

國立交通大學

資訊科學與工程研究所

碩士論文

整合式群組即時通訊系統之設計與實作

Design and Implementation of a Group-based Integrated Instant
Message System

研究生：劉育宗

指導教授：曾建超 教授

中華民國九十六年六月

整合式群組即時通訊系統之設計與實作

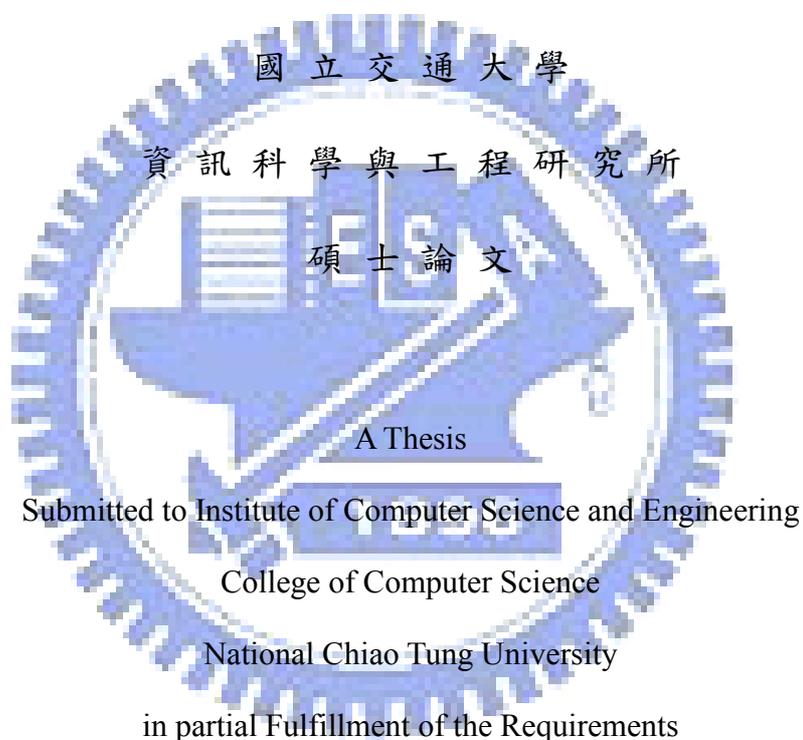
Design and Implementation of a Group-based Integrated Instant Message System

研究生：劉育宗

Student：Yu-Tsung Liu

指導教授：曾建超

Advisor：Chien-Chao Tseng



for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

整合式群組即時通訊系統之設計與實作

研究生：劉育宗

指導教授：曾建超 教授

國立交通大學八訊學院資訊科學與工程研究所

摘要

本論文以群組為基礎並整合時下多種即時通訊網路，設計與實作一個整合性群組通訊平台，讓使用者可以藉由較適當的通訊系統發送訊息給其他使用者。隨著資訊產業、網際網路的發展，資訊科技早就已經是生活的一部份了。資訊科技的最終目的，就是將各式各樣的資訊，帶到每個人的手中。資訊科技普及，網路技術發達，各種通訊方式百家爭鳴，對於一個使用者，要同時面對各式各樣的通訊管道，不僅要花時間去適應，也需要更改原有的使用習慣。

因此我們設計與實作出一套整合性群組通訊系統。本系統以群組為基礎，允許使用者建立通訊群組，發送群組訊息，並且整合即時通訊網路，利用現今廣為使用的即時通訊軟體，將訊息傳送給所有的群組成員，藉由本系統的幫助，使用者的訊息可以傳送給不同即時通訊系統的使用者。本系統除了可以提供給使用者、系統開發者建立通訊群組、發送群組訊息外，另外還針對每個通訊群組、每個使用者提供偏好設定，讓系統運作的更符和使用者的期望。

