

# 國立交通大學

## 多媒體工程研究所

### 碩士論文

多人線上即時會議多媒體錄製工具設計及製作



The Design and Implementation of a Multimedia Recording

Tool for the Online Net Meeting Application

研究生：謝豎鏞

指導教授：陳登吉 教授

中華民國九十七年七月

# 多人線上即時會議多媒體錄製工具設計及製作

## The Design and Implement of a Multimedia Recording Tool for the Online Net Meeting Application

研究生：謝豎鏘

Student : Shu-Kai Hsieh

指導教授：陳登吉

Advisor : Dr. Deng-Jyi Chen



Submitted to Institute of Multimedia Engineering  
College of Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in  
Computer Science  
July 2008  
Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

# 多人線上即時會議 多媒體錄製工具設計及製作

學生：謝豎鏜

指導教授：陳登吉 博士

國立交通大學多媒體工程所碩士班

## 摘要

由於現有的多人線上即時會議系統，在會議檔案呈現部分或是多媒體錄製工具方法，都各有些許的缺失，導致錄製會議過程的多媒體錄製檔不夠完整，在播放時無法重現會議當時的情形。

因此，本研究將實作出一個線上多人會議錄製軟體，讓在距離遙遠的同事，透過網路進行線上會議、暢所欲言，並讓進行會議的同事可以專注在同一個畫面上，進行即時的討論與溝通。且進一步能夠將會議的討論過程全部錄製下來，成為會議記錄。

此系統將取用智勝國際公司所開發的多媒體錄製工具 - 講解手進行開發，由於講解手原本只是單純的多媒體錄製工具，並沒有線上即時會議的功能。因此本研究將在講解手的架構上，建構線上多人會議系統的功能。並取用 Skype API 處理多人線上語音通話的部分。

本研究實作出的多人線上會議系統，附屬多媒體錄製工具之功能，此系統能將 PowerPoint 檔解析後，在會議過程中將使用者的簡報檔流程、聲音流程、簡報筆流程分別記錄下來儲存為一多媒體錄製檔，此檔案大小較一般直接擷取螢幕畫面的軟體更小。在會議結束後，還可將多媒體錄製檔上傳至 Web 2.0 Blog 平台，供他人瀏覽。

# The Design and Implement of a Multimedia Recording Tool for the Online Net Meeting Application

Student: *Shu-Kai Hsieh*

Advisor: *Dr. Deng-Jyi Chen*

**Department of Multimedia Engineering  
National Chiao Tung University**

## Abstract

The mature of network infrastructure and technologies has made many applications under the ubiquitous networking environment possible. People can use internet to create their business even they are not in their office. Today, we can find that a business team uses net meeting application to discuss the project status or to form a new project without actually meeting with each other in the same room. The demand of such an online net meeting system becomes urgent.

In this research, we will design and implement a multimedia recording tool for the online net meeting application. The net meeting member from different places can perform a meeting application on the same screen to discuss and communicate online using PowerPoint slices. Unlike existing net meeting systems demand, a large amount of storage space to store the recorded screen captured slices. The proposed system takes the advantages of combing the PowerPoint slice, audio file, and script file together to produce a compressed file and to produce a related small size recorded file.

Skype API is employed to combine the audio file from different parties involved during the net meeting time. The recorded meeting minutes will be uploaded to a blog once the net meeting is over. A real net meeting example is exercised to demonstrate the application of the proposed tool.

# 誌謝

感謝陳登吉老師，在我碩士兩年的生涯裡，引領著我找到一個令我感興趣的研究方向。並且時時督促著我，讓我的研究進度始終保持的很好。當我有疑惑時，也是老師提供我一盞明燈，讓我依循。在此感謝老師這兩年來的栽培。

感謝實驗室的伙伴們，沒有我就沒有你們。這兩年過得很充實，除了課堂上的學習外，還有課後的交流，以及每個禮拜的運動生活，讓我保有強健的體魄，完成我的課業以及研究。還有學妹乃宣與明彥在我口試時的鼎力相助，謝啦，各位。



# 目錄

摘要.....	i
Abstract.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vi
圖目錄.....	vii
一、緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究方法與步驟.....	1
1.3 專有名詞及研究成果的限制.....	2
1.4 章節概述.....	3
二、相關研究與探討.....	4
2.1 多人線上會議多媒體錄製工具.....	4
2.1.1 多人線上會議系統.....	4
2.1.2 多媒體錄製工具.....	5
2.2 目前現有多人線上會議系統的比較.....	6
2.3 使用工具介紹.....	8
2.3.1 講解手.....	8
2.3.2 Skype API.....	9
2.4 小結.....	10
三、系統需求分析.....	11
3.1 使用者連線模式需求.....	11
3.2 使用者間溝通信息需求.....	12
3.3 會議進行方式需求.....	13
3.4 會議同步需求.....	14
3.4.1 畫面同步需求.....	14
3.4.2 聲音同步需求.....	15
3.5 系統操作流程.....	16
3.5.1 會議前置作業.....	17
3.5.2 會議進行中.....	18
3.5.3 會議結束.....	19
四、系統架構與實作.....	20
4.1 系統架構.....	20
4.1.1 系統示意圖.....	21

4.2 講解手擴充功能設計與實作.....	21
4.2.1 使用Skype4Com.....	23
4.3 中介伺服器設計與實作.....	24
4.4 自訂封包格式設計與實作.....	27
4.4.1 使用者與中介伺服器之間溝通的封包制定.....	29
4.4.2 使用者與中介伺服器之間溝通的封包範例.....	31
4.4.3 使用者與使用者之間溝通的封包制定.....	31
4.4.4 使用者與使用者之間溝通的封包範例.....	34
4.4.5 自訂封包格式列表.....	35
4.5 多媒體錄製工具之實作.....	37
五、系統展示.....	41
5.1 會議的前置作業.....	41
5.1.1 進入會議室前.....	41
5.1.2 進入會議室後.....	44
5.2 會議進行中.....	46
5.3 會議結束.....	48
5.3.1 發佈多媒體錄製檔.....	48
5.3.2 觀看多媒體錄製檔.....	50
六、結論.....	52
6.1 總結.....	52
6.2 未來發展方向.....	52
參考資料.....	53



# 表目錄

表格 2-1 線上會議系統多媒體錄製工具比較表 .....	7
表格 4-1 使用者與伺服器溝通的Command列表.....	36
表格 4-2 使用者與使用者溝通的Command列表.....	36





# 圖目錄

圖 2-1 多人線上會議系統示意圖 .....	4
圖 2-2 講解手的使用介面 .....	8
圖 2-3 Skype API示意圖(Windows Message) .....	9
圖 2-4 Skype API示意圖(Skype4Com).....	10
圖 3-1 連線模式 – 不需中介伺服器 .....	11
圖 3-2 連線模式 – 需中介伺服器 .....	12
圖 3-3 溝通訊息封包格式 .....	13
圖 3-4 會議進行畫面同步內容 .....	15
圖 3-5 會議前置作業操作流程圖 .....	17
圖 3-6 會議進行中操作流程圖 .....	18
圖 3-7 會議結束作業操作流程圖 .....	19
圖 4-1 系統架構圖 .....	20
圖 4-2 系統示意圖 .....	21
圖 4-3 講解手擴充功能示意圖 .....	22
圖 4-4 中介伺服器處理示意圖 .....	25
圖 4-5 中介伺服器的UML Class Diagram.....	25
圖 4-6 接收封包示意圖 .....	28
圖 4-7 使用者與伺服器溝通封包格式 .....	30
圖 4-8 Command Number的實作程式碼 .....	30
圖 4-9 使用者與伺服器溝通封包範例 .....	31
圖 4-10 傳給指定使用者的封包格式 .....	33
圖 4-11 傳給會議室中其他使用者的封包格式 .....	33
圖 4-12 使用者與使用者溝通的封包範例 .....	35
圖 4-13 聲音檔混音示意圖 .....	39
圖 4-14 DirectShow的TimeLine模組示意圖 .....	39
圖 4-15 講解手使用DirectShow的TimeLine模組實作混音 .....	40
圖 5-1 講解手介面 .....	41
圖 5-2 啟動Skype的訊息視窗 .....	42
圖 5-3 Skype視窗介面 .....	42
圖 5-4 Skype API驗證保護機制 .....	42
圖 5-5 線上會議前置準備畫面 .....	43
圖 5-6 主席開啟會議室 .....	43
圖 5-7 使用者選擇會議室加入 .....	44
圖 5-8 會議參與者加入會議室 .....	45

圖 5-9 使用者加入會議室 .....	45
圖 5-10 聊天視窗介面 .....	46
圖 5-11 會議進行時的畫面 .....	46
圖 5-12 核准/禁止發言權 .....	47
圖 5-13 講解筆工具 .....	47
圖 5-14 錄製畫面 .....	48
圖 5-15 會議結束後畫面 .....	48
圖 5-16 WebBlog使用者驗證畫面 .....	49
圖 5-17 發佈設定 .....	49
圖 5-18 發佈選擇主題 .....	50
圖 5-19 Web Blog發佈網頁 .....	50
圖 5-20 講解檔網頁畫面 .....	50
圖 5-21 講解手播程式畫面 .....	51



# 一、緒論

## 1.1 研究背景與動機

在這個資訊便利的時代，網路科技愈來愈發達，全球資訊流通的方式，讓整個世界變得跟地球村一樣零距離。網路科技從一開始的撥接上網、ADSL、到現在的無線網路，普及的程度讓大家隨時隨地都能夠使用到網路科技所帶來的資訊便利。

現在更有電信業者強調出「不在辦公室，也能辦公事」的方案，讓在各地的人才們，就算人不在辦公室中，到外地出差，也能夠透過網路的技術，享受到出門在外也能辦公事的服務。這個方案非常符合一般上班族出門辦公的訴求。一般常見的案例，在企業與學校裡，常常由於主管與老師的出差，無法定時聆聽職員、學生的報告。所以必須藉助於線上會議的軟體，來進行會議的討論。因此，線上會議的需求將提升，使用率也會愈來愈普及。

目前現有的線上多人會議系統，大部分缺乏錄製功能，如：NetMeeting、Gogrok[24]、Co-Life[25]…等等。這些系統只能進行線上會議，卻無法錄製會議的過程。而其他擁有會議錄製功能的線上會議系統，目前都各有些許的缺失，導致錄製的多媒體檔案不夠完善，如：多媒體錄製檔案太大、無法完整重現會議當時情形…等等。

因此本研究將實作出一個線上多人會議錄製軟體，讓在距離遙遠的同事，透過網路進行線上會議、暢所欲言，讓參與會議的同事可以專注在同一個畫面上，進行即時的討論與溝通，並可兼具使用時的流暢度。將會議的討論過程全部錄製下來，成為會議記錄。

## 1.2 研究方法與步驟

為了開發出一套多人線上即時會議的系統，本研究將分為以下幾個步驟來進行：

1. 分析目前已有的線上即時會議系統的優缺點。
2. 找出一套不夠完善的系統來繼續開發。
3. 搜集系統相關的公開 Library 來使用。
4. 了解舊有系統的程式與架構。
5. 系統需求分析：  
分析系統的需求，包括如何達到畫面的同步、聲音的同步、流程的同步…等。考慮使用者的操作需求，規畫出一套使用者流程。
6. 與已有的系統整合的規劃。
7. 系統實作：  
將分析過後的需求進行實作，並測試，檢查有無使用上的問題。
8. 總結論文結論，以及未來的發展方向。

## 1.3 專有名詞及研究成果的限制

本研究的相關名詞解釋及領域，述途如下：

- ◆ **Skype API [1][13][14] :**  
使用 Skype 提供的 API 來控制 Skype 的主程式以及功能。可以用此功能來讓網路雙方的使用者進行網路電話的對話。
- ◆ **DirectX [2][4] :**  
Windows SDK 的工具，提供 COM 的介面處理畫面、影音多媒體的控制。本研究將使用其中一個模組 - DirectShow。
- ◆ **Borland C++ Builder [5][6] :**  
Borland 公司的 Windows GUI 開發工具，程式語言使用 C/C++。本研究將使用它來當開發系統的工具。
- ◆ **OSI( Open System Interconnection ) [3] :**  
網路通訊協定，將網路通訊的工作分為七層處理。此為一個虛擬的網路通訊模型。
- ◆ **TCP/IP [3] :**  
TCP(Transmission Control Protocol)，網路通訊協定，提供高可靠度的資料傳輸。本研究將用此通訊協定來傳輸資料。
- ◆ **講解手[11] :**  
智勝國際科技公司[7]開發的多媒體錄製工具。多應用於數位教材製作方面。
- ◆ 本研究所研發的軟體其應用限制，僅能以Microsoft PowerPoint(PPT)檔

案為會議討論的簡報檔案，尚無法應用於其他格式的檔案，如：PDF、Open Office…等等。

## 1.4 章節概述

本論文的章節安排如下：

1. 本章先提出本研究的動機與目的，探討目前的線上會議系統的缺失，以及本研究的方法與步驟。
2. 在第二章中，會探討研究的相關議題，比較目前幾家較有名的線上會議系統的優缺點，本研究將借鏡這些系統的優缺點，提出未來開發改進重點。
3. 第三章為系統分析，將一一分析出線上會議系統所需要的模組，以及這些模組該如何去設計及實作。
4. 第四章為系統架構與實作，將說明系統如何依照需求分析的結果，進行實作。各模組之間的實作細節將一一說明。
5. 第五章為系統展示，此章節將以圖例的方式展示開發出的線上會議系統。
6. 第六章會對本論文做出結論，包括本系統未來展望，提出一些新的方向。

## 二、相關研究與探討

本章會對目前已開發完成的「線上會議系統」做相關的探討。第一節將討論一個基本的線上會議系統所需要的模組，以及架構流程。第二節將比較各家之間的優缺點，而本系統將取其中一家之系統繼續開發。第三節將介紹未來開發時可應用的工具。

### 2.1 多人線上會議多媒體錄製工具

#### 2.1.1 多人線上會議系統

這節會探討多人線上會議系統所需要的功能，以下就針對「多人線上會議系統」這個名詞來做定義。多人線上會議系統，就是很多個人在不同的地點，遠距離的地點，透過網路連線，進行即時的語音會議對談，同時也可以傳送視訊畫面或是螢幕畫面，讓會議進行時更有真實感。

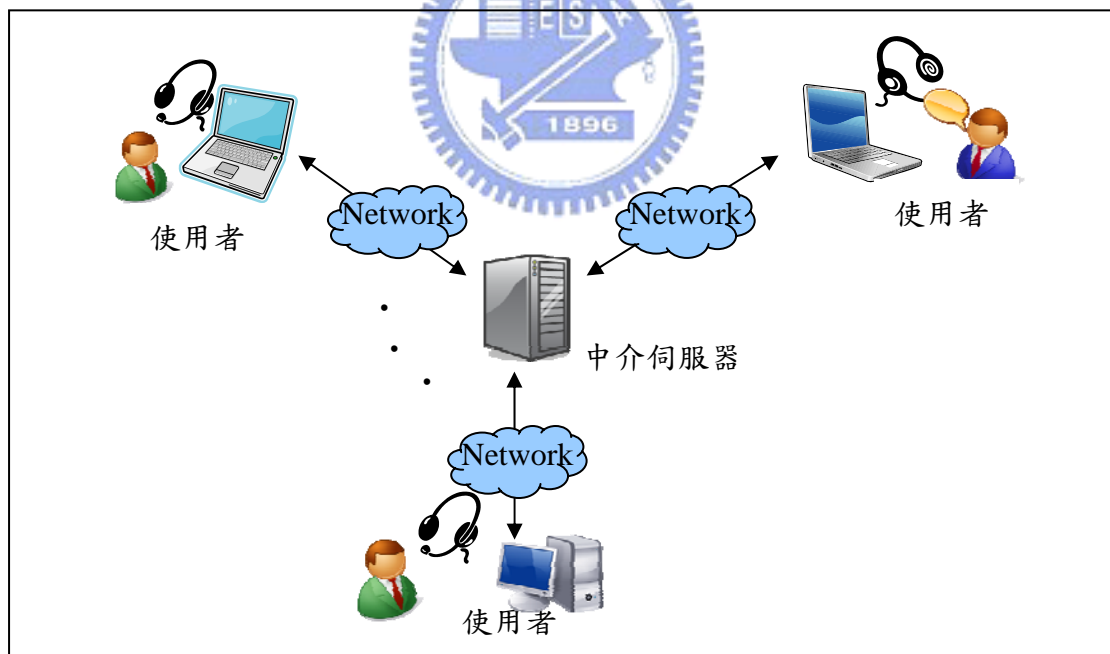


圖 2-1 多人線上會議系統示意圖

由上圖可知多人線上會議系統為何，多名使用者透過網路連線至中介伺服器，再與其他使用者溝通。這種方法是目前大部分的線上會議、線上通訊軟體所採用的方案。很清楚的可以看到，只要使用者擁有一部電腦及螢幕，該部電腦擁有麥克風與耳機的功能，加上網路連線的功能，就可以進行線上會議的討論。



會議進行時，每個人進行溝通的要點為：(1)聲音 (2)畫面，在會議進行時，每個人都必須聽到同樣的談話內容，看到同樣的畫面，這樣子才會有進行會議的感覺。

而每個使用者所看到的螢幕、畫面內容包括兩部分：(1)視訊 (2)單一檔案，第一部分就是視訊的內容，也就是透過網路攝影機拍攝使用者的畫面傳輸過去。第二部分就是單一檔案，即會議流程所需要的講解檔案，包括投影片(PPT)、PDF或是影片檔案，這部分即為引導整個會議流程的內容，讓每一位會議參與者都能順著這個檔案了解會議討論的議題。

