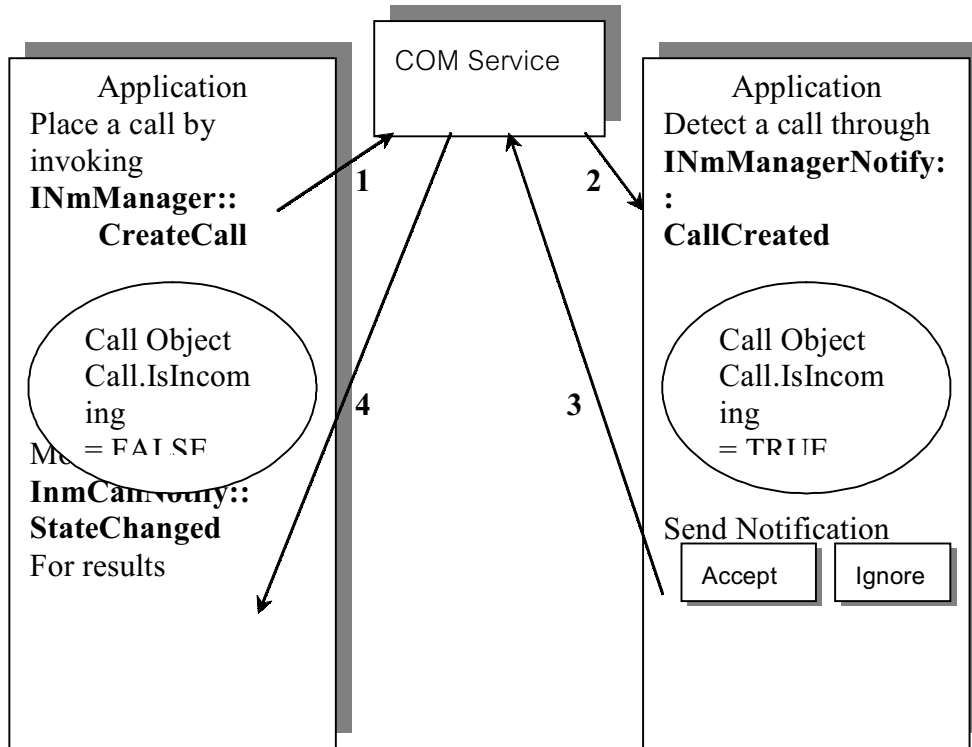
Call Notification Object



เราจะต้องทำการอิมพลีเมนต์ INmCallNotify เพราะว่าออกแบบเจ็ทตัวนี้จะรับอีเวนต์มาจาก Call object อย่างเช่น method INmCallNotify::StateChanged จะเป็นตัวที่บ่งบอกถึง state ต่าง ๆ ของการ call หรือว่าเมธอด INmCallNotify::Accepted จะเกิดขึ้นก็ต่อเมื่อตัว remote computer ได้ทำการ accept การ call แล้ว เมื่อเป็นเช่นนี้ทาง local computer จะเกิดอีเวนต์ขึ้น

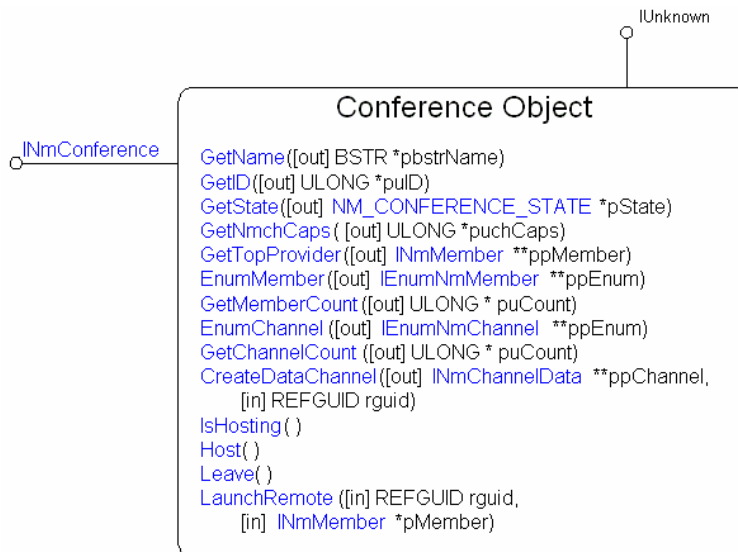
ในการ call แต่ละครั้งนั้นจะมีค่า default time-out อยู่ที่ 20 วินาทีสำหรับ incoming call แต่จะไม่มี time-out สำหรับ outgoing call