此外，本群組通訊系統亦可與其他應用程式整合，做為應用程式的通訊平台以開發新型態的網路服務。

Design and Implementation of a Group-based Integrated Instant Message System

Student : Yu-Tsung Liu

Advisor : Dr. Chien-Chao Tseng

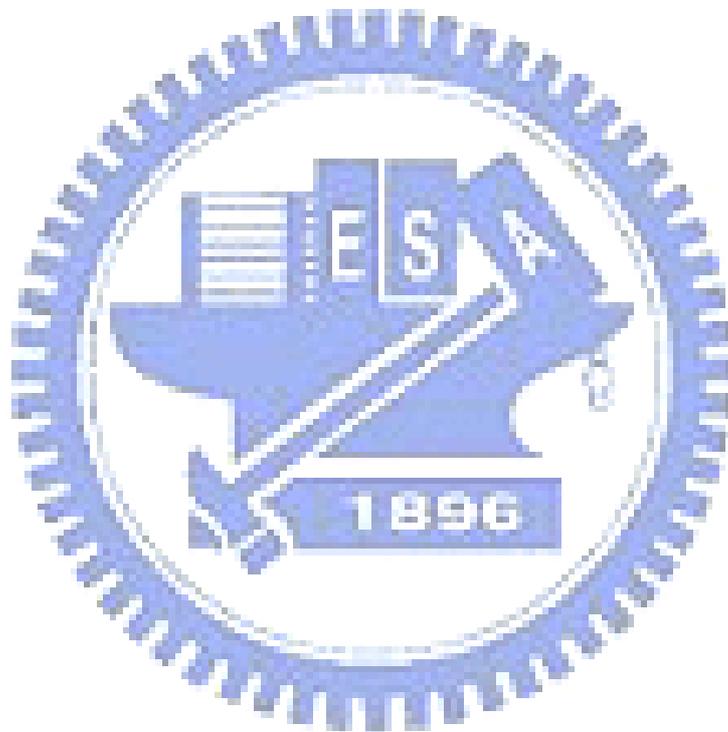
Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

In this thesis, we design and implementation a group-based communication platform that integrates different instant message systems (IMS), such as MSN or Skype, and enables a user to communicate with other users via the most appropriate IMS. With the advance of network and device technologies, people can now communicate with each other through various communication vehicles. However with so many different communication systems in use, a person may need to maintain many different communication identities, such as PSTN or PLMN numbers, and MSN or Sktpe IDs, of his contacts on various communication systems. Furthermore, the person needs to equip his device with appropriate software or hardware so that he can use an appropriate communication vehicle to communicate with a corresponding person. Therefore, we design and implemented an integrated communication system that can help a user to organize the identities of his contacts, and provide a one-click communication interface for the user to communicate with a contact through the most appropriate communication vehicle.

In this thesis, we further extend the integrated communication system to support group-casting. With the extension, a user can set up his own

communication groups and send group-casting messages to a designated group of contacts by using the most appropriate communication vehicle for each contact. As a consequence, our system can send messages to a group of members who belong to different IMSs. Furthermore, our system can provide mechanism for a user to set up the preference of the user or a group. Besides, it also provides interfaces for other applications to send group messages effectively. The experimental results show that the groupcasting mechanism can indeed deliver messages to a group of users on different IMSs.



致謝

研究所兩年過的很快也很充實，在這段期間，最主要是要感謝我的指導教授曾建超教授，在研究上、在報告上以及求知的方式等給我的建議、指導與輔導，帶領我完成兩年研究生生涯，在論文的撰寫期間，曾建超教授的多方指導及殷殷教誨，給了我許多重要的意見，使得本論文得以順利的完成，在此致上無限的謝意。