因此一個好的線上會議系統，最重要的就是讓所有參與會議者，能夠看到相同的討論畫面，聽到每個人討論的聲音，會議中間所需要的網路頻寬是愈少愈好，這樣子使用者所需要的硬體設備門檻就可以降低。

## 2.1.2 多媒體錄製工具

上一小節說明的多人線上會議系統，可以利用多媒體錄製工具，將整個會議過程錄製下來，存成一個多媒體記錄檔案，可供使用者做其他用途，包括作為多媒體教材、儲存會議記錄…等等。

錄製的內容包括會議進行時的兩大部分：(1)聲音 (2)畫面，多媒體錄製工具必須將這兩部分的內容記錄下來，讓使用者在播放錄製檔案時，可以達到重現會議進行時的討論情形。而最重要的就是這兩大部分必須要同步，否則會出現「聲音跟不上畫面」或是「畫面跟不上聲音」，這種情形發生。這種情形一發生，就無法重現當時會議進行的討論實況。

一般來說這種會議用的多媒體錄製工具可分為兩種：(1)獨立於會議軟體 (2)會議軟體的附屬功能。以下將對這兩種方式進行解釋與分析：

### 1. 獨立於會議軟體：

通常一般的會議軟體不附有錄製功能，因此使用者想將會議過程記錄下來，必須使用其他的錄製工具將會議過程錄製下來。這種方式，一般而言都是針對畫面進行錄製，錄製效果通常取決於影片編碼的格式，如：MPEG-3，MPEG-4 [18]…等。愈好的錄製軟體，加上好的編碼格式，可以讓錄製出來的結果很好，而且檔案大小不會太大。

缺點：錄製效果不一，錄製結果的影片檔案較大，影片檔不具文件保護功能

優點：可用一般的影片播放軟體播放。

### 2. 會議軟體的附屬功能：

並非所有的會議軟體都包含此功能。有此功能的會議軟體，通常錄製下

來的檔案，都是自行定義格式的檔案。用此方法，可以將錄製結果檔，進行加密，達到文件保護的功能。

缺點：錄製的檔案，必須用自行開發的播放軟體才能播放。

優點：可對錄製的檔案進行加密，達到文件保護的功能。

基於一般使用者都是因為公司行號學校的需求，所以文件保護的機制是有其必要性，所以多媒體錄製工具會以「會議軟體的附屬功能」為較好的選擇。

## 2.2 目前現有多人線上會議系統的比較

考量目前主要用於會議的簡報檔案，大多以 Microsoft 的 PowerPoint 檔(PPT 檔)為主流。在進行會議時，以 PPT 檔為會議的主軸，讓會議參與者們，能夠依此 PPT 檔進行相關的討論。因此本研究也將依大部分使用者的需求，將 PPT 檔作為本研究的簡報檔案格式。

此小節將舉出幾家有開發多人線上會議系統的軟體出來作比較。比較彼此之間的優缺點在哪裡。比較的重點在於：功能、支援 PPT 檔格式，以及錄製的檔案大小。

### 比較的系統：

1. 講解手：智勝國際開發的講解手(SoEzLecturing)，主要的目的是用來錄製多媒體教材的工具。目前的版本只能用來錄製PPT檔。
2. Skype：時下熱門的VOIP網路電話軟體，目前有支援線上會議的功能。
3. Netmeeting：Microsoft Windows OS內建的線上會議軟體。
4. MultiVideo[9]：毅上科技開發的線上會議軟體。
5. WebEx[10]：Cisco公司開發的線上會議軟體。

### 比較功能解釋：

1. 會議功能：該軟體有進行會議的功能(僅限於單機進行)。
2. 線上會議功能：該軟體可讓多人進行線上會議功能。
3. 線上共同觀看PPT功能：該軟體在進行線上會議的同時，可讓多人同時觀看同一個PPT檔(投影片檔)。
4. 錄製功能：該軟體有進行會議錄製的功能(僅限於單機錄製)。
5. 處理PPT檔特效：該軟體能夠完整呈現PPT檔的特效功能
6. 處理PPT檔聲音：該軟體能處理嵌入在PPT檔中的聲音檔(以wma檔進行測試)
7. 處理PPT檔影片：該軟體能處理嵌入在PPT檔中的影片檔(以wmv檔進行測



試)

8. 錄製檔案(共 5min.)：該軟體有錄製功能，並錄製一份 5 分鐘的 PPT 檔所需要的檔案空間大小。
9. 錄製檔案(以 1hr.來計算)：以上一個功能，錄製 5 分鐘的檔案大小計算 1 小時錄製後，檔案大小的空間。

表格 2-1 線上會議系統多媒體錄製工具比較表

軟體 功能	講解手 3.1	Skype 3.6	Netmeeting 3.01	MultiVideo 5.0	WebEx 8.0
會議功能	●	●	●	●	●
線上會議功能		●	●	●	●
線上共同觀看 PPT	*註 1		●	●	●
處理 PPT 特效	●				●
處理 PPT 聲音	●				
處理 PPT 影片	●				
錄製功能	●			●	●
錄製檔案 (共 5min.)	1779 KB			1920 KB	6824 KB
錄製檔案 (以 1hr.來計算)	20.5 MB			22.5 MB	81.88 MB

● - 表示有此功能

註 1：講解手具有處理 PPT 檔的功能，但是並沒有「線上共同觀看 PPT」的功能

結果分析：

1. 講解手、Skype、Netmeeting 在線上會議及錄製功能方面，都有不足之處。
2. 講解手、MultiVideo、WebEx 三者的多媒體錄製檔，都是自行制定的檔案格式，因此無法客觀比較其畫質。
3. MultiVideo 與 WebEx 都是目前商業化的軟體，在錄製多媒體檔案大小的比較上 MultiVideo 優於 WebEx。但是 MultiVideo 並沒有辦法完整展示 PPT 檔的特效功能，因此在線上會議時 MultiVideo 會喪失 PPT 的特效功能。
4. MultiVideo 與 WebEx 都無法在會議進行時，完整呈現嵌入在 PPT 中的聲音及影片檔。
5. 講解手在錄製功能上略勝於 MultiVideo，且講解手可以對錄製後的檔案做文件保護的功能，MultiVideo 與 WebEx 並無此功能。
6. 講解手可以將錄製下來的多媒體檔案與 Web2.0 的 Blog 平台(樣版國度[12])作結合，MultiVideo 與 WebEx 並無此功能。

## 2.3 使用工具介紹

本研究選擇2個成熟的開發軟體作為開發工具:(1)智勝國際講解手 (2)Skype API。根據上一小節的比較結果，由於智勝國際開發的講解手，有出色的多媒體錄製功能，與嵌入在PPT中完整的特效、聲音及影片的呈現，但是缺乏了線上會議系統的功能。因此若講解手有了多人線上會議系統的功能，再加上它出色的多媒體錄製功能，相信是個不錯的選擇。

Skype 是目前熱門的網路電話軟體，而且有開放出 Skype API 可以使用，所以可以使用到 Skype 網路傳輸對話聲音的優點，包括速度快、音質佳…等。以下兩個小節將一一介紹這兩樣工具。

### 2.3.1 講解手

SoEzLecturing 講解手，是由智勝國際所開發的多媒體錄製工具，目前多使用於教材錄製方面的應用，包括學校老師教材錄製、公司員工訓練教材錄製…等等。

這套工具可以將教師在講課過程中的投影片、講解筆、聲音流程錄製下來，而其錄製的方法並非直接將螢幕畫面擷取成為影片檔，而是將投影片解析後，轉換為自訂格式，再分別記錄投影片、講解筆、聲音的流程，所以它錄製後的檔案大小較小。

講解手錄製的多媒體錄製檔，可與 Web 2.0 的 Blog 平台 – 樣版國度，相結合，使用者的錄製檔案可以上傳至此平台，再供其他人瀏覽及下載，相當方便。而且還提供了 DRM 的文件保護機制。

而且這套軟體目前尚缺乏多人線上即時會議的功能，因此選擇它作為開發軟體。

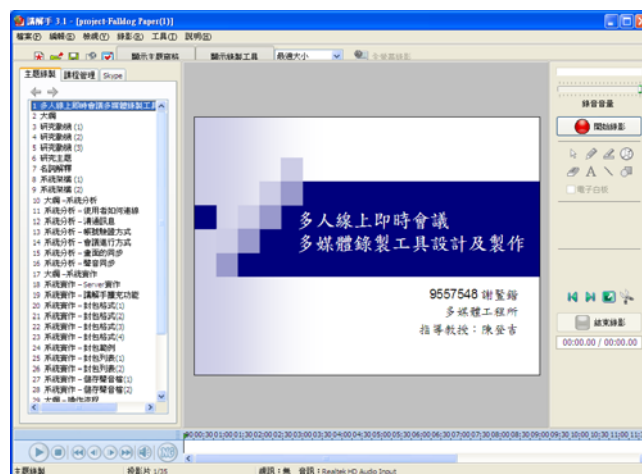


圖 2-2 講解手的使用介面

## 2.3.2 Skype API

網路通訊對話工具 – Skype，它是時下熱門的 VOIP 網路電話軟體，語音通訊採 P2P 技術，利用 UDP 傳輸聲音，因此具備傳輸速度快、音質好的特色，讓許多使用者愛不釋手，在全球使用者已超過 5000 萬人。而 Skype 也有釋出 API 供開發者開發使用。

Skype API 並非像一般的 Library 一樣，提供標頭檔(.h 檔)與編譯完成的連結檔(.obj 檔)。而是透過類似外掛(plug-in)的方式去控制 Skype 的主程式。所以執行 Skype API 的程式時，務必需先執行 Skype 程式，才會有效。在 Windows 底下可透過兩種方法去控制 Skype：

### 1. Windows Message

執行使用者自行撰寫的 Skype API 程式，傳送 Windows 的 Message 告訴 Skype 主程式一些訊息，Skype 主程式會回傳相關的 Message 給 Skype API 的程式。

比如，我的程式想獲得Skype的好友清單，可以傳送「*SEARCH FRIENDS*」的Message給Skype主程式，等Skype主程式傳回好友清單的訊息「*USERS falldog, mary, jake*」，表示此Skype帳號有三個好友：falldog, mary, jake。

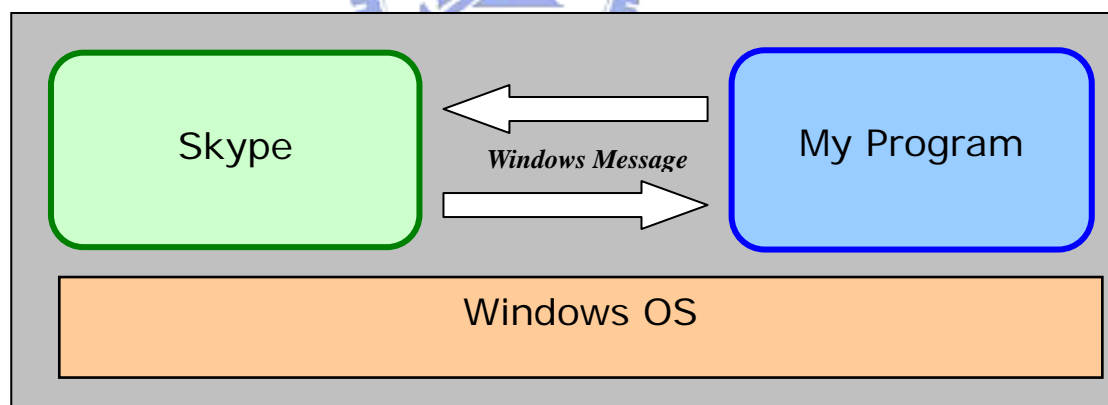


圖 2-3 Skype API 示意圖(Windows Message)

### 2. COM

在安裝新版的 Skype(Version 3.6)時，Skype 就會自動在 Windows 註冊 Skype 的 COM 元件：「**Skype4Com**」了，因此使用者可以透過已註冊的 Skype4Com[15]控制 Skype 主程式。

而此方法與 Windows Message 的方法，主要的差異在於，程式碼的直觀性。由於用 Windows Message，兩支程式都是透過訊息溝通的，所以接收訊息的處理都必須寫在同一個 Function 中，比如要取得好友清單，必須先送出

「*SEARCH FRIENDS*」的訊息，但是必須等待下一個程式處理的 Cycle 才會接收到 Skype 回傳的訊息「*USERS falldog, mary, jake*」，這樣會讓程式變得有點支離破碎。

而 Skype4Com 的方法，程式撰寫就很直觀，直接透過 COM 元件，就可立即取得使用者的好友清單(類似 Function 的呼叫)。所以用程式的觀點來看，使用 COM 的方法來撰寫 Skype API 是比較好的。

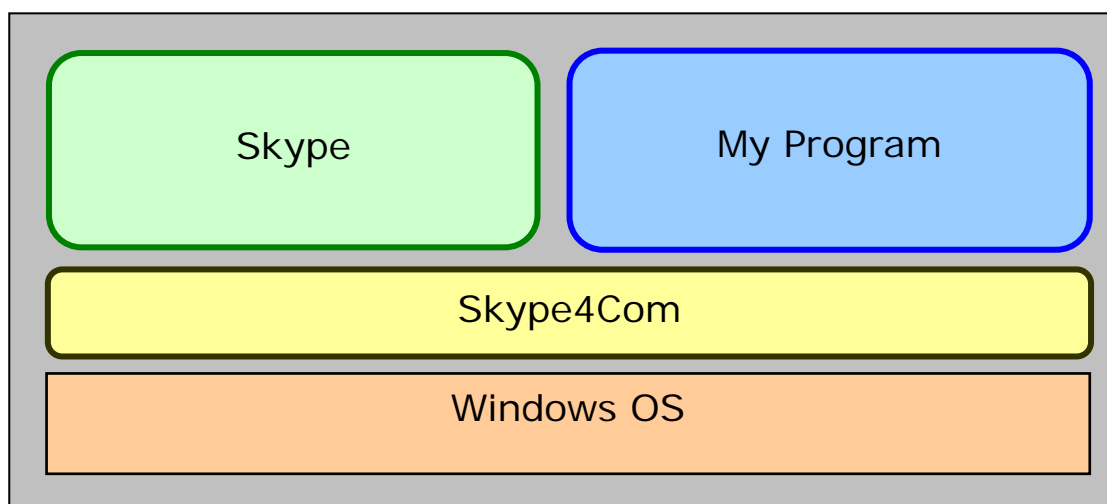


圖 2-4 Skype API 示意圖(Skype4Com)

## 2.4 小結

綜合以上分析，要實作出一個出色的多人線上會議多媒體錄製工具，本研究選擇在多媒體錄製方面相當出色的「講解手」，再加上網路語音技術純熟的 Skype，相信這樣子的結果會比目前市面上的多人線上會議軟體都還要好。

在之後的章節，將先分析系統的架構，與如何結合 Skype API 的功能，再說明如何依照這樣子的系統架構下進行實作。

## 三、系統需求分析

本研究要實作出多人線上會議系統加上多媒體錄製工具的功能。因此這個章節將就使用者需求方面來進行分析，等分析完成後就能依此需求進行實作。

以下將依照整個系統使用的流程，做出系統分析。包括使用者在一開始使用時，該如何進行連線，連線後該如何進行溝通，以及進行會議時，會議參與者畫面與聲音的同步…等等。

### 3.1 使用者連線模式需求

考慮到眾多的使用者，在開啟軟體後，該如何透過網路連線溝通呢？這邊列出兩個常見的作法來比較：

#### 1. 不需要設置中介伺服器

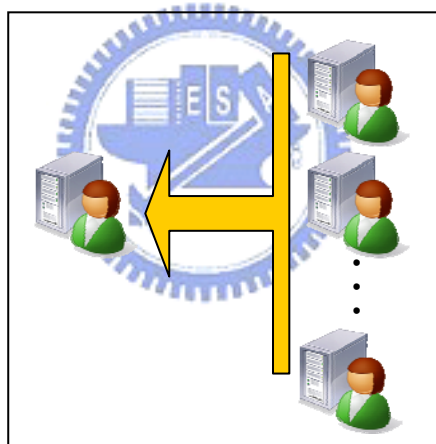


圖 3-1 連線模式 – 不需中介伺服器

這個方法必須在會議的參與者中，找出一個擁有實體IP的使用者當主席，其他使用者需連線至主席的機器後，才能與其他的使用者進行溝通。也就是將主席的電腦暫時當作中介伺服器，並由主席負責會議進行中所有人之間的溝通訊息傳遞。

- ◆ 優點：不需要架設額外的硬體設備(中介伺服器)，所需的經費較少。
- ◆ 缺點：造成使用者的不方便，因為並非所有的使用者都具備「實體IP」與「虛擬IP」的概念。而且，也並非所有的人網路都能擁有實體IP。也會造成主席端的負擔 – 包括傳遞所需要的運算及頻寬。