ส่วนขั้นตอนของการ call กันระหว่างแอปพลิเคชัน นั้นจะปรากฏอยู่ในรูปถัดไป ซึ่งจะแสดงให้เห็นถึงขั้นตอนต่างๆของการ call รวมถึงจะแสดงถึงเมธอดที่มีการนำมาใช้ด้วย



จากรูปจะเห็นได้ถึงขั้นตอนของการ call กันระหว่างแอปพลิเคชัน (application) ว่ามีขั้นตอนการทำอะไร และมีเมธอดไหนบ้างที่ต้องนำมาใช้งาน ซึ่งรูปที่ได้นำมาให้ดูนี้จะเป็นเพียงแผนภาพอย่างง่าย เพื่อที่จะให้เข้าใจถึงหลักการของการ call เท่านั้น แต่ในการอิมพลีเมนต์จริงแล้วจะต้องมีการตรวจเช็คค่าต่าง ๆ เพิ่มเติมมากกว่านี้อีก

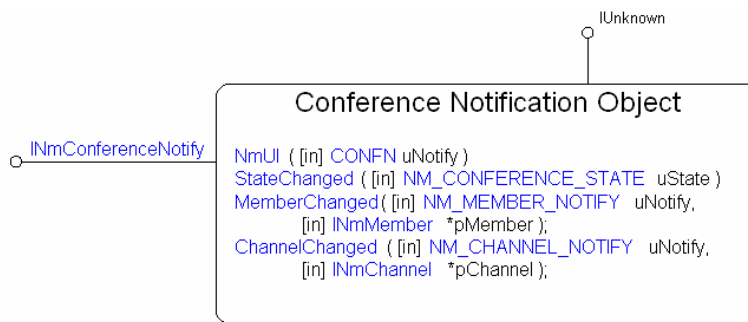
Conference Object



Conference object นี้จะมี interface มาให้คือ INmConference ซึ่งจะเอาไว้ใช้สำหรับการ เข้าถึง เมธอดต่างๆ ของออบเจกต์ ในการสร้าง Conference object นี้จะทำได้โดยการเรียกเมธอด InmManager::CreateConference เราสามารถที่จะสร้าง conference object ที่ตัวก็ได้ แต่ในขณะใดขณะ หนึ่งจะมี object เพียงตัวเดียวเท่านั้นที่สามารถเรียกใช้งานได้ ส่วนตัวอื่น ๆ จะต้องอยู่ในสภาวะ Idle

เมธอดโดยหลัก ๆ ของ conference object นี้ก็คือ INmConference::Host ซึ่งจะเป็นเมธอดที่จะ ทำให้ตัว local computer ทำหน้าที่เป็นโฮสต์ (host) เพื่อที่จะรองรับการ connect เข้ามาของ computer ตัวอื่น ๆ ได้

Conference Notification Object



ออบเจกต์ตัวนี้จะมี interface INmConferenceNotify มาให้ เพื่อที่จะใช้ในการอิมพลิเมนต์ โดย เมื่อมีอีเวนต์ที่เกิดขึ้นจาก Conference object แล้วอีเวนต์เหล่านั้นก็จะถูกส่งต่อมาให้ Conference Notification object เพื่อที่จะให้ผู้พัฒนาโปรแกรมสามารถที่จะเขียนโปรแกรมเพื่อดักอีเวนต์ได้