還要感謝在這兩年中修課的教授，在這兩年修課當中，各位教授，認真辛苦，費盡心思的授課與教導，讓我受益良多。

此外，感謝所有我的師長、同學們。感謝你們的熱心幫忙與耐心教導。感謝我的朋友們，在我面對壓力時願意傾聽並給予鼓勵。

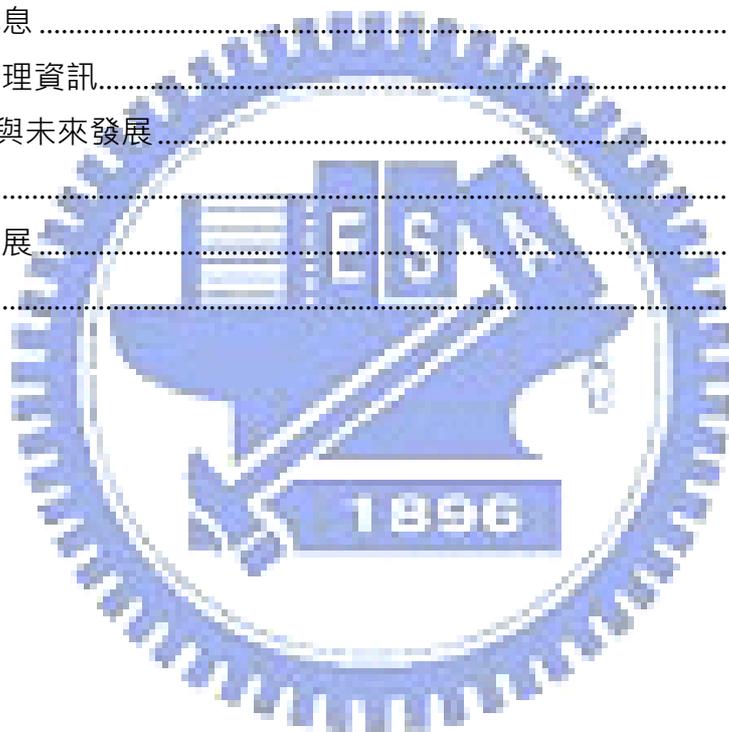
最後，感謝我的家人，特別是養育我、栽培我的父母，讓我全心全意地完成學業與論文，在此獻上我最衷心的感謝。



目錄

摘要.....	I
ABSTRACT.....	II
致謝.....	IV
目錄.....	V
圖目錄.....	VII
表目錄.....	VIII
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 研究目標.....	2
1.2.1 群組管理.....	2
1.2.2 喜好設定.....	2
1.2.3 群組訊息.....	3
1.3 章節概要.....	3
第二章 背景與相關研究.....	4
2.1 即時通訊系統.....	4
2.2 YAHOO! MESSENGER 和 MSN MESSENGER 的整合.....	4
2.3 LIBPURPLE (LIBGAIM).....	5
2.4 XMPP AND SIMPLE.....	6
2.5 CMS.....	7
第三章 系統功能需求分析.....	9
3.1 伺服器端系統之功能需求.....	9
3.2 系統功能設計與實作上之考量.....	11
3.3 系統應用實例示意.....	11
第四章 系統設計與實作.....	13
4.1 系統架構概觀.....	13
4.2 伺服器端之系統設計.....	15
4.2.1 伺服器端架構.....	15
4.2.2 User and Group - The Data on Server.....	15

4.3 系統元件介紹.....	18
4.3.1 Profile Management.....	18
4.3.2 Group Management.....	22
4.3.3 Groupcasting.....	25
4.3.4 與其他系統合作.....	27
4.3.5 資料庫結構.....	30
4.3.6 AJAX 和 Remote procedure call 的使用.....	36
第五章 實作成果演示.....	38
5.1 WEB 使用者介面.....	38
5.2 偏好設定.....	40
5.3 群組訊息.....	41
5.4 群組管理資訊.....	42
第六章 結論與未來發展.....	43
6.1 結論.....	43
6.2 未來發展.....	44
參考文獻.....	45