## 2. 需要設置中介伺服器

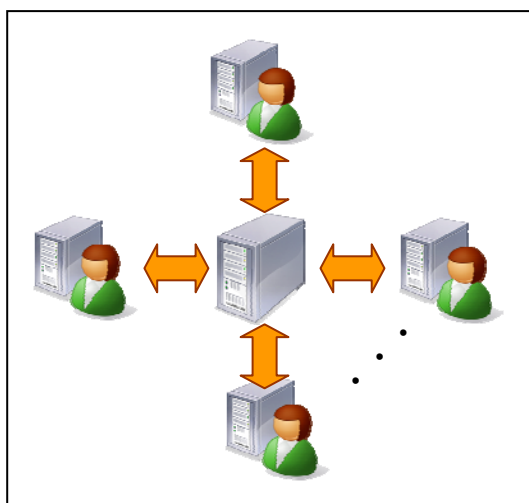


圖 3-2 連線模式 - 需中介伺服器

所有的使用者，連線至同一部中介伺服器，再由中介伺服器負責傳遞所有使用者的溝通訊息。

- ◆ 優點：使用者的操作上較簡單易懂，只要點選連結功能按鈕，即能自動連線至預設的中介伺服器。很多即時通訊軟體都是依照這個概念做的，如：MSN[20]、Skype、Yahoo即時通[21]…等等。
- ◆ 缺點：需要架設硬體設備(中介伺服器)，要一筆額外的經費。

由於多人線上會議並非網路聊天，本研究定位所研發完成的軟體，將應用於公司企業的線上會議討論。基本上公司企業內部使用，考量其使用者皆以其公司的員工為主，故較傾向第 2 種方法，需要一部內部的伺服器來自行控管。

因此本研究選擇方法 2：需要設置中介伺服器，作為實作基礎架構。

## 3.2 使用者間溝通訊息需求

根據上一小節，本研究採用「需要設置中介伺服器」的方法為連線的實作。而在眾多使用者連線至伺服器後，該如何與伺服器以及其他的使用者進行溝通呢？考量伺服器是自行設置的，所以傳遞的封包訊息是可以自行制訂的。

依封包的需求，封包格式必須提供兩種版本：(1)使用者→伺服器 (2)使用者 A → 使用者 B。第一種是使用者連線後，與伺服器的一些溝通訊息。第二種是會議進行中時，使用者 1 有訊息要通知使用者 2 時使用。



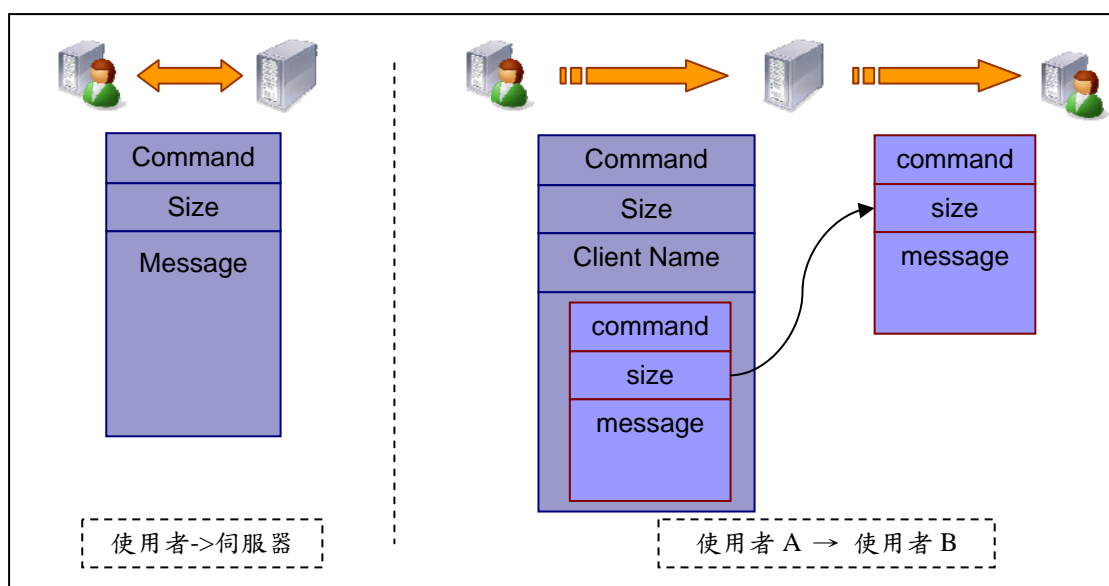


圖 3-3 溝通訊息封包格式

兩種封包格式制定如上圖所示，左半圖表示使用者→伺服器；右半圖表使用者 A→使用者 B。由於所有的訊息都是透過中介伺服器負責傳遞，可看到右半圖使用者 A 會先將訊息傳給中介伺服器，中介伺服器將封包解析後，再傳給封包內「Client Name」欄位指定的使用者 B。

至於詳細的封包格式定義，將於之後的系統實作中再詳盡解釋。

### 3.3 會議進行方式需求

這一節將會討論會議進行方式，使用者的需求，依照簡單明瞭的步驟與流程，讓使用者容易上手操作。分析後的結果，可以當作實作的參考依據，讓軟體使用操作環境更貼近人性需求。

以下將會議進行劃分為幾個項目進行討論，而會議的進行也會依照這個流程去進行：

#### 1. 開啟講解手

使用者想要使用線上會議的功能，首先必須開啟講解手這套軟體，接著點選「線上會議」的功能鈕，然後講解手會自動連線上預設的中介伺服器(預設並不會自動連線到中介伺服器)。

連上中介伺服器後，講解手會列出目前在線上的使用者清單，以及已開啟的會議室清單。

## 2. 開啟/加入會議室

完成第一步驟後，使用者可以向中介伺服器提出申請，選擇開啟會議室，自己當主席；或是選擇加入已經開啟的會議室。

## 3. 會議前置準備

開啟會議室的使用者，即為主席。主席在開啟會議室後，必須將會議進行時所需要的投影片檔，傳輸至中介伺服器，等待其他的使用者加入會議室後，中介伺服器再將主席的投影片檔傳給其他的使用者。

## 4. 開始會議

主席等待所有應該加入會議的使用者加入後，然後再等待所有的使用者完成下載會議所需的投影片檔。主席即可決定開始進行會議。

## 5. 控制權轉移

會議開始進行時，主席擁有「會議控制權」，主席有權將控制權轉移給其他的會議參與者。當其他的會議參與者拿到控制權後，即可操控會議畫面的進行。並讓所有的使用者都看到相同的講解畫面。

## 6. 會議結束

會議結束後，主席可以將會議的過程錄製下來，製作成多媒體錄製檔。主席可以選擇將多媒體錄製檔儲存至本機端，或是發佈至 Web2.0 Blog 平台，供其他的會議參與者瀏覽。



# 3.4 會議同步需求

這一節將會討論在會議進行的時候，會議同步的需求，讓每一個會議參與者都能夠宛如身處於同一間會議室中，因此必須讓每個人的電腦螢幕畫面、以及對話的聲音做即時的同步。

以下將分兩個小節說明會議進行時，畫面同步、聲音同步的需求。

## 3.4.1 畫面同步需求

在會議進行時，所有的會議參與者的畫面必須與目前擁有控制權的使用者畫面同步。會議開始時控制權為主席所擁有，之後主席可選擇將控制權轉移給其他的使用者。

畫面同步的內容，主要分兩大部分：(1)投影片內容流程 (2)使用者講解筆流程，擁有控制權的使用者，在操縱講解手時，改變投影片、講解筆的過程時，即送出訊息給其他使用者，其他使用者接收到訊息後，再將畫面進行處理即可同步。這兩部分完成同步，即表示畫面已經同步。以下將解釋這兩部分同步的細節：



## 1. 投影片內容流程

投影片檔即會議一開始時，由主席負責傳送的檔案。當會議進行時，即以此投影片檔作為會議的主軸，當擁有控制權的使用者切換投影片時(上一頁、下一頁)，其他的使用者也必須看到同樣的投影片畫面。

## 2. 使用者講解筆流程

會議進行時，擁有控制權的使用者，移動講解筆(滑鼠)至說明的段落上，也有權在投影片畫面上標記重點，及輸入文字，而標記的目的就是讓所有人關注到同一個重點段落。因此所有的會議參與者都必須看到同樣的講解筆標記的內容。

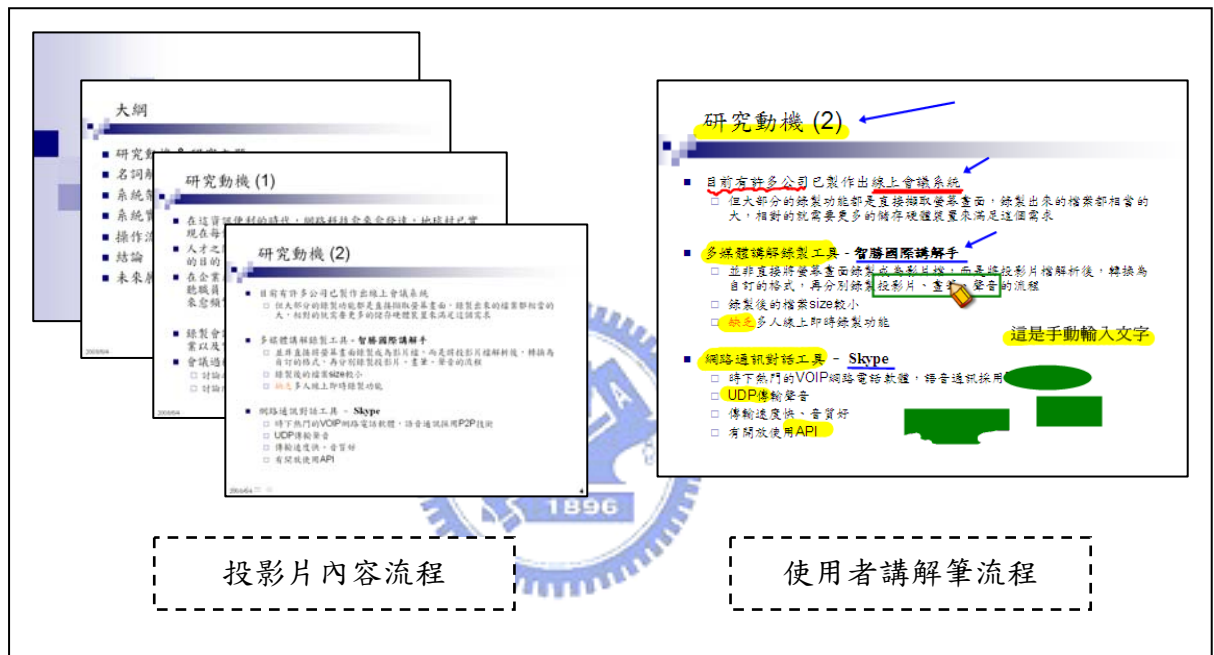


圖 3-4 會議進行畫面同步內容

## 3.4.2 聲音同步需求

在會議進行中，所有的會議參與者要可以透過語音進行溝通、對話，讓所有的使用者可以透過談話討論達到會議溝通的效果。

聲音同步的處理，全部交由 Skype 處理。只要透過講解手控制 Skype 即可。

1. 只要在會議開始時，由講解手透過 Skype4Com 控制 Skype 的行為。
2. 由主席負責撥號給其他的使用者，等待其他使用者接通後，即可開始進行會議。
3. 結束會議後，再由主席負責將通話結束即可。

## 3.5 系統操作流程

根據上面幾個小節所介紹的需求分析，可以擬定大概的使用者操作流程。這個操作流程無關系統內部的實作部分，主要考量是使用者所看到的操作環境。因此先放在這個小節介紹，之後系統實作時，即可依照這樣子的使用者操作流程下去實作使用者所需要看見的畫面。

以下會將操作流程分為三個部分介紹，包括(1)會議前置作業 (2)會議進行時 (3)會議結束。

這三個部分所需要的項目，分別為：

### 1. 會議的前置作業

- A. 確定已安裝好講解手與 Skype
- B. 登入 Skype 與連線至中介伺服器
- C. 確定所有的會議參與者皆取得相同的投影片
- D. 主席等待所有的使用者準備完成，即可開始會議

### 2. 會議進行時

- A. 講解手開始錄製
- B. 主席掌控會議
- C. 主席可將發言控制權交給其他的會議參與者
- D. 擁有控制權的使用者，負責傳送畫面控制同步的訊息，給其他所有的會議參與者

### 3. 會議結束

- A. 主席將所有的會議討論過程、對話過程儲存下來
- B. 製成發佈檔
- C. 上傳至 Web2.0 的 Blog 平台

### 3.5.1 會議前置作業

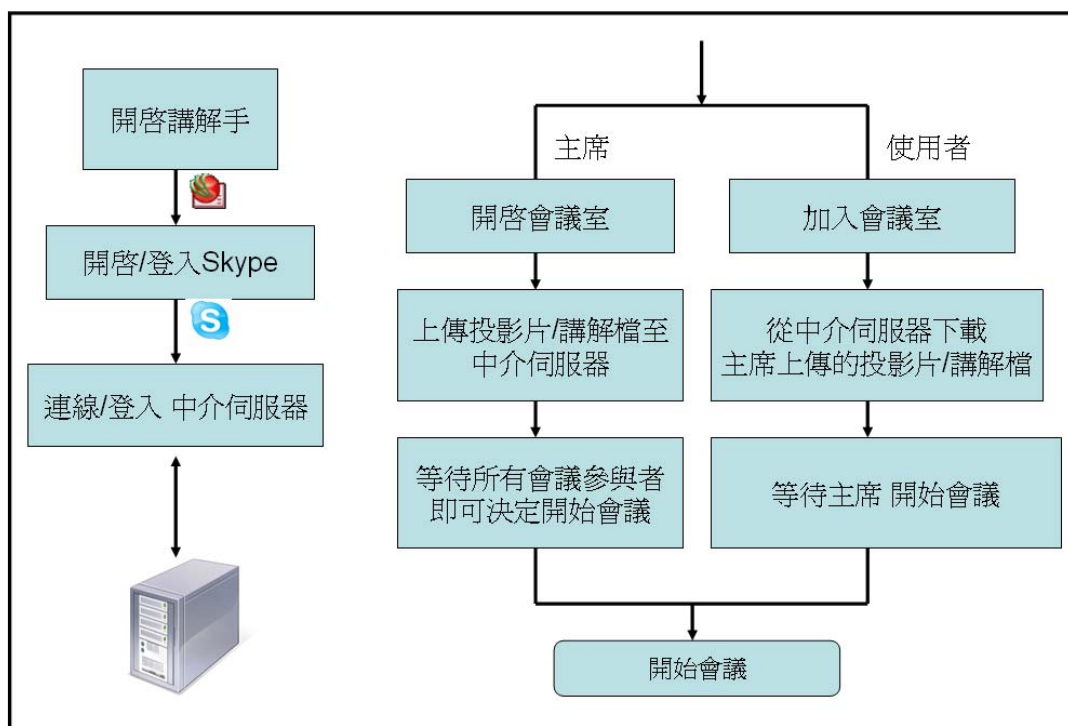


圖 3-5 會議前置作業操作流程图

由上圖內容的描述，可清楚了解一位使用者，如欲使用線上會議的功能，只要按照著上面的流程图步驟，一步一步執行，即可開始進行線上會議。

左邊的流程图為一開始的軟體操作，包括講解手、Skype 的啟動，以及連線至中介伺服器。右邊的流程图為連線至中介伺服器之後的處理動作，主要區分為主席與其他的使用者，兩者的定位不一樣，因此操作的流程內容也不盡相同。

而主席可以選擇上傳投影片或是講解檔，作為會議的簡報檔案。

如果主席上傳投影片，其他使用者加入會議室，下載投影片後，必須自行將投影片解析成為講解手能處理的講解檔，這個動作必需要花一段時間處理。如果有些使用者的電腦較為老舊，則勢必拖慢會議的進行。

而主席如果選擇上傳講解檔，此為主席講解手解析過後的檔案，只要將其資料夾壓縮後，上傳至伺服器。其他的使用者下載後解壓縮此講解檔，即可馬上進行會議。

## 3.5.2 會議進行中

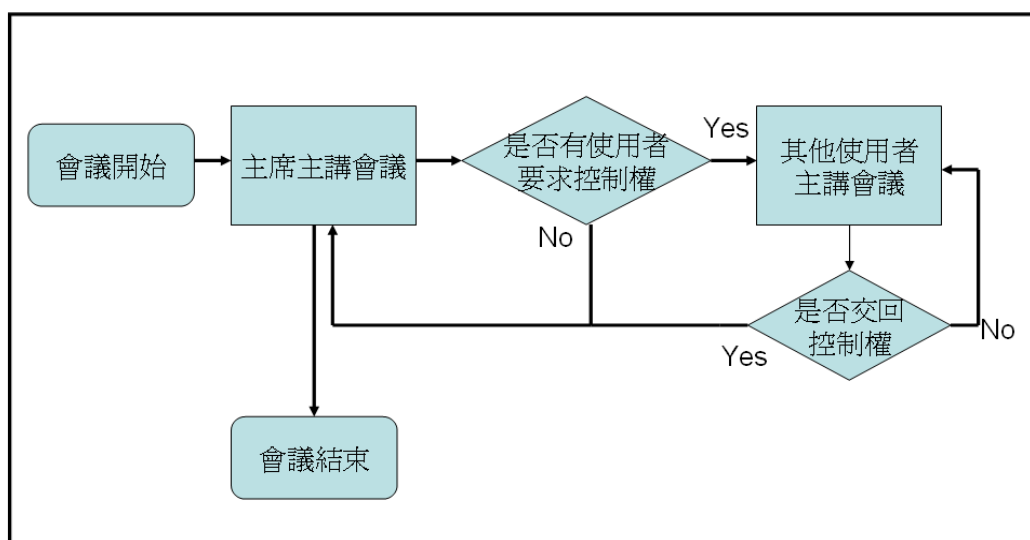


圖 3-6 會議進行中操作流程图

由上圖內容的描述，可清楚了解一位使用者，在使用線上會議的過程時，所需要使用的操作流程。

此內容無關系統實作部分，因此不會出現系統畫面、聲音同步的部分，只呈現使用者在操作上，所會遇到的步驟。

會議開始進行後，由主席開始主講會議，如前面幾個小節介紹的，主席擁有發言權控制權，所以在發言的過程中，如果有使用者要求發言控制權，主席可決定是否給予發言控制權。若其他人擁有發言控制權時，可自動繳回發言權，或是由主席主動取回發言權。