ขั้นตอนที่ 2 การออกแบบโปรแกรม

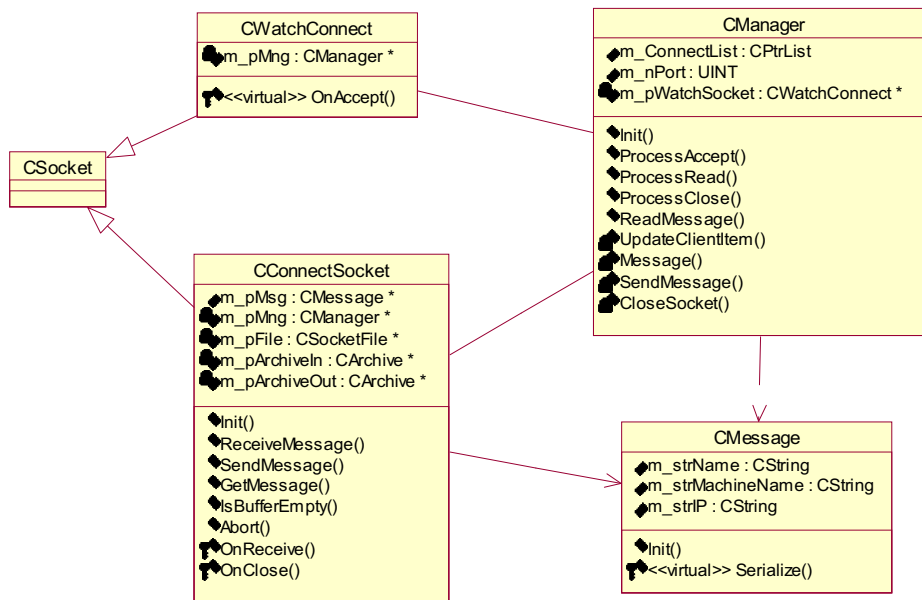
ในโครงงานวิจัยขั้นต้นได้มีการแบ่งโปรแกรมออกเป็น 2 ส่วนด้วยกันคือ

1. โปรแกรมส่วนของ Server
2. โปรแกรมส่วนของ Client

โปรแกรมในส่วนของเซิร์ฟเวอร์นั้น จะทำหน้าที่คล้าย ๆ กับการทำงานของ DNS (Domain Name System) คือ สามารถที่จะแปลงชื่อของผู้ใช้งานในขณะนั้น ๆ ให้เป็น IP address ได้ เพื่อที่จะให้ไคลเอนต์ สามารถสร้างการติดต่อถึงกันได้ เช่น เมื่อมีไคลเอนต์ตัวใหม่เข้ามา register กับตัว เซิร์ฟเวอร์ ซึ่งเซิร์ฟเวอร์จะทราบได้ทันทีว่าไคลเอนต์เครื่องนั้นชื่ออะไร และมี IP Address อะไร จากนั้นเซิร์ฟเวอร์จะเก็บ ข้อมูลเหล่านี้เอาไว้ แล้วก็ทำการส่งข้อมูลนี้ให้ไคลเอนต์ตัวอื่น ๆ ที่ได้ทำการ register ไว้ก่อนหน้านี้แล้ว ทำให้ไคลเอนต์ทุกตัวที่ register กับเซิร์ฟเวอร์จะทราบตลอดเวลาว่า ณ. ขณะใด ๆ มีไคลเอนต์เครื่องไหน บ้างที่สามารถติดต่อกันได้ และมีสถานะเป็นอย่างไร

ในส่วนของไคลเอนต์นั้น จะต้อง register ตัวเองเข้ากับเซิร์ฟเวอร์ก่อน เพื่อที่จะขอรับ ข้อมูลต่าง ๆ จากเซิร์ฟเวอร์ เช่น ต้องรู้ว่าในขณะใด ๆ มีเครื่องไหนพร้อมที่จะติดต่อได้บ้าง เมื่อได้รับ ข้อมูลแล้ว จากนั้นไคลเอนต์สามารถที่จะติดต่อไปยังไคลเอนต์เครื่องอื่น ๆ เอง โดยไม่ต้องผ่านเซิร์ฟเวอร์

ในการออกแบบโปรแกรมนั้นจะใช้ UML(Unified Modeling Language) ซึ่งจะเหมาะสำหรับการออกแบบโปรแกรมแบบ Object - Oriented โดย Class diagram ที่เขียนขึ้นมานั้น จะทำให้เห็นถึงโครงสร้างของโปรแกรมทั้งหมดว่าประกอบขึ้นด้วย Class อะไรบ้าง มี Interface อะไรบ้าง และแต่ในแต่ละ Class มีความสัมพันธ์กันอย่างไร การสร้าง Class Diagram นี้จะเป็นส่วนพื้นฐาน เพื่อที่จะนำไปใช้ในการสร้าง Diagram อื่น ๆ ต่อไป