圖目錄

圖 1 Libpurple	5
圖 2 Jabber 架構圖.....	7
圖 3 CMS Architecture.....	8
圖 4 Global Architecture.....	13
圖 5 伺服器端架構	15
圖 6 User 和 Group 關係圖.....	16
圖 7 User profile metadata 產生 user interface 和 group profile.....	19
圖 8 User profile metadata	20
圖 9 User profile.....	21
圖 10 Group profile.....	21
圖 11 User Level.....	22
圖 12 Context-aware autogroup 流程.....	23
圖 13 手動邀請加入通訊群組的流程	24
圖 14 IM Controller.....	25
圖 15 網頁使用者介面.....	38
圖 16 通訊群組工具列-我所建立的通訊群組.....	39
圖 17 通訊群組工具列-我所加入的通訊群組.....	39
圖 18 成員資訊.....	39
圖 19 偏好設定.....	40
圖 20 輸入群組訊息	41
圖 21 接收訊.....	41
圖 22 等待回覆的要求.....	42

表目錄

表 1 CMS 資料庫.....	31
表 2 聯絡人資訊.....	31
表 3 Communication Group Administrator.....	32
表 4 通訊群組成員	33
表 5 需要 autogroup 的通訊群組.....	33
表 6 等待回覆的要求	34
表 8 等待回覆的要求-要求加入群組 (例子)	35
表 9 狀態訊息	35
表 10 通訊群組成員數量.....	36



第一章 緒論

1.1 研究動機

隨著網路的普及，網路在日常生活中所扮演的角色越來越吃重，隨之而起的，是各式各樣的網路增值服務，這些服務，帶給我們許許多多的知識和資訊，資訊科技已經完全融入我們的日平生活中了。

對於每個服務而言，其最終目的就是將寶貴的資訊送至使用者的手上，因此，每個服務都必需有一套傳訊方法。站在使用者的立場，每註冊使用一項服務，便需重新適應其通訊方法。倘若我們同時使用多個服務時，同時面對各式各樣的通訊管道，這就可能成為一種負擔。此外，以系統服務的開發商的觀點來看，每次要開發新的系統便需重新設計開發一套新的通訊方式，或是再次重新設計介面，無疑也是大費周張，平白浪費人力。

生活在這個資訊爆炸的時代，本身可能也有許多資訊、消息，想和朋友家人或者特定的某些人分享。當然，要將訊息傳送到對方手上，方法十分地多，再加上各種即時通訊系統蓬勃發展，每個人對此類的通訊軟體需求不同，也有各自習慣及偏好的軟體，因此接收訊息的一方並不一定是屬於相同的通訊網路，可能有些人是使用 Email，有些人是使用 MSN Messenger，亦有些人是使用 Skype，甚至，有些人會因不同的場合，使用不同的通訊軟體。如此一來，要將訊息傳送到這些人的手上就變得複雜起來。

可能下面這個情況會有似曾相似的感覺：假設今天我有一些資訊想要和朋友 A、B、C、D 四個人分享，A 跟我都同樣使用 MSN Messenger，可以簡單輕易的發送訊自給 A，而 B 使用 Skype、C 使用 ICQ、D 又是另一套即時通訊軟體，偏偏 B、C、D 三人所用的即時通訊軟體我都沒在用，為了能夠把訊息傳送給 B、C、D 三人，這

時就必需安裝 B、C、D 三人所使用的即時通訊軟體，或者另尋其他管道發送訊息，結果是既耗時又費事。更甚之，安裝了那麼多的即時通訊軟體就僅為了這次的通訊。

因此，我們希望能有一套系統、一個平台，包含群組通訊的功能，並且整合現有的即時通訊，讓使用者能夠簡單輕易地發送訊息給群組的成員，而這些成員並不限定於使用相同的即時通訊軟體。並且支援使用者設定自己的偏好，如第一優先傳送的通訊網路等。本系統除了給一般使用者傳送群組訊息外，亦可應用於公司、學校的場景，例如發送公告等。