而且只有主席有權利將會議結束。

### 3.5.3 會議結束

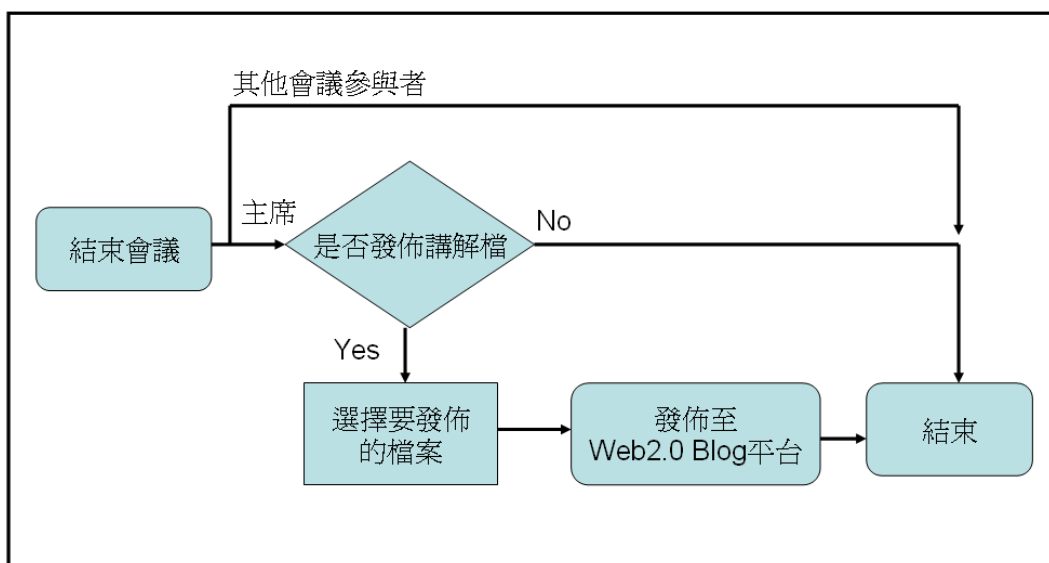


圖 3-7 會議結束作業操作流程圖

在結束會議後，除了主席外，其他的會議參與者已不需要其他額外的操作了。因為主席要負責將這次會議進行時，需要將錄製下來的多媒體錄製檔，發佈至 Web2.0 Blog 平台。

本研究是以智勝國際開發的「樣版國度 Blog」為此 Web 2.0 Blog 平台。因此主席可以將多媒體錄製檔發佈至樣版國度 Blog 後，再請其他有興趣的使用者，連上此 Blog 觀看此次會議的討論過程。

## 四、系統架構與實作

本系統的開發環境是 Borland C++ Builder 6.0，作業系統環境是 Microsoft Windows XP

### 4.1 系統架構

本系統是基於講解手 3.1 版進行功能性擴充。因此本研究必須建置原本講解手的開發環境(Borland C++ Builder 6.0)，才能夠繼續開發。

所以本研究在講解手原本的架構下，加上一層處理多人線上即時會議的 Layer，規劃後的系統架構圖如下所示。

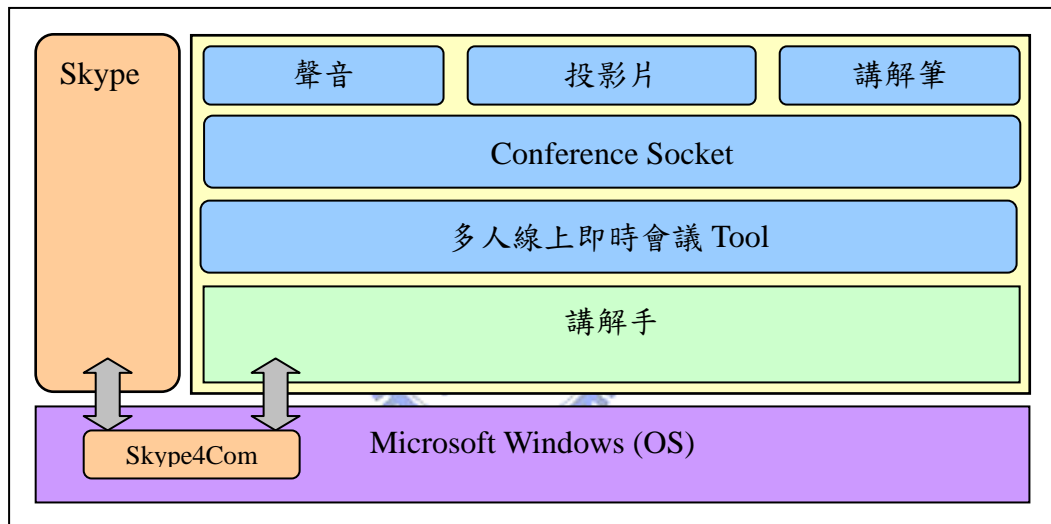


圖 4-1 系統架構圖

分析與解釋：

1. 本研究建構於 Microsoft Windows 的 OS 上。
2. 必須同時執行 Skype 與講解手，才能讓講解手使用 Skype4Com 的功能。
3. 在講解手上建構一層「多人線上即時會議 Tool」處理多人線上即時會議的功能。
4. 由「Conference Socket」處理網路的三個會議同步模組(1)聲音 (2)投影片 (3)講解筆。
5. 聲音模組必須由「多人線上即時會議 Tool」透過 COM 元件：Skype4Com 去操控 Skype 主程式去處理。

### 4.1.1 系統示意圖

在使用者的角度來看「多人線上會議系統」。在這個系統中，分為兩個主要的角色(1)使用者 (2)中介伺服器，而使用者又分為(1)主席 (2)其他會議參與者。

每個使用者的電腦上必須安裝講解手與 Skype，才能透過多人線上會議功能，連線至中介伺服器進行會議。

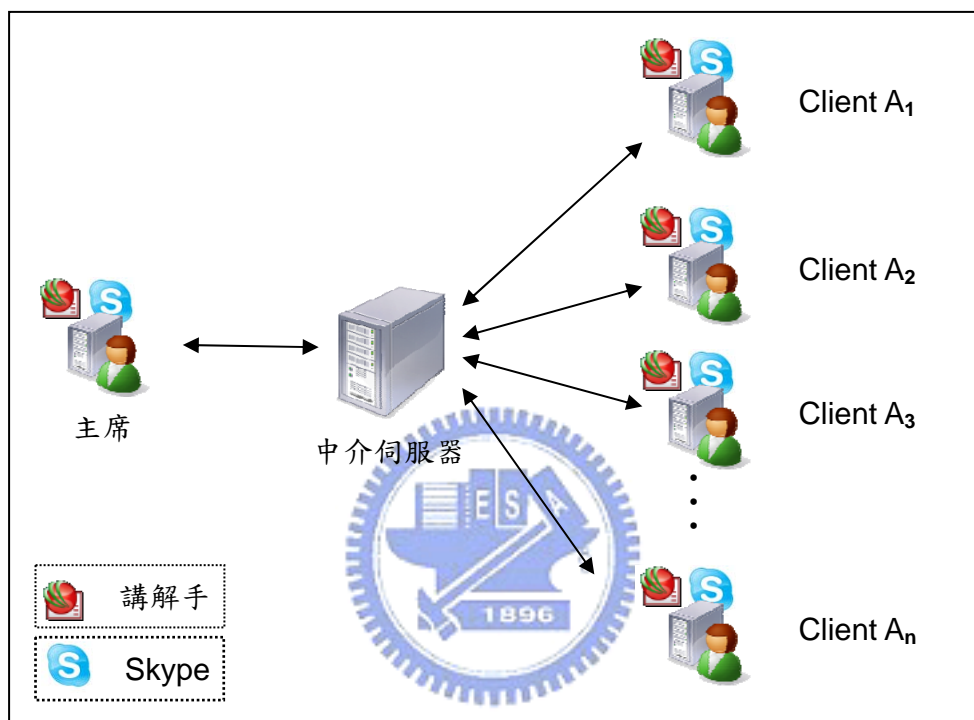


圖 4-2 系統示意圖

## 4.2 講解手擴充功能設計與實作

本研究是基於講解手 3.1 版，進行功能性擴充，所以必須在講解手原始的架構下進行功能性擴充。以下將介紹本研究在原本的講解手架構下，如何運作：



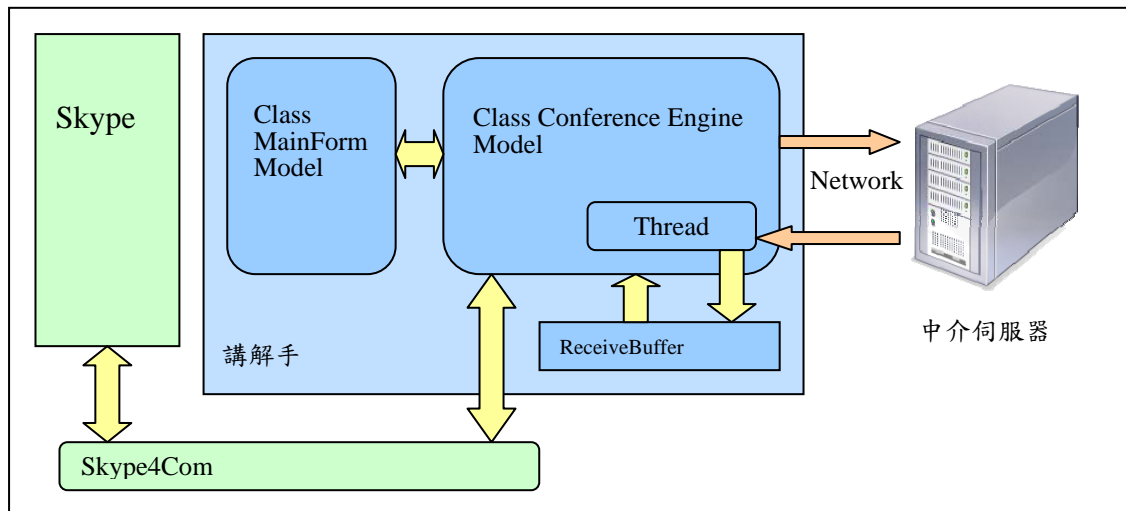


圖 4-3 講解手擴充功能示意圖

由圖 4-3 講解手擴充功能示意圖可知，講解手原本主要的模組為 MainForm 模組。在實作上，我自行增加了一個 Class Conference Engine 模組，負責處理所有線上會議應該有的功能。

由 Conference Engine 模組負責與中介伺服器的訊息溝通，並產生一個 Thread 專門負責處理中介伺服器傳送過來的封包訊息。Thread 將封包訊息寫入 ReceiveBuffer 中，然後再通知 Conference Engine 處理。由 Conference 負責將 ReceiveBuffer 中的訊息做相對應的處理。以下有簡略的程式碼供參考：

```

class ConferenceEngine : public TThread{
public:
   fastcall ConferenceEngine( TComponent* Owner );
   fastcall ~ConferenceEngine();
   void  ConSendCommand(const enum UserMessage & _cmd);
   void __fastcall ConClientOnConnecting(TObject *, TCustomWinSocket *);
   void __fastcall ConClientOnDisconnect(TObject *, TCustomWinSocket *);
   void __fastcall ConClientOnRead(TObject *, TCustomWinSocket *);
    ...
protected:
    void __fastcall Execute();
private:
    void SBP_Receive( char* _text, int _size );
    void SBP_Process();
    TClientSocket * mClient;
    TSkype * mSkype; //Skype4Com
    char * mReceiveBuffer;
    int  mReceiveBuffer_ptr;
    ...
};

```



程式碼說明：

1. class ConferenceEngine 即為剛提到的 Conference Engine 模組
2. 網路傳輸部分
  - A. 利用 BCB 的元件 TClientSocket 連線至中介伺服器。
  - B. ConClientOnConnect & ConClientOnDisconnect & ConClientOnRead 為 mClient 的 Event，當連線、斷線、收到資料時，就會觸發這幾個 Event。
  - C. 讓 class ConferenceEngine 繼承 BCB 的 TThread，再實作 Execute() 函式，即可實作出一個 Thread。
  - D. 當有資料想傳送給中介伺服器時，就呼叫 ConSendCommand() 去處理。
  - E. 當中介伺服器傳輸資料過來時，就會觸發 ConClientOnRead 去處理，在此函式中會喚醒 Thread 來接收資料。此時會呼叫 SBP\_Receive() 函式，將資料寫入 mReceiveBuffer 中。
  - F. 當接收完成後，再呼叫 SBP\_Process() 函式去處理。
3. 與 Skype 溝通
  - A. 利用 BCB 的功能匯入 Skype 提供的 Skype4Com.dll 後，就會自動安裝一個 TSkype 的 COM 元件。
  - B. 宣告一個 TSkype 的變數 mSkype，即可透過 Skype4Com 與 Skype 主程式溝通。



## 4.2.1 使用 Skype4Com

由於開發環境是 Borland C++ Builder，因此要使用 Skype4Com 來與 Skype 溝通，則必須透過 Borland C++ Builder 「Import ActiveX Control」 的功能，匯入 Skype 官方提供的 Skype4Com.dll(for Windows)。完成後，就會自動安裝一個 TSkype 的元件可以使用。

在開發時，即可用此 Class TSkype 與 Skype 主程式連線，進而操控 Skype，包括：取得 Skype 登入帳號、取得好友清單、撥號給其他 Skype 帳號…等等。

以下列出幾個在實作中使用的 Function 與 Property：

1. 與 Skype 主程式建立連線：  
`mSkype->Attach( 5, true );`
2. 取得 Skype 的登入帳號：  
`mSkype->CurrentUserHandle`。
3. 撥電話給其他使用者：  
`mSkype->PlaceCall( "user_name", ... )`。

4. 掛斷通話：

```
ICallCollectionPtr call_list = mSkype->ActiveCalls;  
for( int i = 1 ; i <= call_list->Count ; i++ )  
    call_list->get_Item(i)->Finish();
```

5. 將通話中其他使用者的聲音錄製到檔案中：

先取得所有正在進行的通話(ActiveCalls)，比對之前記錄的 CallId，再呼叫 set\_OutputDevice()將 output 導至檔案中(callIoDeviceTypeFile)。

```
ICallCollectionPtr call_list = mSkype->ActiveCalls;  
for( int i=1 ; i <= call_list->Count ; i++ ){  
    if( call_list->get_Item(i)->get_Id() == mCurrentCallId )  
        call_list->get_Item(i)->set_OutputDevice( callIoDeviceTypeFile, "FileName" );  
}
```

## 4.3 中介伺服器設計與實作

中介伺服器的功能主要在於協調所有連線進來的使用者，讓大家可以開創或加入會議室的功能，並且負責會議開始後，所有資料訊息的傳遞及導傳。其主要任務為將訊息由使用者 1 傳送給使用者 2，可能是 1 對 1 或是 1 對多，故稱其為「中介伺服器」。

所以中介伺服器必須具備下面幾個功能：

1. 管理所有連線的使用者
2. 管理所有開創的會議室
3. 可接收/傳送單一個使用者訊息
4. 使用者可加入/離開會議室
5. 使用者開啟會議室後(主席)，可上傳檔案至中介伺服器
6. 使用者加入會議室後，可下載主席上傳的檔案
7. 在會議進行中，可接收單一個使用者訊息後，廣播至其他加入會議室的成員