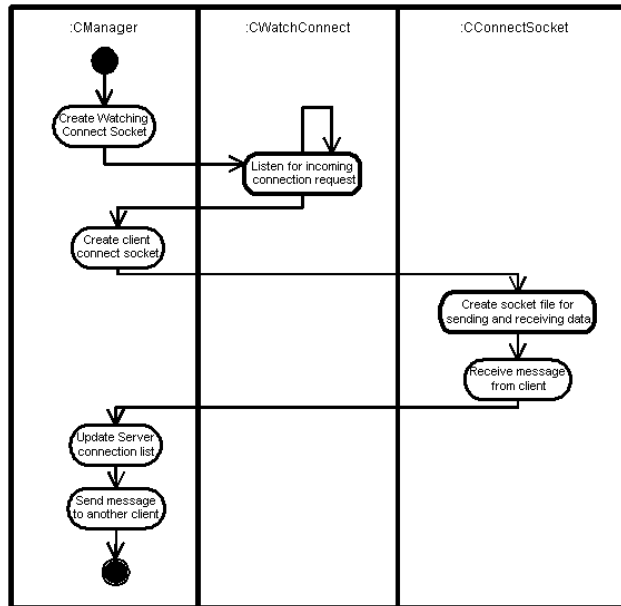


Server Class Diagram

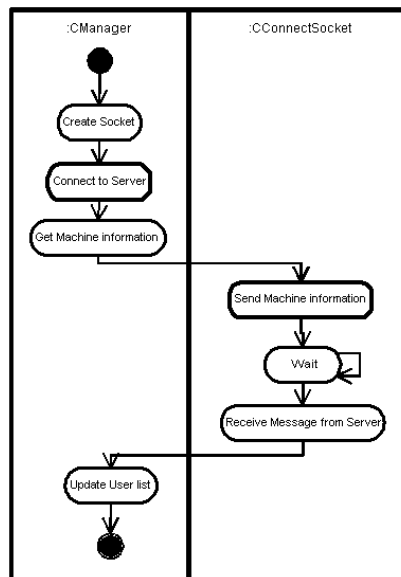
จากรูปจะแสดงถึงตัวอย่างของคลาสและเมธอดต่างๆ ของเซิร์ฟเวอร์ และไคลเอนต์ โดย

- CManager คลาสนี้จะทำหน้าที่เป็นคลาสหลักของโปรแกรม ซึ่งจะใช้ในการ Initial Socket และคอยควบคุมการทำงานทั้งหมดของโปรแกรม
- CWatchConnect คลาสนี้จะทำหน้าที่เป็นตัวที่คอยรับการ Connect มาจาก Client
- CConnectSocket คลาสนี้จะทำหน้าที่ในการติดต่อสื่อสารกับ Client การที่ Server กับ Client แต่ละตัวจะติดต่อส่งข้อมูลกันได้ ก็จะใช้คลาสนี้เป็นหลักในการรับ-ส่งข้อมูลระหว่างกัน
- Cmessage Class นี้จะเอาไว้สำหรับการเก็บ Information ต่าง ๆ รวมไปถึงการทำ Serialization ด้วย

จาก Class Diagram ที่ได้แสดงให้เห็นแล้วข้างต้นนั้นทำให้เห็นถึงการออกแบบโปรแกรมว่าจะประกอบไปด้วยคลาสอะไรบ้าง ต่อไปจะแสดงตัวอย่างของการออกแบบ Activity Diagram เพื่อให้เห็นถึงกระบวนการทำงาน และความสัมพันธ์กันระหว่างออบเจกต์