1.2 研究目標

我們的目標是發展出一個以群組為基礎，整合即時通訊網路的系統，作為一個群組通訊的平台。系統除了發送群組訊息外，並希望能保有每個人的喜好設定。

目標與主軸有三：(1)群組管理、(2)喜好設定，以及(3)群組訊息。

1.2.1 群組管理

我們希望以社群的概念讓使用者能夠建立自己的通訊群組，群組的成員間可能會有某種的關聯性，例如都是網球社的社員等。因此，系統除了提供一般的加入退出群組外，我們另外希望能有一個容易及快速的機制，利用事先設定好的條件，來幫助使用者選擇、管理群組成員。

群組成員的組成，主要有兩個來源：(1)系統的使用者。(2)使用者本身的聯絡人。無論成員的來源為何，只是身為某個通訊群組的成員，都可以透過系統，將資訊發送給該群組的成員。

1.2.2 喜好設定

因為每個人都會有自己的偏好及要求，也會有不同的使用習慣，諸如喜歡以 MSN

做為通訊的媒介、不希望隨便被別人加入群組等。所以由系統做統一的設定或套用統一的行為模式都不是一個好方法。

對於使用者所建立的通訊群組，我們希望每個通訊群組可以有各自獨立的喜好設定。這些設定包含：通訊群組的基本資料、收到使用者要求加入時的動作等。

因此我們希望能讓使用者可以設定自己的喜好，以及設定通訊群組的喜好，而往後系統的所有動作，都會根據使用者的設定，以使用者所期望的結果來設計系統，讓使用者能夠擁有更好的使用體驗。系統會將使用者及通訊群組的喜好設定，分別存在使用者及通訊群組的 profile 中。

1.2.3 群組訊息

這個部分，我們希望可以把訊息傳透過各種不同的通訊網路將訊息傳送出去，例如 MSN、Skype、手機簡訊等。

因此，保持新增通訊方式的可擴充性，便顯得十分重要。在這邊，主要希望能設計出一個具有高度擴充性的通訊平台，整合各種通訊管道，使我們在傳送訊息時，能夠依照每個人的意願，正確無誤的送達。

1.3 章節概要

在第一章，描述本論文的研究動機與目標；第二章則是說明相關背景及相關研究；第三章根據我們的目標來制定詳細的系統需求；第四章開始介紹本系統的設計、系統元件及實作；第五章展示本系統實作後的成果；最後一章，闡述本論文之結論以及未來期望之發展方向。

第二章 背景與相關研究

2.1 即時通訊系統

即時通訊(Instant Message, IM)系統可說是現今最熱門的網路商品之一，不管是否同意，即時通訊確實已走入你我的生活中，亦有凌駕 e-mail 之趨勢成為最受大家歡迎的訊息傳遞軟體。即時通訊系統已經不僅僅是大家聊天的工具，更漸漸地成為公司行號中的員工與客戶互通訊息的主要媒介之一。目前市面上一般的即時通訊系統，大都具有下列特色：

即時性：不同於以往的方向，即時通訊系統提供了第一時間內的溝通，省去了一來一往耗費大半天的時間。

便利性：聯絡人資訊一目瞭然地顯示出來，想與哪個聯絡人交談一點即可，省去了電話通知的費用，更節省了時間。

娛樂性：除了一般的通訊功能外，即時通訊系統所附加的娛樂功能也越來越多，如表情符號、傳遞動畫、視訊等。

毫無疑問地，即時通訊於公於私皆已經成為我們生活的一部份，然而市面上的即時通訊系統十分地多，也皆有各自的支持者，因此如何讓使用不同即時通訊系統的使用者彼此通訊，逐漸成為大家關注的焦點。

2.2 Yahoo! Messenger 和 MSN Messenger 的整合

2006 年第二季，Yahoo 和 Microsoft 宣布旗下的即時通訊軟體可以互通，亦即表示使用 MSN Messenger 的人可以加入 Yahoo! Messenger 的使用者作為聯絡人，並且可以相互傳送訊息。就目前看來，即時通訊的整合是個趨勢，但又困難重重，