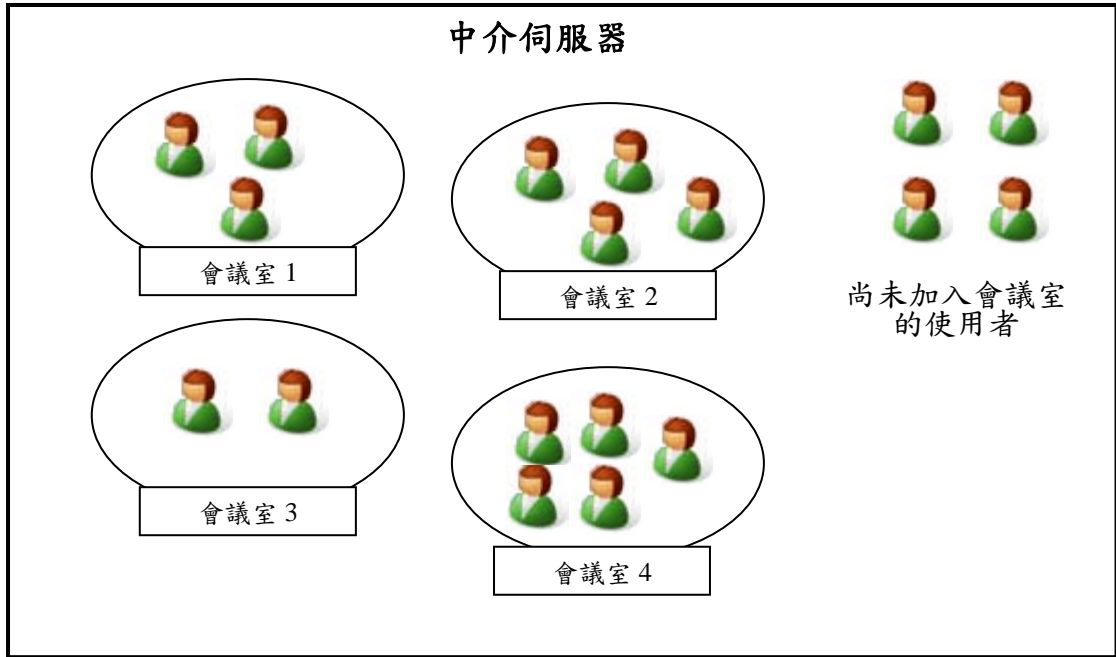


圖 4-4 中介伺服器處理示意圖

由上圖可以看出一個中介伺服器處理的概念，連線至中介伺服器的使用者，一開始是屬於「尚未加入會議室的使用者」。而會議室就像個小容器一樣，可以容納想要加入的使用者。一個使用者同一時間只允許加入一間會議室，而一間會議室可以容納多個使用者。

#### UML Class Diagram

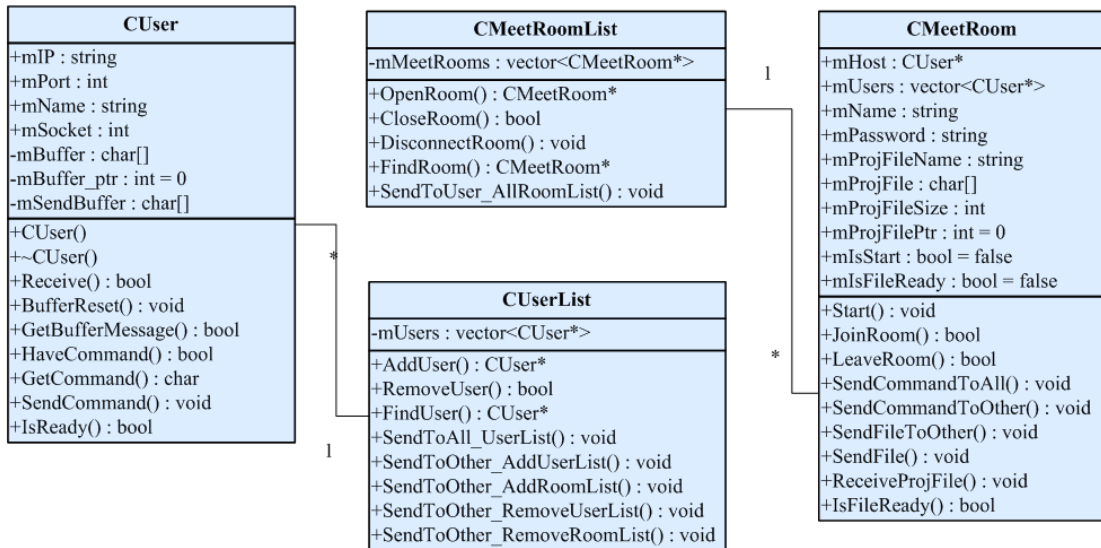


圖 4-5 中介伺服器的 UML Class Diagram

本研究開發中介伺服器的環境是 Ubuntu Server 6.10 (Linux) [16]。  
編譯器是 G++(GCC) 4.1.2 [17]。

由上圖可知，本研究將中介伺服器的功能，利用物件導向的概念，劃分為四個 Class 去實作處理：(1)CUser (2)CUserList (3)CMeetRoom (4)CMeetRoomList，前兩個是專門處理使用者相關的，後兩個是專門處理會議相關的。

本研究實作出一個伺服器，建立一個 TCP 的 Socket。由於使用者與中介伺服器溝通的訊息，一個都不能漏失掉，所以實作選用 TCP，而不用允許資料漏失的 UDP。當伺服器啟動後，就開始 Listen，等待使用者連線進來，當有使用者連線時，就 create 出一個 CUser 的物件專門負責此使用者的溝通。當有使用者要建立一個會議室時，就 create 出一個 CMeetRoom 專門負責此間會議室的訊息傳遞。

以下將一一簡介此四個 Class 的功能：

#### ◆ CUser

1. 簡介：此為專門處理單一位使用者溝通的 Class，一位使用者就對應到一個 CUser。
2. Class 中會記錄使用者連線的訊息，包括：  
mIP 記錄使用者的 IP  
mPort 記錄使用者連線進來的 Port  
mSocket 記錄使用者的 Socket FD (File Description)
3. 記錄使用者的 Skype 帳號名稱：mName
4. 每一個使用者都各自配置一段 Receive Buffer(mBuffer)，來接收傳來的訊息。用 mBuffer\_ptr 記錄目前 mBuffer 的訊息長度。
5. 收到封包時呼叫 Receive() 將封包訊息寫入 mBuffer 中，再呼叫 GetCommand() 判斷是否有完整的指令可以處理。
6. 要傳送訊息給此使用者時，呼叫 SendCommand() 即可。

#### ◆ CUserList

1. 簡介：此為管理所有連線加入的使用者 (CUser)。
2. mUsers 記錄所有連線加入的使用者(CUser)。
3. 當伺服器偵測到有使用者加入時，即呼叫 AddUser(); 有使用者離開時，即呼叫 RemoveUser()。
4. 當有新的使用者加入後，必須通知其他已連線的使用者，「有新的使用者加入」的訊息，呼叫 SendToOther\_AddUserList(), 好讓這些使用者的講解手立即更新顯示的使用者清單。
5. 其他類似功能包括：  
有使用者離開：SendToOther\_RemoveUserList()  
有人建立會議室：SendToOther\_AddRoomList()  
有人關閉會議室：SendToOther\_RemoveRoomList()

#### ◆ CMeetRoom

1. **簡介**：此為管理一間會議室的Class，處理所有加入的使用者，以及會議進行中訊息的傳輸。
2. Class 記錄會議室中應有的訊息：  
mHost：會議室的主席  
mUsers：其他加入的使用者  
mName：會議室主題名稱  
mPassword：使用者加入會議室時，所需要輸入的密碼  
mProjFile：會議進行主軸投影片檔  
mIsStart：是否已開始進行會議
3. 當一名使用者建立會議室時，就建立一個 CMeetRoom 的物件，然後將 mHost 指定為此使用者。
4. 當主席建立此會議室後，就會上傳會議室所需的投影片檔，呼叫 ReceiveProjFile() 將檔案封包寫入 mProjFile，直到傳送完成後，將 mIsFileReady 設為 True。
5. 當主席將投影片檔案上傳完成後，即會呼叫 SendFileToOther()，傳送檔案給其他已經加入的使用者們。
6. 當有其他使用者加入時，呼叫 JoinRoom()。
7. 當有其他使用者離開時，呼叫 LeaveRoom()。
8. 當有使用者想用廣播的方式告訴其他加入會議室的使用者時，即呼叫 SendCommandToOther()。

#### ◆ CMeetRoomList

1. **簡介**：管理所有已建立的會議室(CMeetRoom)。
2. 記錄所有已建立的會議室：mMeetRooms。
3. 有使用者建立一個新的會議室時，即呼叫 OpenRoom()。
4. 有主席建立的會議室結束時，即呼叫 CloseRoom()。
5. 有主席已建立會議室，但卻不正常的斷線時，即呼叫 DisconnectRoom()。

## 4.4 自訂封包格式設計與實作

而前面幾個小節講了這麼多，包括講解手與中介伺服器的設計與實作，而最重要的部分是，講解手該如何與中介伺服器溝通呢？必定是透過網路傳送一些封包訊息才能讓二者溝通。而封包的內容該如何制定？

由於本研究是自行撰寫、架設中介伺服器，所以可以自行制定講解手與中介伺服器溝通的封包格式，來符合我們使用上的需求。內容應該包含著什麼？這些



都應該是事先規劃好的，在實作時會更容易。

而這邊提到的封包格式，是指封包在 OSI 中的 Application Layer 的內容，並不包含在 OSI 中較底層的 TCP 封包格式。

首先，先考慮講解手與中介伺服器在接收封包訊息時的處理模式。由於透過網路傳輸一段資料，可能在第一次接收時，只會收到前半段，還要隔一段時間才會收到後半段，或是要分很多次接收，才會收到完整的資料。基於這個考量，所以封包裡面，一定要有記錄資料長度的欄位，才能確保資料接收完成。

如下圖所示，講解手與中介伺服器都會有各自的 Receive Buffer 接收資料，這個 Receive Buffer 就像個 Queue 一樣，先接收的資料先處理。下圖中講解手與中介伺服器各有著三段資料，其他講解手的 Receive Buffer 中三段資料皆已接收完成，但是中介伺服器的資料 3 尚未接收完畢。

因此當中介伺服器取出資料 1、2 處理完成後，要再處理資料 3 時，必須先判斷資料 3 是否已經接收完成才能進行處理，否則會有問題。

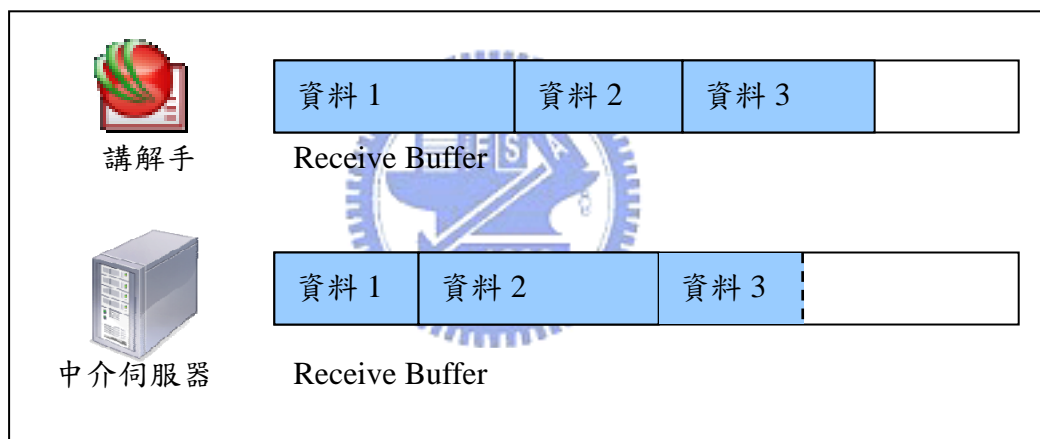


圖 4-6 接收封包示意圖

再考慮訊息傳送的來源與目的，可以概略上分為兩種：

1. 使用者 ↔ 伺服器：  
主要是在會議的前置處理時所需要的，比如使用者傳送 Skype 帳號給伺服器，使用者通知伺服器要求加入會議室…等等。
2. 使用者 → 使用者：  
主要是在會議進行時，使用者想要告訴會議室中另一位使用者訊息，或是通知所有會議室中所有的使用者一些訊息。

以下幾個小節將一一介紹這兩種訊息的封包該如何制定：

## 4.4.1 使用者與中介伺服器之間溝通的封包制定

使用者與伺服器之間的封包該如何制定？首先，先考量溝通訊息的主要型式與目的為何，包括下面幾點：

1. 通知對方，該處理什麼事情
  - A. 使用者告訴伺服器，想離開已加入的會議室
  - B. 通知加入會議室的使用者，會議開始
  - C. 通知加入會議室的使用者，會議結束
  - D. …等等
2. 通知對方一些訊息，包含文字內容
  - A. 使用者將自己的 Skype 帳號通知伺服器
  - B. 使用者通知伺服器，要求加入某間會議室
  - C. 伺服器通知所有的使用者，有新的使用者連線進入。將此新加入使用者帳號，加入講解手的使用者清單中
  - D. 伺服器通知所有的使用者，有使用者離開，在講解手的使用者清單中清除該名使用者
  - E. 伺服器通知所有的使用者，有使用者建立新的會議室，加入講解手的會議室清單中
  - F. …等等

綜合以上的分析，可以將訊息編號，組成一個個 Command Number，依照 Command 下去編號 1、2、3…。也由於需要的 Command 數量並不是太多(約為 50 個左右)，考量以後版本擴充後的需求、功能的增加可能會需要多一點 Command。所以可以用一個欄位，大小為 1Byte 來表達 Command 的編號，數量可達 256 個 Command。

依照先前圖 4-6 的解釋，因為文字訊息的長度並不一定，所以必須有個欄位記錄訊息的長度。由於文字內容的長度較長，無法像 Command Number 一樣，用 1Byte 即可表達，故使用 4Bytes 的大小，剛好是一個 32-bit OS 的 Integer 的長度，不使用負數的話，即可表達 0 ~ 4,294,967,295 長度的 Character 文字訊息。

考量可能發生的 **Exception** 情形：當伺服器端更新後，新的伺服器版本有新增的 Command，而講解手卻沒有更新，造成講解手收到新的 Command 卻無法處理。本研究的處理作法為，收到一個未知的 Command 時，秀出一個使用者訊息：「接收到未知的訊息，請使用者下載新的講解手版本，讓會議順利進行」，並將此完整的封包訊息丟棄。

依照這樣子的需求與分析，可以制定出圖 4-7 的封包格式，符合使用者與中介伺服器的需求。

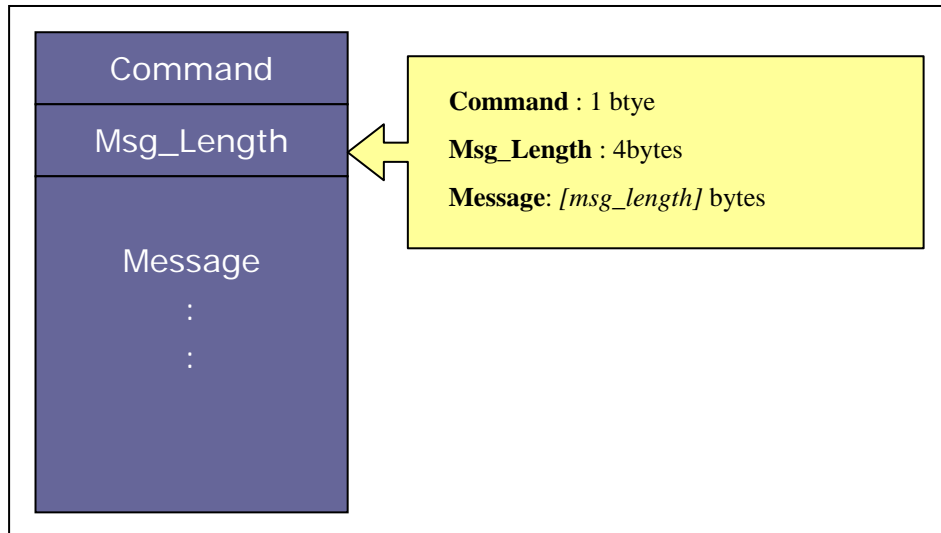


圖 4-7 使用者與伺服器溝通封包格式

如上圖所示，使用者與伺服器溝通的封包格式制定如上，解釋如下：

- ◆ **Command**：代表指令的編號
- ◆ **Msg\_Length**：為之後的 Message 欄位內容的長度
- ◆ **Message**：為訊息附屬的文字內容，可以無此內容，則 Msg\_Length 的值設為 0 即可

在 Command 實作的部分是以 C/C++ 中的 Enum 處理，因為在寫程式時以有意義的文字取代無意義的編號，讓寫作的程式碼更有可讀性。

```

// Command Number
enum UserMessage
{
    umCreateRoom      =1,
    umCreateRoomAck=2,
    umCloseRoom       =3,
    umCloseRoomAck   =4,
    umJoinRoom        =5,
    umJoinRoomAck    =6,
    umLeaveRoom        =7,
    umLeaveRoomAck    =8,
    umName            =9,
    umMessageToClient=10,
    umBroadCast       =11,
    :
};

```

圖 4-8 Command Number 的實作程式碼



## 4.4.2 使用者與中介伺服器之間溝通的封包範例

根據上一個小節所制定的規則，在圖 4-9 列出幾個實作的範例。以下將解釋實作內容的意義。

- ◆ 使用者傳送自己的 Skype 帳號給伺服器
  - A. 第一個欄位是 Command，對照圖 4-8 的內容，使用 Command - *umName*，而它的編號為 9，因此此欄位的內容為 9。
  - B. 第二個欄位是 Msg\_Length，內容是依照 Message 的內容而定。因此，Message 的內容為 Falldog，長度為 7。
  - C. 第三個欄位是 Message，內容為 Skype 的帳號：Falldog。
  
- ◆ 使用者要求伺服器開創一間會議室
  - A. 第一個欄位是 Command，對照圖 4-8 的內容，使用 Command - *umCreateRoom*，而它的編號為 1，因此此欄位的內容為 1。
  - B. 第二個欄位是 Msg\_Length，內容是「Create a Meeting Room」的長度 20。
  - C. 第三個欄位 Message，內容為會議室的名稱：「Create a Meeting Room」