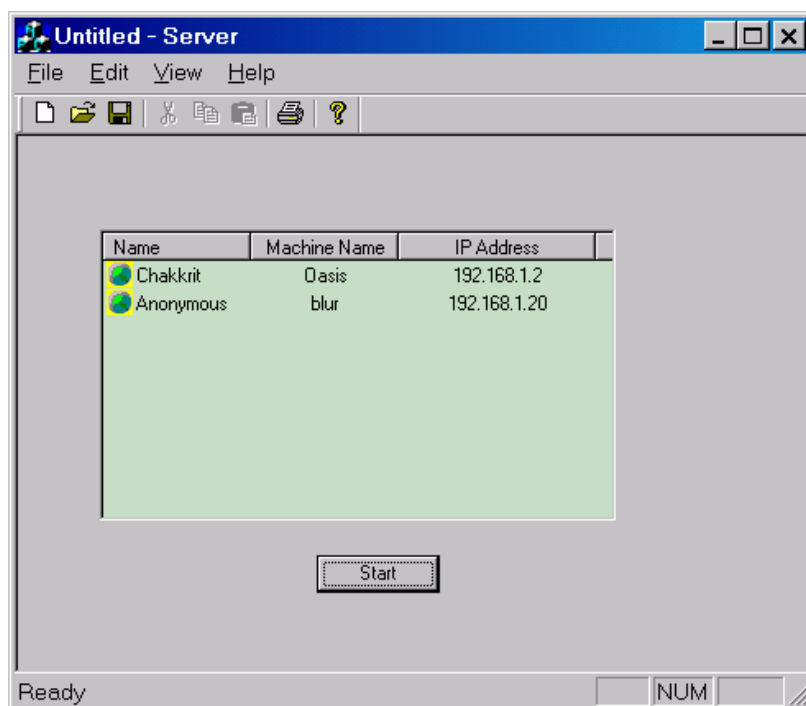


Server Activity Diagram



Client Connect to Server

ขั้นตอนที่ 3 การทดสอบโปรแกรมที่ได้พัฒนาขึ้นมา โปรแกรมส่วนของ Server

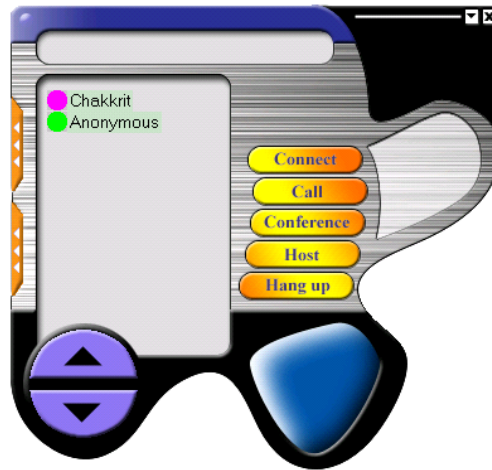


โปรแกรมในส่วนของเซิร์ฟเวอร์นี้จะทำหน้าที่ในการรับการติดต่อ (connection) มาจาก ไคลเอนต์ โดยที่เซิร์ฟเวอร์จะเก็บ information ต่าง ๆ ของไคลเอนต์เอาไว้ เมื่อมีไคลเอนต์ตัวใหม่ connect เข้ามาในระบบ ตัวเซิร์ฟเวอร์จะทำการส่ง information ของไคลเอนต์ที่เข้ามาก่อนหน้านี้ไปให้กับไคลเอนต์ที่เพิ่ง login เข้ามาใหม่โดยอัตโนมัติ นอกจากนั้นยังจะต้องส่ง information ของไคลเอนต์ตัวใหม่ ไปให้กับไคลเอนต์ตัวอื่นๆ ที่เข้ามาก่อนหน้านี้ด้วย

ตัวเซิร์ฟเวอร์จะเก็บ ชื่อของผู้ที่เข้ามาใช้งาน, ชื่อเครื่อง และ IP address ของ Client แต่ละตัวเอาไว้ จากนั้นก็จะนำมาเก็บเป็นออบเจกต์ ซึ่งจะใช้ลิงก์ลิสต์ (Link list) ในการเก็บออบเจกต์เอาไว้ เพื่อที่จะให้ง่ายต่อการจัดเก็บ และการค้นหา