Yahoo! Messenger 和 MSN Messenger 皆有不少的市占率，這兩家公司的合作，對市場的影響不可謂不大。

毫無疑問地，這的確帶給我們許多方便，MSN Messenger 的使用者，不必為了跟使用 Yahoo! Messenger 的朋友聊天、談事情而特別安裝 Yahoo! Messenger，Yahoo! Messenger 的使用者也是相同的道理。能夠使用平常習慣的即時通訊軟體、習慣的介面，取代了重新安裝即時通訊軟體、取代了重新適應介面，無疑的這是每位使用者所希望的看到的，的確帶給了每位使用者全新的感受。

然而在實際使用後，我們可以發覺即使這兩大即時通訊系統互通了，卻仍然有他的限制在。有許許多多的加值服務，像是多方會談、自定表情符號、語音通話等，幾乎都不能使用。簡單來說，MSN Messenger 與 Yahoo! Messenger 的互通，目前只支援到簡單的文字訊息的交換。

2.3 Libpurple (libgaim)

Libpurple 舊稱為 libgaim，是一套支援多種即時通訊 protocol 的函式庫。利用 libpurple 開發出來的應用程式可以同時跟多種即時通訊的聯絡人傳送訊息，但必須注意的，即使是同一個聯絡人，但分別使用了 2 種不同的即時通訊系統，在 libpurple 開發出來的程式中，會被視為 2 個不同的聯絡人。

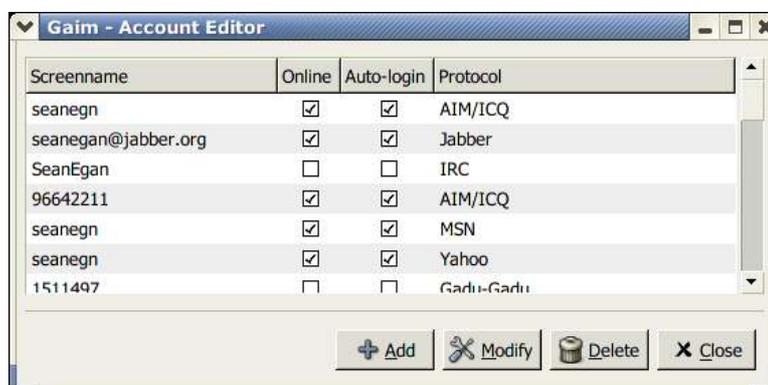


圖 1 Libpurple

如圖 1 所示，使用 libpurple 所開發出來的應用程式，可以說是同時包含 MSN、Yahoo! 等不同的即時通訊軟體，將這些通訊軟體整合在一隻程式中，省去使用者必須安裝多個程式的麻煩及困擾。但如同前面所說的，若要同時跟 2 個朋友交談，而又分別使用不同的通訊軟體，那麼就必須把同樣的話重覆打兩次。針對使用不同即時通訊協定的聯絡人，彼此間並無法直接相互傳送訊息。

2.4 XMPP and SIMPLE

為了解決不同即時通訊系統間的互通性，在 IETF 中有兩個標準的通訊協定被提出來，分別為 Extensible Messaging and Presence Protocol (XMPP)[11] 和 SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE)[10]。

XMPP 和 SIMPLE 被視為現今即時訊息 (Instant Messaging) 與現狀資訊 (Presence) 技術的兩大陣營。有別於以往訊息系統 (如 Email)，採用先儲存再轉送的機制 (store and forward)，透過即時訊息及現狀資訊，我們可以容易的知道誰在線上並可以與之交談，若不在線上或者在忙碌狀態，就可能要改用別的溝通方式或者稍後再與之聯絡。

現今市面上的即時通訊服務有很多種，主要功能並無太大的差異，但在通訊上，皆採用獨立私屬的通訊協定，不同的 IM Network 間無法相互溝通。因此，若有一個標準的通訊協定，讓場商在開發應用程式時，有參考依循的標準，在不同 IM network 間相互溝通，會變得簡單許多。