圖 4-9 使用者與伺服器溝通封包範例

## 4.4.3 使用者與使用者之間溝通的封包制定

使用者與使用者之間的封包該如何制訂？首先，考量溝通訊息的主要型式為何，包括下面幾點：

1. 此種訊息，只有會議正在進行中才會發出
2. 使用者之間的溝通封包，還是需要先傳遞給伺服器，再由伺服器傳遞給指定的使用者。所以伺服器是使用者們之間溝通的橋樑，因此，使用者

之間溝通的封包必須與「使用者與伺服器溝通」的封包相容才行！

3. 通常分為兩種型式

I. 使用者 A 想要通知使用者 B 一些訊息

*Client A* → *Server* → *Client B*

II. 使用者 A 想要通知所有正在會議室的其他成員訊息

*Client A* → *Server* => *Other Clients*

綜合以上的分析，可以從「使用者與伺服器溝通的封包格式」進行改良，以達到相容的目的。

在「使用者與伺服器溝通的封包格式」中，由於文字訊息的長度是由 *Msg\_Length* 所記錄，不需要知道 *Message* 的內容為何。因此可以指定兩個 *Command Number* 供「使用者與使用者溝通的封包格式」使用，以達到相容的目的。並且，在 *Message* 的欄位裡，附上一個要傳給其他使用者的完整封包訊息 – **附屬訊息**。

依照上面的分析，封包訊息可以傳給指定的一位使用者，或是其他會議室中的使用者。如果傳給指定的一位使用者的話，必須在封包內，增加一個欄位說明此封包要傳給哪一位使用者。此欄位的內容為該使用者的 Skype 帳號，由於該使用者的 Skype 帳號的長度不一定，所以可以效仿 C/C++ 處理字串的方法，在此欄位後面加上一個字串的結束字元(\0)。因此當伺服器收到封包後解析，即可知該使用者的 Skype 帳號為何。

至於要傳給會議室中其他的使用者，只要伺服器處理到指定的 *Sub-Command* 欄位(不同於傳給一位使用者的 *Command*)，即可直接將附屬訊息解析出來，傳送給其他加入會議室的使用者。

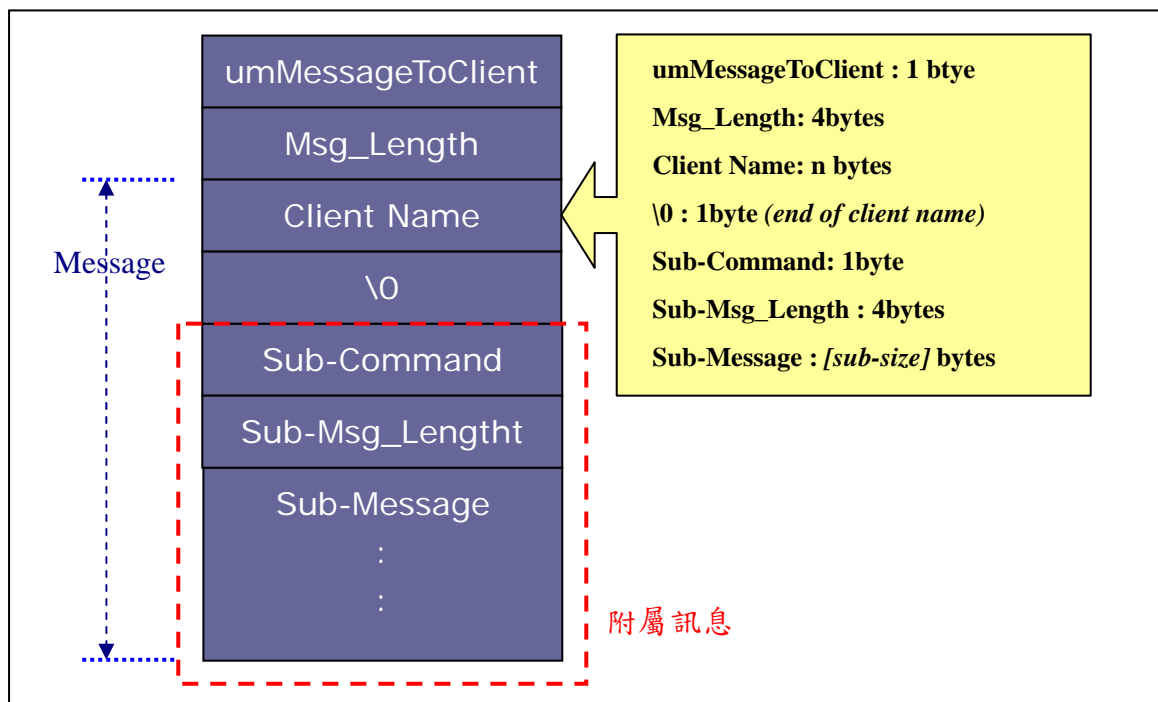


圖 4-10 傳給指定使用者的封包格式

如上圖所示，傳送給指定的使用者封包格式制定如上，解釋如下：

- ◆ **umMessageToClient**：此為指定的 Command Number，對照圖 4-8 後，此 Command Number 的值為 10
- ◆ **Msg\_Length**：此為 Message 的長度，這邊的 Message 即指以下這此欄位的總和：「Client Name + \0 + Sub-Command + Sub-Msg\_Length + Sub-Message」
- ◆ **Client Name**：此欄位為傳送的的標使用者 Skype 帳號
- ◆ **\0**：此判斷 Client Name 字串的結束字元
- ◆ **Sub-Command**：此為附屬訊息的 Command Number
- ◆ **Sub-Msg\_Length**：此為附屬訊息的 Message 長度
- ◆ **Sub-Message**：此為附屬訊息的 Message

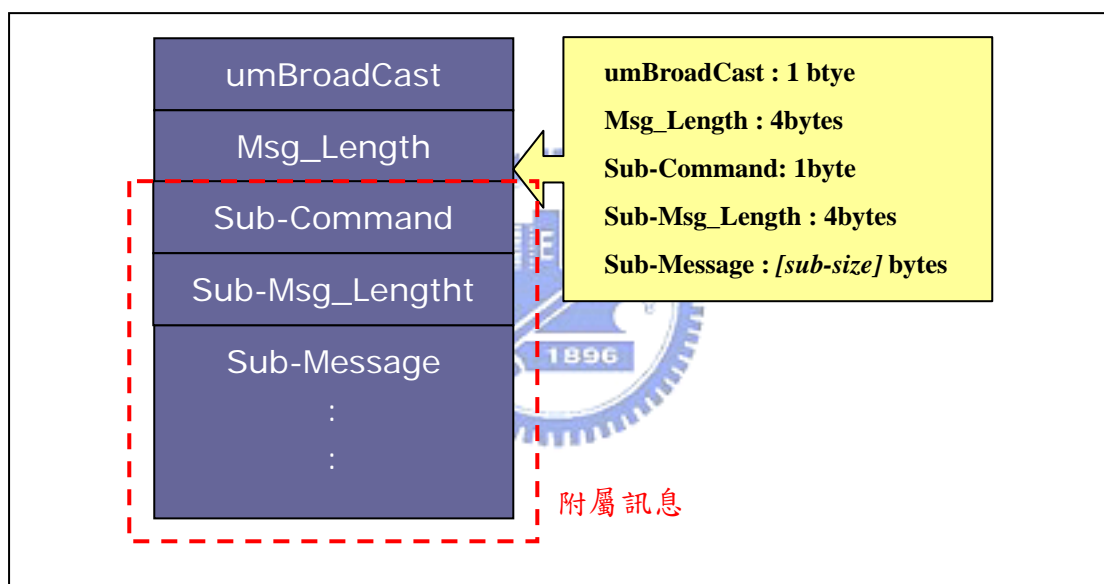


圖 4-11 傳給會議室中其他使用者的封包格式

如上圖所示，傳送給其他在會議室的使用者封包格式制定如上，解釋如下：

- ◆ **umBroadCast**：此為指定的 Command Number，對照圖 4-8 後，此 Command Number 的值為 11
- ◆ **Msg\_Length**：此為 Message 的長度，這邊的 Message 即指以下這此欄位的總和：「Client Name + \0 + Sub-Command + Sub-Msg\_Length + Sub-Message」
- ◆ **Sub-Command**：此為附屬訊息的 Command Number
- ◆ **Sub-Msg\_Length**：此為附屬訊息的 Message 長度
- ◆ **Sub-Message**：此為附屬訊息的 Message

## 4.4.4 使用者與使用者之間溝通的封包範例

根據上一小節所制定的規則裡，在下圖列出幾個實作的例子。以下將解釋實作內容的意義。

- 在會議中，主席告訴擁有發言權的使用者(Falldog)，交回發言權
  - A. 參考圖 4-12，圖左半部
  - B. 第 1 個欄位是 Command – umMessageToClient
  - C. 第 2 個欄位是 Message 的長度，包括第 3、4、5、6 個欄位內容的長度，加起來後的值為 13
  - D. 第 3 個欄位是訊息傳送目的使用者的帳號，此為 Falldog
  - E. 第 4 個欄位是用來標明第 3 個欄位內容的字串結束字元
  - F. 第 5、6 個欄位為「附屬訊息」的內容
  - G. 第 5 個欄位是附屬訊息的 Command – cmdClientListen
  - H. 第 6 個欄位是附屬訊息的 Message 長度，由於這個指令不需要額外的 Message，因此此欄位的值為 0
  - I. 因為附屬訊息的 Message 的長度為 0，因此不需要第 7 個欄位記錄 Message 的內容
  
- 在會議中，主席告訴所有的使用者，目前簡報筆移動到哪個座標
  - A. 參考圖 4-12，圖右半部
  - B. 第 1 個欄位是 Command – umBroadCast
  - C. 第 2 個欄位是 Message 的長度，也就是圖中標示的「附屬訊息」的長度
  - D. 第 3、4、5 個欄位為「附屬訊息」的內容
  - E. 第 3 個欄位是附屬訊息的 Command – cmdPanelMouseEvent
  - F. 第 4 個欄位是附屬訊息的 Message 長度，內容為「MouseMove\n10\n23」，文字長度為 15，所以此欄位的值為 15
  - G. 第 5 個欄位是附屬訊息的 Message，其內容為「MouseMove\n10\n23」，代表說目前的簡報筆移動到座標(10,23)的位置

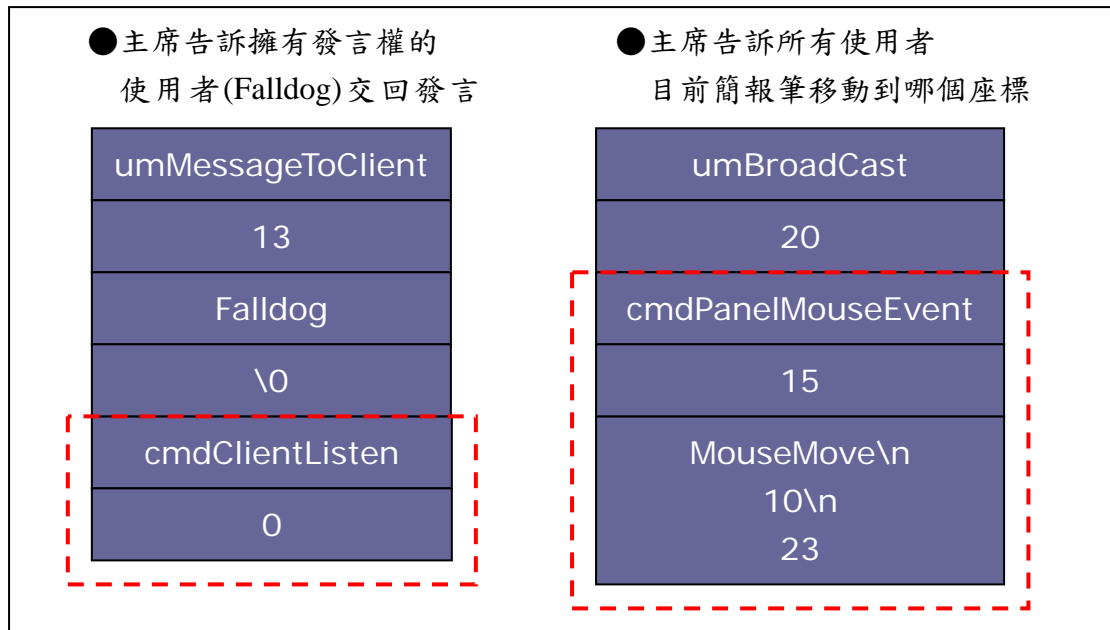


圖 4-12 使用者與使用者溝通的封包範例

#### 4.4.5 自訂封包格式列表

根據上面幾個小節的介紹，在使用者(講解手)與中介伺服器溝通的封包中，最重要的環節就是第一個欄位「Command」，在講解手或是中介伺服器接收封包後，要處理時，都會先判斷 Command 是屬於哪一類，再根據 Command 原本的定義去處理相對應的事情。

這邊將列出幾個本研究實作時定義的 Command。主要分為之前介紹的兩類：(1)使用者與伺服器溝通的 Command (2)使用者與使用者溝通的 Command

◆ 使用者與伺服器溝通的 Command 列表如下：

此類的 Command，通常用來執行會議進行前置處理，以下列出包括 umName, umCreateRoom...等等，會議前置處理的 Command。很明顯可以發現，每個 Command 皆是以 um 或是 hm 作為開頭，其意為 UserMessage 與 HostMessage。而在實作部分如圖 4-8 所述，利用 C/C++的 Enum 進行實作。

如 umName 的用途是用來處理使用者的 Skype 帳號，而它的封包儲存格式為： $\{ umName + size + Name \}$ ，umName 表示 Command，size 表示第 2 個欄位 Message 的長度，Name 表示使用者的 Skype 帳號文字內容。詳細封包格式說明請參考 4.4.1 節的說明。

表格 4-1 使用者與伺服器溝通的 Command 列表

Command 封包名稱	說明	格式
umName	使用者的 Skype 帳號	umName + size + Name
umCreateRoom	要求開創會議室	umCreateRoom + size + Name
umCloseRoom	要求關閉會議室	umCloseRoom + size
umJoinRoom	要求加入會議室	umJoinRoom + size + Name
umLeaveRoom	要求離開會議室	umLeaveRoom + size
hmAddUserList	增加使用者清單	hmAddUserList + size + Name
hmAddRoomList	增加會議室清單	hmAddRoomList + size + Name
...	...	...

◆ 使用者與使用者溝通的 Command 列表如下：

此類的 Command，通常用來執行會議進行中，會議流程的處理，以下列出包括 cmdPPTNextPage，cmdPPTPrePage...等等。很清楚可以看到這一類的封包皆是以 cmd 為開頭，在實作部分仿效「使用者與伺服器溝通的 Command」，利用 C/C++的 Enum 去實作。

此類的 Command 因為是在會議過程中所需要，所以在會議過程中封包的傳遞，通常是使用者 A→使用者 B，中間尚需要透過中介伺服器的處理，所以必須使用 4.4.3 小節介紹的封包格式。而傳給使用者的封包內容，就放在「附屬訊息」中。

而在下列的表格中，格式欄位內容，即為附屬訊息中的欄位格式：  
{ Sub-Command + Sub-Size + Sub-Message }，詳細的說明皆在 4.4.3 小節中說明。

表格 4-2 使用者與使用者溝通的 Command 列表

Command 封包名稱	說明	格式 ( Sub-Command + Sub-Size + Sub-Message )
cmdPPTNextPage	投影片換下一頁	cmdPPTNextPage + size
cmdPPTPrePage	投影片換上一頁	cmdPPTPrePage + size
cmdPanelMouseEvent	講解筆的事件(移動、繪製)	cmdPanelMouseEvent + size + Message
cmdPanelClearAll	清除畫面上的講解筆紀錄	cmdPanelClearAll + size
cmdClientListen	主席要求使用者 交回控制權	cmdClientListen + size



cmdClientActive	主席許使用者 擁有控制權	cmdClientActive + size
...	...	...