ในการทดสอบส่วนนี้ปรากฏว่า โปรแกรมสามารถทำงานได้อย่างถูกต้อง สามารถที่จะรองรับการติดต่อ (connect) จากไคลเอนต์ได้อย่างไม่จำกัด อีกทั้งยังสามารถรับการติดต่อจากไคลเอนต์ที่เข้ามาพร้อมๆ กันได้ด้วย เนื่องจากว่าภายในตัวโปรแกรมนั้นได้จัดระบบการเข้าคิว (Queue) เอาไว้ให้

โปรแกรมส่วนของ Client



ส่วนนี้จะเป็นหน้าจอหลักของโปรแกรมในส่วนของไคลเอนต์ซึ่งได้มีการนำเอาบิตแมพ (Bitmap) มาทำเป็น skin เพื่อที่จะให้เกิดความสวยงามและนำใช้งานมากขึ้น โดยที่จะมีปุ่มต่าง ๆ ดังนี้

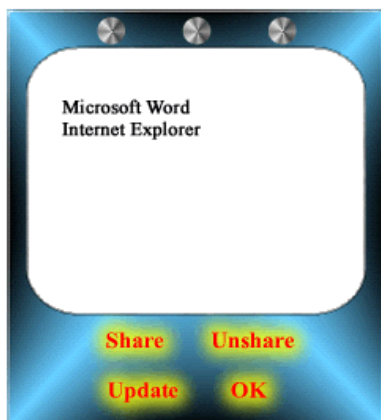
-Connect เป็นปุ่มที่ไว้ใช้สำหรับการที่จะให้ไคลเอนต์ทำการติดต่อสื่อสารกับเซิร์ฟเวอร์ ได้ ทุกครั้งที่รันโปรแกรมจะต้องใช้ปุ่มนี้เสมอ

-Call เป็นปุ่มที่ไว้ใช้สำหรับการ Call กันระหว่างไคลเอนต์ด้วยกันเอง แต่การ Call นี้ จะเป็นการติดต่อกันระหว่างเครื่อง 2 เครื่องเท่านั้น

-Conference เป็นปุ่มที่ไว้ใช้สำหรับการ Call กันในแบบ Conference คือสามารถที่จะติดต่อสื่อสารกันได้หลายเครื่อง

-Host เป็นปุ่มที่ไว้ใช้สำหรับการที่จะให้ไคลเอนต์ตัวใดตัวหนึ่งทำหน้าที่เป็นโฮสต์ เพื่อที่จะให้ไคลเอนต์ตัวอื่น ๆ Call แบบ Conference เข้ามาได้

-Hang up เป็นปุ่มที่ไว้ใช้สำหรับการยกเลิกการ Call



ภาพด้านซ้ายมือ เป็นส่วนที่จะใช้สำหรับการ Share Application โดยจะมี Application ให้เลือกกว่า ณ.ขณะใดขณะหนึ่ง มี application ตัวใดบ้างที่สามารถที่จะ Share หน้าจอให้กับไคลเอนต์ตัวอื่นได้

- ปุ่ม Share จะใช้เมื่อต้องการที่จะแชร์หน้าจอให้
- ปุ่ม Unshare เอาไว้สำหรับยกเลิกการแชร์
- ปุ่ม Update ไว้สำหรับ update เมื่อมีการรัน application ใหม่ขึ้นมา

การทดสอบโปรแกรมนั้นจะใช้เครื่องคอมพิวเตอร์ของผู้สอนทำหน้าที่ในการแชร์หน้าจอของโปรแกรมต่างๆ โดยเครื่องคอมพิวเตอร์ของผู้เรียนจะต้อง call มาที่เครื่องของผู้สอนก่อน จากนั้นหน้าจอของเครื่องผู้สอน จะไปปรากฏอยู่ที่เครื่องของผู้เรียน ในการทดสอบการส่งหน้าจอจากเครื่องหนึ่งไปสู่อีกเครื่องหนึ่งก็สามารถทำได้ดีมาก เนื่องจากสามารถส่งได้แบบ Real time ยกตัวอย่างเช่น เมื่อมีการขยับเมาส์ (mouse) ที่เครื่องของผู้สอน ตำแหน่งของเมาส์ที่ขยับไปนั้นจะสามารถไปปรากฏในเครื่องของผู้เรียนได้ทันที หรือ เมื่อมีการรันโปรแกรมอื่นๆ ขึ้นมา โปรแกรมเหล่านั้นจะปรากฏที่เครื่องผู้เรียนได้ทันทีเหมือนกัน