Session Initiation protocol (SIP) 是一種傳遞信號 (signal) 的通訊協定，主要用在建立語音通話，通訊雙方的管理、協商，但 SIP 的文件中，並沒有指明 SIP 只能用於語音通話中，因此擴充 SIP 則是很自然的想法。SIMPLE 則將 SIP 導入即時通訊的領域。

XMPP 是一個基於 XML 的協定，制定標準的標籤(tag)，如 from、to、message 等，讓即時通訊系統的開發商遵循，同是支援 XMPP 這個通訊協定的即時通訊系統間，便可彼此通話，傳送訊息。

XMPP 為 Jabber 的核心技術。Jabber 既是一種基於伺服器(Server)，也是一種分散式的 IM Network，基於伺服器這點，是因為用戶端之間彼此不直接相連。而分散式是指 Jabber Network 並沒有一個中央官方的伺服器，實際上，每個人都可以在自己的網域架上一台 Jabber Server，並提供服務。

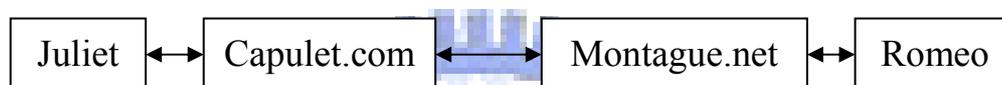


圖 2 Jabber 架構圖

圖 2 為 Jabber 服務的架構圖。在 Jabber 中，每個使用者以一個 JID 來做識別，JID 看起來就像是一般的 email，如 Juliet@Capulet.com，前半部是用戶名，後半部是 Jabber 伺服器的網域名，中間以@隔開。

圖 2 中，Juliet 和 Romeo 分別由 Capulet.com 和 Montague.net 這兩家廠商提供服務。當 Juliet 和 Romeo 相互傳訊息時，不必擁有對方伺服器的帳號，也不必成為對方業者的會員。Juliet 想要傳送訊息給 Romeo 時，Juliet 所發送的訊息，會先傳送到自家的系統提供商 Capulet.com，再由 Capulet.com 將訊息轉發給 Montague.net，由於是使用標準的通訊協定，兩方的伺服器都能解讀封包，以進行溝通，最後 Montague.net 再把訊息傳遞給 Romeo，完成了整個訊息傳送的过程，Romeo 收到 Juliet 所傳送的訊息。

2.5 CMS

本實驗室學長所完成的系統，Contact Management System (CMS)主要的目的是將通訊錄(Windows Address Book, wab.exe)中的聯絡人資訊，與伺服器上的聯絡人資訊相互同步。無論是在 Windows Address Book 上操作聯絡人的增減，還是

在 CMS 所提供的 Web 介面操作，皆可透過 CMS 系統讓兩邊的聯絡人同步，達到一致。

此外，我們透過另一隻小程式，取得聯絡人的 MSN Messenger 和 Skype 的線上狀態並傳至伺服器，往後可以透過瀏覽器一目瞭然地掌握聯絡人的動態。圖 3 為 CMS 的架構圖。