## 4.5 多媒體錄製工具之實作

本研究在多媒體錄製工具方面的目的，是將線上會議的討論過程錄下來，再透過播放器將多媒體錄製檔案播放出來，重現當時會議的畫面與聲音。

而本研究挑選的開發工具講解手，原先的功能就是以多媒體錄製工具著稱，所以講解手原本即有多媒體錄製的功能。

原本講解手錄製功能的是專門錄製投影片檔(PPT 檔)，錄製使用者講解投影片的過程，將使用者講解的投影片過程、講解筆、聲音錄製下來，成為一個講解手特定的專案檔(\*.bst 檔)。如果要觀看講解檔的話，使用講解手播放器即可重新播放。

並非每一位參與會議的使用者都需要錄製多媒體錄製檔，在本研究的設計中，只需要會議的主席將線上會議的過程錄製下來，再透過Web Blog2.0平台的分享方式，分享給其他的使用者即可。所以只需要主席負責錄製的工作，其他的使用者只要單純的參與會議即可。

本研究必須在原有的講解手架構下，加上錄製線上多人會議系統的功能。原本的講解手的錄製功能，會錄製到(1)投影片 (2)講解筆 (3)聲音的部分，應用到本研究上，只需要修改錄製聲音檔的部分即可。

因為原本的講解手在錄製的環境中，只需要錄製一個畫面、一隻講解筆、一個聲音檔。而在本研究的多人線上會議系統，在一位使用者的電腦上，只會同時出現一個畫面、一隻講解筆，但是卻需要同時出現兩個聲道的聲音：**(1)**本機端的使用者的聲音 **(2)**透過 Skype 對話的會議室其他的使用者的聲音。所以在原有的講解手錄製架構下，即可錄製線上會議過程中的畫面和講解筆。所以只要再修改成可錄製兩個聲道的聲音，即可符合本研究的需求。

原本的講解手只會錄製本機端使用者麥克風的聲音，而本系統除了錄製本機端使用者麥克風的聲音外；還必須要錄製透過 Skype 傳來，在會議室中其他使用者談話的聲音。但是由於同一個時間，錄製本機端麥克風的聲音，以及揚聲器播放的聲音，會有一些問題的出現。以下列出同時錄製兩個聲道的解決方案的分析和比較：

### 1. 單一張音效卡混音

- ◆ 此種方法，音效卡必須要具備全雙工(Full-duplex)的功能，才能夠同時將兩個聲道的聲音混音

- ◆ 優點：方便、簡單
  - ◆ 缺點：測試過後，本機端使用者在講話時，在網路的另一邊使用者會有雜音與迴音的干擾
2. **單一張音效卡錄製兩個聲道**
- ◆ 此種方法，是由音效卡將本機端使用者的聲音，和 Skype 傳來的其他使用者聲音，分別錄製，一個聲道錄製一個聲音檔。
  - ◆ 優點：方便、簡單
  - ◆ 缺點：少有音效卡提供這種功能，通常目前只有較高階的音效卡才有提供這種功能。
3. **兩張音效卡各錄製一個聲道**
- ◆ 此種方法，必須在本機端安裝兩張音效卡，然後在錄製時，指定一張音效卡錄製本機端的聲音，另一張音效卡錄製 Skype 傳來的其他使用者聲音。
  - ◆ 優點：這種方式，本機端的 CPU 在錄製時負擔較低
  - ◆ 缺點：很少有電腦會安裝兩張以上的音效卡
4. **將 Skype 中透過網路傳來的使用者聲音，導到一個檔案或是一個 Port**
- ◆ 此種方法，由於 Skype API 有提供這樣子的功能，可以將 Skype 對話時，其他使用者的聲音在播放的同時，導至一個檔案，或是導至本機的另外一個 Port 處理。
  - ◆ 優點：只需要一張音效卡，而且在錄製時，並不會出現像「方法 1」的迴音與雜音
  - ◆ 缺點：本機端在錄製時須將 Skype 的聲音導至一個檔案，使本機端的負擔變大

經過以上四點的分析，基於一般性的考量，讓大部分的電腦都能夠實現的方法，**選擇方法 4**：將 Skype 中透過網路傳來的使用者聲音，導至一個檔案。

因此講解者在錄製多人線上會議系統時，除了將本機端使用者的麥克風聲音錄製成一個檔案外，還必須將 Skype 中透過網路傳來的使用者聲音另外錄製為一個聲音檔。

這樣子最後錄製的結果會有兩個檔案，所以必須透過混音的方式，將這兩個聲音檔混音，成為一個聲音檔。這樣子在播放時，即可播放單一個聲音檔即可，不需要在播放時，再即時混音。播放時即時混音會造成播放電腦的負擔，並非是一個很好的選擇。因此本研究在錄製完成後，再將兩個聲音檔混音後儲存。



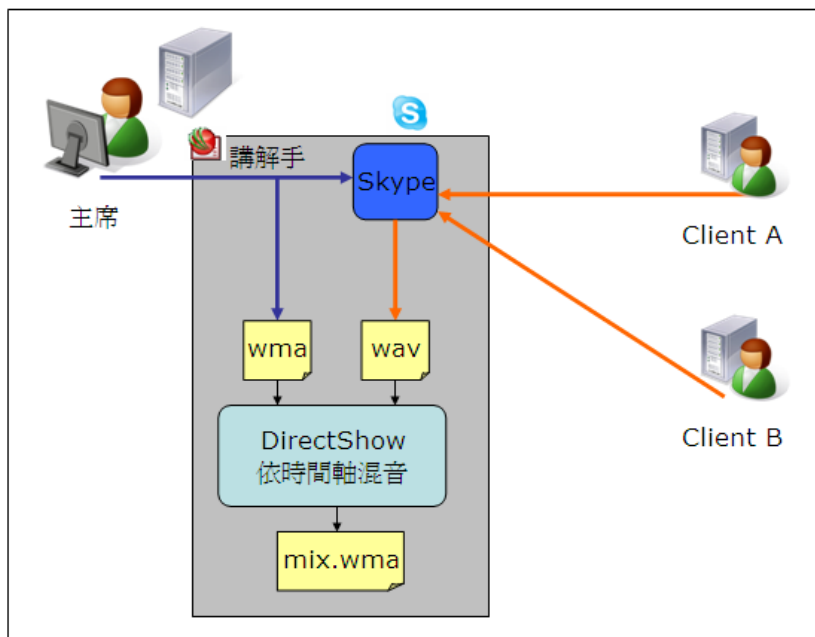


圖 4-13 聲音檔混音示意圖

上圖為本研究混音的方法，在會議進行中，將主席的聲音檔錄製下來，儲存成一個 wma 檔。再透過 Skype 將使用者 A 與使用者 B 的聲音導至一個 wav 檔儲存。再利用 DirectX 的其中一個模組 DirectShow 的功能，將 wma 與 wav 兩個檔案依時間軸進行混音，混音後的檔案儲存為 mix.wma 檔。

本研究使用 DirectX 8.1 版進行開發與測試，其中的 DirectShow 的功能只有在 DirectX 8.0 或是更新的版本才有支援。以下將簡介使用的 DirectShow 模組的 Timeline 結構的功能。

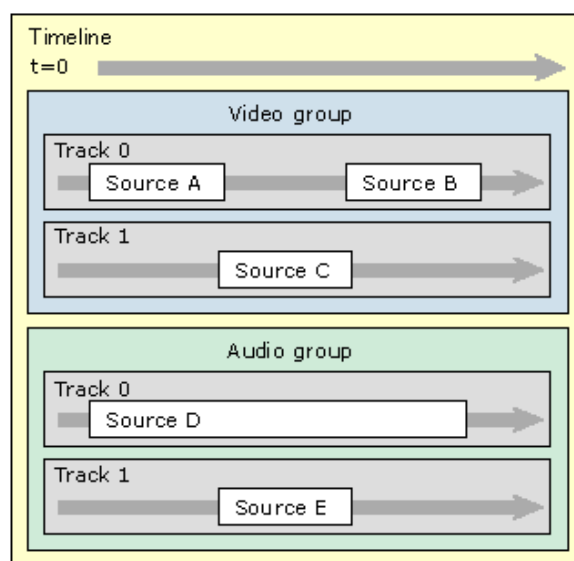


圖 4-14 DirectShow 的 TimeLine 模組示意圖

在上圖 4-14 中清楚的秀出了 DirectShow 的 TimeLine 模組的架構。在一個 TimeLine 的結構中，可容納兩個 Group：(1)Video Group (2)Audio Group。而這兩個 Group 中可容納多個 Track，每個 Track 可依時間軸存放多個 Source 檔。

比如在上圖 4-14 的 Video Group 中的 Track 0，在前半段的時間軸裡，存放一個 Source 檔 A，在後半段的時間軸裡存放一個 Source 檔 B。而 Track 1 在中間段的時間軸中存放的是 Source C 檔。最後這個 TimeLine 結構，依時間軸播放時，在 Video 的部分就會參考 Video Group 中的 Track 的 Source，所以會依序播放 Source A、C、B 檔。如果 Video Group 中各 Track 中的 Source 有重疊的部分，會以 Track 1 為較高的優先權取用。

同理，在 Audio Group 中的 Track 0 與 Track 1 中，各存放一個 Source 檔 D、E。由於這兩個 Track 中的 Source 有重疊的部分，所以在最後 TimeLine 依時間軸播放時，就會將這段重疊的部分進行混音(不同於 Video Group 中的機制)。

所以本研究利用此 TimeLine 的結構，使用其中的 Audio Group 的部分，即可將兩個聲音檔進行混音。如下圖所示，在一個 TimeLine 結構中，加入兩個 Track 進入 Audio Group，兩個 Track 各使用一個 Source 檔，包括講解手錄錄的 wma 檔，與 Skype 所錄製的 wav 檔。最後將 TimeLine 結構，依時間軸混音輸出一個檔案為 mix.wav 檔。此檔案即為 wma 與 wav 兩個檔案的混音檔。

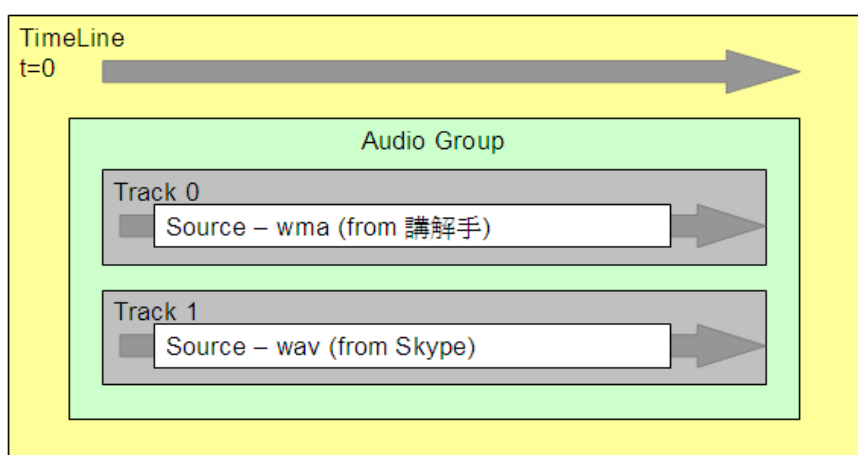


圖 4-15 講解手使用 DirectShow 的 TimeLine 模組實作混音

## 五、系統展示

本章節將會以圖示展現實作後系統介面，而展示的順序，會依照使用者在實際使用時的操作步驟，一步一步展示。詳細的使用者操作流程，已於 3.5 節完整介紹過，這邊不再多做贅述。

### 5.1 會議的前置作業

#### 5.1.1 進入會議室前

首先，開啟講解手，一開始出現的畫面如下圖所示。點選下圖標示的按鈕「啟用 Conference」，以進行線上會議。

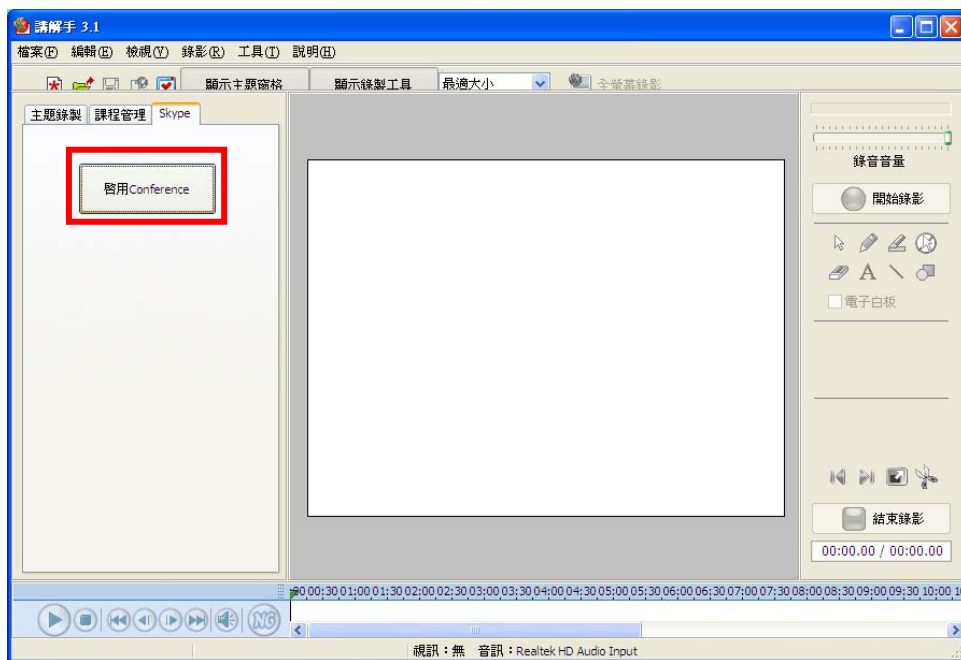


圖 5-1 講解手介面

接著講解手會偵測系統是否有安裝 Skype，如有安裝的話，是否已啟動 Skype，若尚未啟動 Skype，則會出現一個對話視窗，請使用者啟動並登入 Skype。

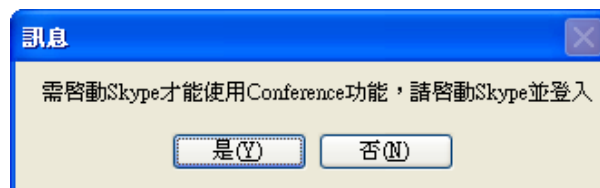


圖 5-2 啟動 Skype 的訊息視窗  
啟動完成 Skype 後的視窗畫面如下圖：



圖 5-3 Skype 視窗介面

等 Skype 啟動完畢後，再回到第 1 個步驟，點選「啟用 Conference」的按鈕。如果是第一次啟動「線上會議」的功能，則會出現 Skype 的保護機制視窗，此視窗會詢問使用者是否允許講解手程式控制 Skype。

因為講解手有使用者 Skype API，因為 Skype API 會操控到 Skype 主程式的行為，所以才會出現這個視窗透過使用者的選擇，保護 Skype 主程式。

只要選擇第一個選項「永久開放此程式使用 Skype」，之後再用講解手啟用線上會議的功能，就不會再出現這個詢問視窗。



圖 5-4 Skype API 驗證保護機制

此時會自動連線至中介伺服器，成功連線後，即會出現線上會議準備視窗，下圖分兩個主要的部分：(1)左手邊的會議室列表 (2)右手邊的使用者列表。只要有新的使用者連線至中介伺服器，使用者列表就會自動更新。而當有使用者建立會議室時，會議室列表也會自動更新。

此時主席要建立會議室，即可按下面標示的「建立會議室」按鈕。

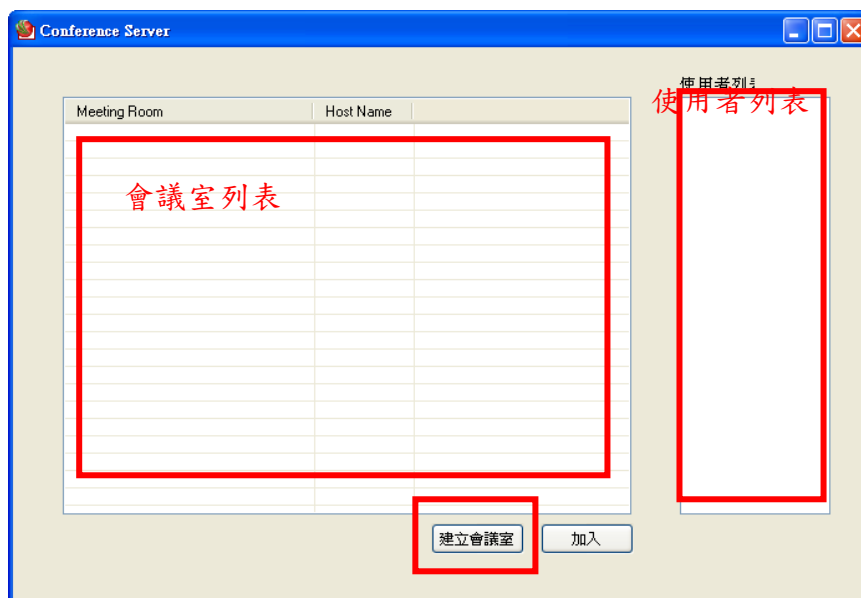


圖 5-5 線上會議前置準備畫面

此時會出現一個視窗，請使用者填入會議室的名稱以及密碼，不一定要填寫密碼。第三個欄位是選擇會議進行主軸的投影片檔，或是選擇已經過講解手解析過的專案檔(\*.elproj)。