ในกรณีผู้สอน และผู้เรียนไม่ได้อยู่ในห้องเดียวกัน ก็สามารถที่จะทำการสอนได้โดยตัวโปรแกรมนี้จะทำหน้าที่ในการส่งทั้งหน้าจอ และเสียงของผู้สอนไปพร้อมๆ กัน จากการทดสอบเสียงที่ส่งออกไปนั้นอยู่ในขั้นที่ค่อนข้างน่าพอใจ นอกจากนั้นแล้วผู้เรียนยังสามารถพูดคุยกับผู้สอนได้แบบ Real Time

สรุป

ในงานวิจัยชิ้นนี้ได้นำข้อดีของการออกแบบและพัฒนาโปรแกรมแบบ Object-Oriented ซึ่งทำให้การพัฒนาสามารถทำได้รวดเร็ว สะดวกต่อการเพิ่มเติมหรือการแก้ไขโปรแกรม โดยมีการนำ COM Object มาประยุกต์ใช้งาน เพื่อช่วยให้การเรียนการสอนภายในห้องปฏิบัติการคอมพิวเตอร์มีประสิทธิภาพมากขึ้น เพราะผู้สอนสามารถสาธิตการใช้งานให้ผู้เรียนทราบได้อย่างทั่วถึง และยังทำให้ทั้งผู้เรียนและผู้สอนสามารถโต้ตอบกันได้ทันทีที่มีข้อสงสัยหรือไม่เข้าใจ โดยการใช้ทั้งภาพและเสียงไปพร้อมๆ กัน ทำให้ผู้เรียนไม่ต้องเสียเวลามากกับการอ่านและทำความเข้าใจด้วยตัวเอง นอกจากนั้นแล้วยังเป็นประโยชน์กับผู้สอนที่สามารถประเมินความสามารถของผู้เรียนได้ทันทีว่าเข้าใจในบทเรียนมากน้อยเพียงใด ทำให้ผู้สอนสามารถปรับเปลี่ยนวิธีการสอนเพื่อให้เหมาะสมกับผู้เรียนได้อย่างทันท่วงที

ข้อเสนอแนะ

- จากการทดสอบโปรแกรมที่ได้มีการพัฒนาขึ้นมา นั้น มีข้อเสนอแนะสำหรับการวิจัยต่อไปดังต่อไปนี้
1. ในการส่งข้อมูลภาพและเสียงนั้น ควรจะต้องมีการใช้เทคโนโลยีในการบีบอัดข้อมูล เป็นการลด traffic. ในเครือข่าย เพื่อให้สามารถส่งข้อมูลใน WAN ได้
 2. ควรจะนำเทคโนโลยี QoS (Quality of Services) มาใช้ เพื่อให้การส่งข้อมูลเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น

บรรณานุกรม

พิพัฒน์ หิรัญยวณิชชากร และ ศักดิ์ชัย เอี่ยมกิจสัมฤทธิ์. 2544. โปรแกรมมัลติมีเดียช่วยในการเรียนการสอนซอฟต์แวร์ภายใต้วินโดวส์. พัฒนบริหารศาสตร์ ปีที่ 41 ฉบับที่ 2/2544.

Craig Larman. 2002. Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and the Unified Process 2nd Edition Prentice Hall Publisher.

Glenn Maughan, Raphael Simon. 2000. Windows Programming Made Easy : Using Object Technology, COM, and the Windows Eiffel Library. Prentice Hall Publisher.

Microsoft Corporation. <http://msdn.microsoft.com> [2003]

Santiago Comella-Dorda. Component Object Model (COM), DCOM, and Related Capabilities.

Sara Williams, Charlie Kindel. 1994. The Component Object Model : A Technical Overview.

URL : http://www.sei.cmu.edu/str/descriptions/com_body.html [2003]

URL : http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncomg/html/msdn_comppr.asp [2003]

W. Richard Stevens. 1994. TCP/IP Illustrated Volume 1. Addison-Wesley Publisher.