主席開啟完成會議室後，即可開始等待使用者連線加入。

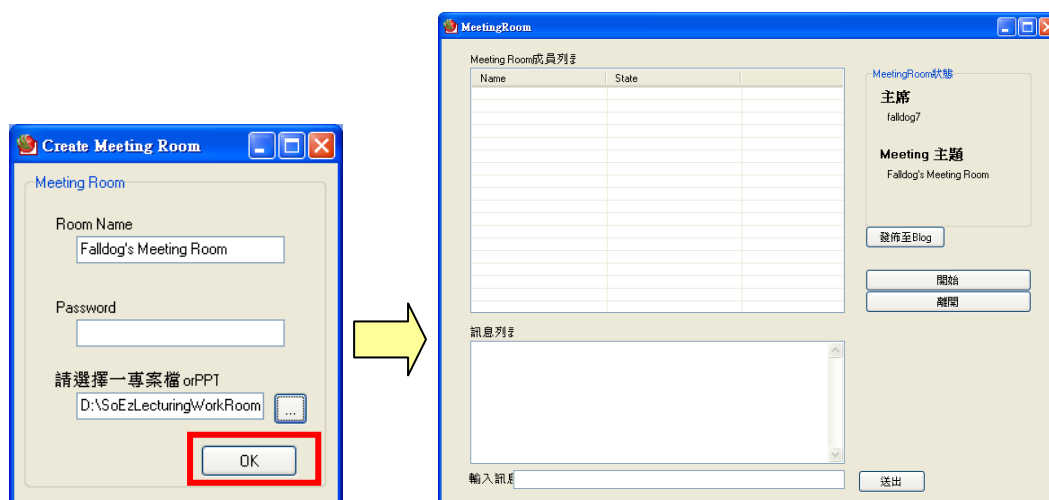


圖 5-6 主席開啟會議室

以上為主席開啟會議室的流程，以下是使用者加入會議室的流程：

使用者在連線至中介伺服器後的畫面，只要點選要在會議室清單中，選擇一間會議室，然後在下圖標示的「加入」按鈕，點選後即可進入會議室。

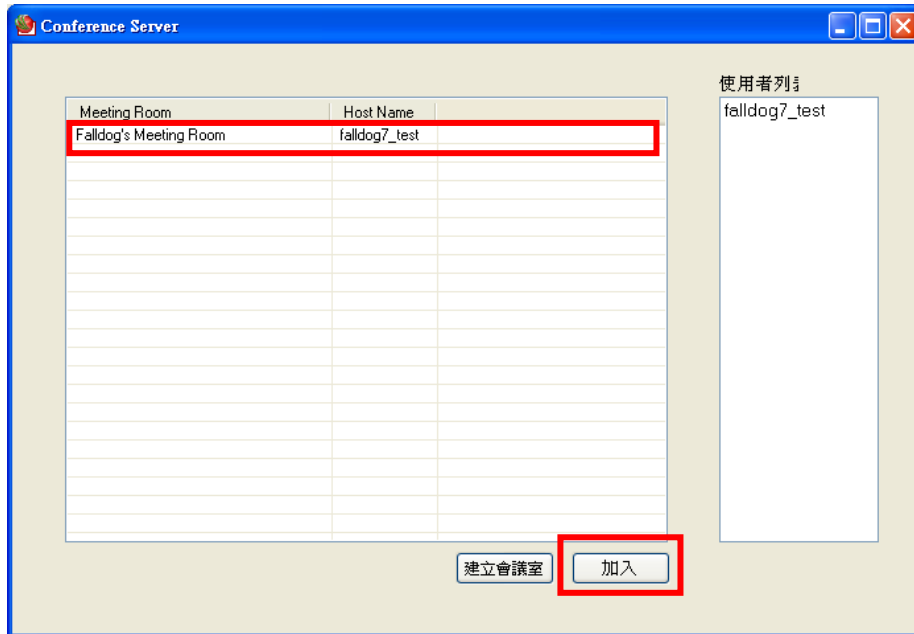


圖 5-7 使用者選擇會議室加入

## 5.1.2 進入會議室後

當一般使用者加入別人開好的會議室時，會開始下載主席上傳的投影片檔，或是講解手的專案檔。此時會出現一個「完成進度 Bar 的視窗」，顯示目前下載的進度。

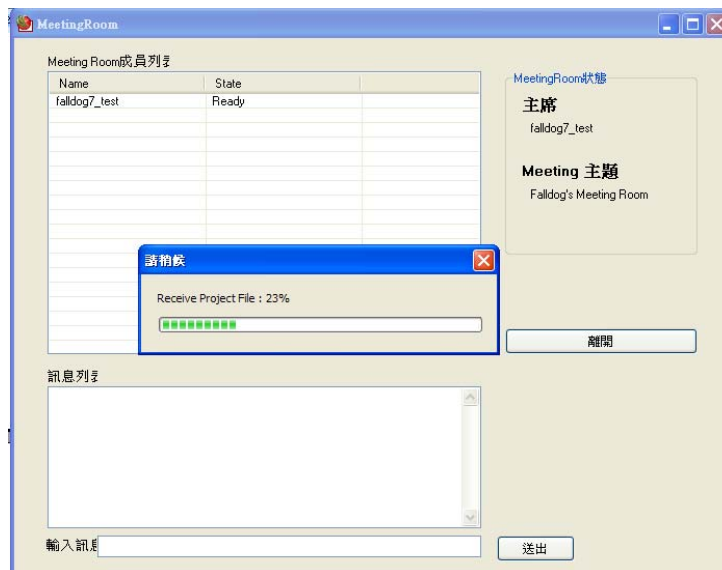




圖 5-8 會議參與者加入會議室

會議參與者看到的會議室畫面如下圖中的上半部，主席看到的畫面如下圖中的下半部。

每個人都會看到，左上方有會議室的成員列表，以及目前使用者的準備狀態。下方有文字聊天的欄位。右上方是此間會議室的狀態，包括主席名稱以及會議室的主題名稱。

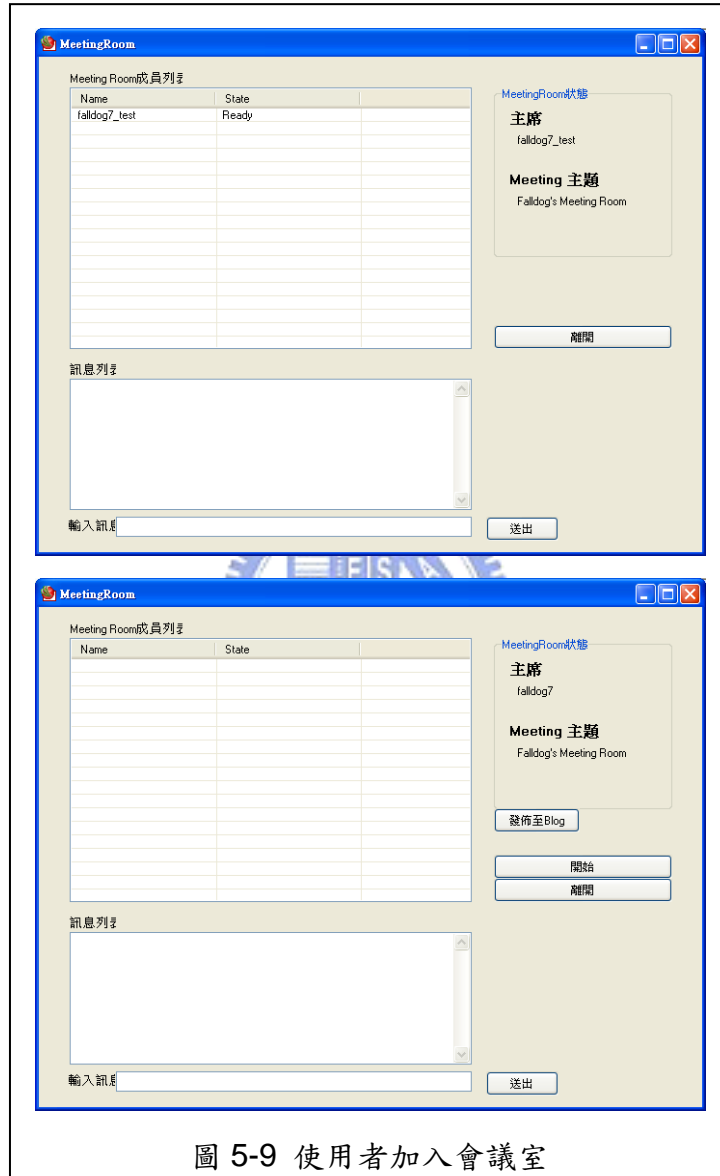


圖 5-9 使用者加入會議室



可以操控整個會議的流程的進行。



圖 5-12 核准/禁止發言權

使用者的滑鼠可以移到右下方的工具列上，就會顯示出可用的工具有哪些(如左半圖所示)。或是直接點擊滑鼠右鍵，即可看到功能選單，選擇使用者所需要的功能(如右半圖所示)。



圖 5-13 講解筆工具

使用者可以在投影片畫面上，進行講解筆的塗鴨，而這一些塗鴨的過程，全部皆會記錄下來。

## 研究動機 (2)

- 目前有許多公司已製作出線上會議系統
  - 但大部分的錄製功能都是直接擷取螢幕畫面，錄製出來的檔案都相當的大，相對的就需要更多的儲存硬體裝置來滿足這個需求
- 多媒體講解錄製工具 - 智勝國際講解手
  - 並非直接將螢幕畫面錄製成為影片檔，而是將投影片檔解析後，轉換為自訂的格式，再分別錄製投影片、畫筆、聲音的流程
  - 錄製後的檔案size較小
  - 缺乏多人線上即時錄製功能
- 網路通訊對話工具 - Skype
  - 時下熱門的VOIP網路電話軟體，語音通訊採用P2P技術
  - UDP傳輸聲音
  - 傳輸速度快、音質好
  - 有開放使用API



圖 5-14 錄製畫面

## 5.3 會議結束

### 5.3.1 發佈多媒體錄製檔

會議結束後，主席回到一開始進入會議室的畫面，此時，可選擇將剛才會議討論過程的多媒體錄製檔，發佈至 Web 2.0 Blog 平台，如下圖的標示處。

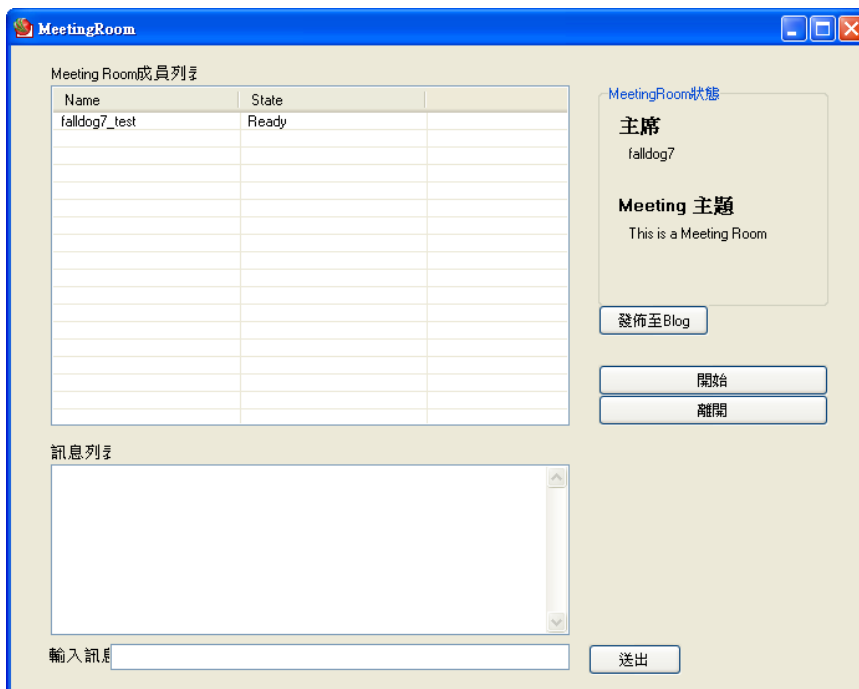


圖 5-15 會議結束後畫面

此時會先出現 Web 2.0 Blog 樣版國度的使用者驗證畫面，只要輸入使用者的 ID 以及 Password，即可開始上傳的動作。





圖 5-18 發佈選擇主題

### 5.3.2 觀看多媒體錄製檔

發佈完成後，會跳出多媒體錄製檔，發佈後在網頁上的連結，即可點入觀看。



圖 5-19 Web Blog 發佈網頁

點進該連結後，會秀出之前設定的發佈檔資訊。點選下圖標示的連結即可開始觀看會議錄製的過程。



圖 5-20 講解檔網頁畫面



此時會執行講解手播程式，下載多媒體發佈檔，即可開始進行會議錄製過程的播放以及瀏覽。

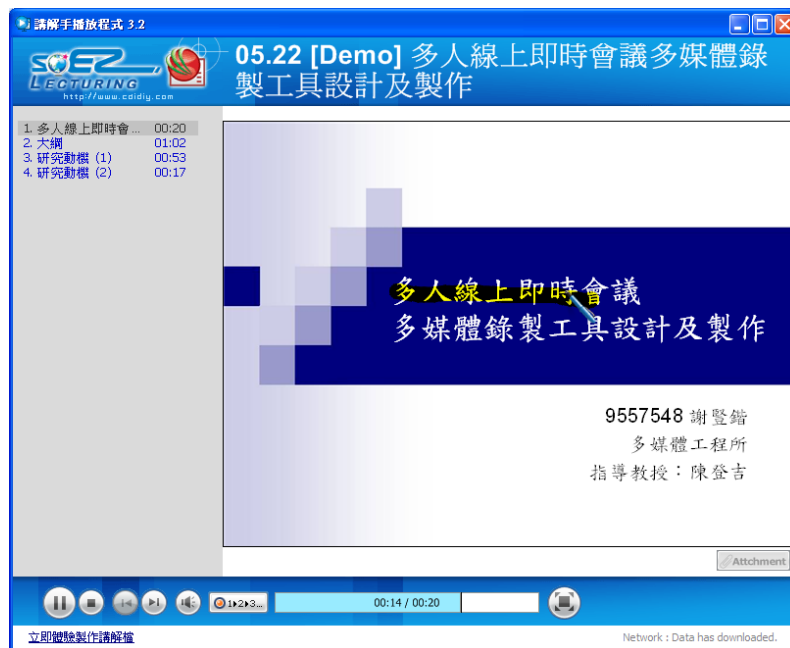


圖 5-21 講解手播程式畫面



# 六、結論

## 6.1 總結

本研究針對目前已有的多人線上即時會議系統，在多媒體錄製工具方面的缺失與不足，進行改良。因而拿取智勝國際公司開發的多媒體錄製工具－講解手，進行功能性的擴充，補足講解手工具在多人線上會議系統的不足之處。

此系統能將每一次會議討論的過程，完整地記錄下來，儲存成多媒體錄製檔。更可以透過播放多媒體錄製檔，完整重現當時會議討論的過程。這樣子會議記錄的功能，實是許多公司或是學校所需要的功能。

本研究實作出一個多人線上會議系統，以及附屬多媒體錄製工具之功能。並具備下列幾項特點：

- (1) 透過 Skype 的整合，群體線上會議的使用者可在 WAN 的環境下進行線上會議，會議的系統可將整個會議過程整合成一個多媒體會議記錄檔，並上傳到某一特定的社群供他人參考。
- (2) 比較其他現有的線上會議系統，本系統可以完整呈現 PPT 檔簡報功能，以及較小的多媒體錄製檔案大小。
- (3) 錄製後的多媒體錄製檔，可以上傳至 Web2.0 的 Blog 平台進行管理，讓使用者更容易分享此錄製檔。

## 6.2 未來發展方向

對於本系統尚有一些不夠完備之處，以下列出幾個尚待改進的項目：

1. 在講解手與中介伺服器的溝通的封包訊息，為了避免第三者竊取封包得到重要資訊，加強雙方溝通的安全性。可以建立 SSL(Secure Socket Layer)來處理封包訊息的傳送。
2. 在會議進行中，每個使用者封包傳遞的方式可以改為使用 P2P(Peer to Peer)的方式，降低中介伺服器的負擔。
3. 在會議進行中，可以增加傳遞使用者方的視訊畫面內容。

# 參考資料

- [1] Skype API,  
<https://developer.skype.com/DevZone>
- [2] Microsoft MSDN,  
<http://www.microsoft.com/taiwan/msdn/default.msp>
- [3] Wikipedia 中文維基百科,  
<http://zh.wikipedia.org/w/index.php>
- [4] DirectX,  
<http://msdn.microsoft.com/en-us/directx/default.aspx>
- [5] Delphi K. Top討論區,  
<http://delphi.ktop.com.tw/board.php>
- [6] Time研究室, C++ Builder 6 完全攻略, 金禾出版社, 2003
- [7] 智勝國際科技,  
<http://www.caidiy.com/>
- [8] Netmeeting,  
<http://www.netmeeting.org/>
- [9] 毅上科技, MultiVideo,  
<http://www.digienjoy.com/>
- [10] Cisco - WebEx,  
<http://www.webex.com/index.html>
- [11] 講解手, soEzLecturing,  
<http://www.caidiy.com/download/soezlecturing.htm>
- [12] 智勝國際Web 2.0 Blog – 樣版國度,  
<http://www.soezblog.com/>
- [13] Skype Community – Skype Public API,  
<http://forum.skype.com/index.php?showforum=16>
- [14] Skype API Document,  
<https://developer.skype.com/Docs/ApiDoc>
- [15] Skype4Com Reference,  
<https://developer.skype.com/Docs/Skype4COM>
- [16] Ubuntu正體中文站,  
<http://www.ubuntu.org.tw/>
- [17] GCC – the GNU Compiler Collection,  
<http://gcc.gnu.org/>



- [18] MPEG Home,  
<http://www.mpeg.org/>
- [19] 程式設計俱樂部 – Programmer Club,  
<http://www.programmer-club.com/index.asp>
- [20] MSN – Windows Live Messenger,  
<http://get.live.com/zh-cht-tw/messenger/overview>
- [21] Yahoo!即時通訊,  
<http://tw.messenger.yahoo.com/>
- [22] C++ Reference,  
<http://www.cppreference.com/>
- [23] CPlusPlus,  
<http://www.cplusplus.com/>
- [24] Gogrok,  
[http://www.gogrok.com/zh\\_TW/index.html](http://www.gogrok.com/zh_TW/index.html)
- [25] Co-Life,  
<http://colife.nchc.org.tw/index.aspx>