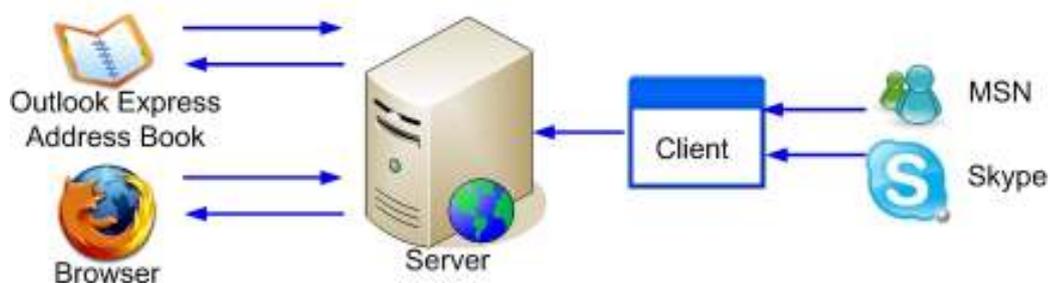


圖 3 CMS Architecture



第三章 系統功能需求分析

首先，我們先看個例子。場景是學校。

每個班級有數十名學生，可能每天都有一些注意事項或是家庭作業，這些訊息需要通知學生本人或者學生家長。因此系統必須要有機制將學生及學生家長組成一個通訊群組，方便老師交待事項、學生及家長們交換訊息。

網球社需要兩個通訊群組：(1)社團內部使用，成員限定為社員及指導老師，所交換的訊息可能為社團行政、訓練時程等。(2)開放給社團外部的人員使用，社團可能有些比賽訊息、活動宣傳等，此通訊群組開放給對網球社有興趣的學生或老師自由加入群組，分享球賽資訊。因此，系統必須能夠分辨通訊群組間的需求差異，以及對群組成員的控管。

建築物的閉館通知。校內的建築，系館、圖書館、教學大樓、行政中心等，皆不是 24 小時開放的，當一棟建築即將要關閉時，可以通知還在館內(建築內)的人員、學生、老師，請他們儘速離開，以免建築物關閉後，仍然有人還困在建築物裡面。因此，系統需要一個機制將在建築裡面的人員群成群組，方便通知。

再來，換個方向想想，當我們是接受訊息的人時，想必會希望系統能以我們最習慣的通訊網路來通知訊息，或者是使用最適合當下的通訊方式。因此系統需要知道使用者的喜好，甚至要知道什麼情況下要用什麼種的通訊網路。

綜合上述的內容，我們理想中的系統，應滿足下列的功能需求。

3.1 伺服器端系統之功能需求

1. 基於網頁的存取方式

我們希望能以 Web 的方式呈現系統。隨著 World Wide Web 的盛行，無論是用哪種裝置，個人電腦或是 PDA，瀏覽器(Browser)都可見其蹤影。而 www 也有其標準供大家遵循。因此，使用網頁的方式來呈現系統能夠跨足多數平台，滿足多數的使用時機。

2. 存取權限管理

由於通訊資料是屬於個人較隱私的部份，為了避免被不相干的外人得到資料，因此我們要有授權認證的機制，來確保存取資料的合法性。

3. 通訊群組管理

讓系統的使用者可以建立自己的通訊群組。包含建立、刪除通訊群組，及群組成員的選擇及管理。在通訊群組建立方面，需要一個額外的機制，一個能夠自動維護成員，自動選擇成員的機制，這是一方面是給使用者方便，另一方面則是考慮到類似閉館通知的通訊群組，某方面來說，這樣子的通訊群組並不會很在意其成員的來源、成員是誰，只要做到「通知」即達成這通訊群組的目的。

4. 個人、通訊群組喜好設定

每個人都自己使用習慣、每個通訊群組也可能有各自的偏好設定。因此，我們必需讓使用者、群組管理者能夠替個人、群組設定各別的喜好設定。讓系統以最合乎使用者的習慣，替他服務。

5. 結合即時通訊

結合現今市場上已有的即時通訊系統，讓使用者能以自己最習慣的方式接收資訊。甚至可以因時因地因情形進一步來選擇使用的即時通訊系統。如此一來，使用者不必重新適應新的軟體、新的介面，可以在適當的時機，使用適當的即時通訊系統。

3.2 系統功能設計與實作上之考量

本系統旨在整合，期望能夠支援多種通訊軟體及通訊管道。因此，我們在設計上，儘可能地保持彈性及擴充性，以便在後續的維護及持續開發上簡單化，方便往後的支援，同時我們以模組化的設計，將功能切割開來，每個系統元件有專職負責的部份，方便程式的維護及改進，也方便後續功能的開發，且減少對於原本系統的影響，將衝擊降至最低。

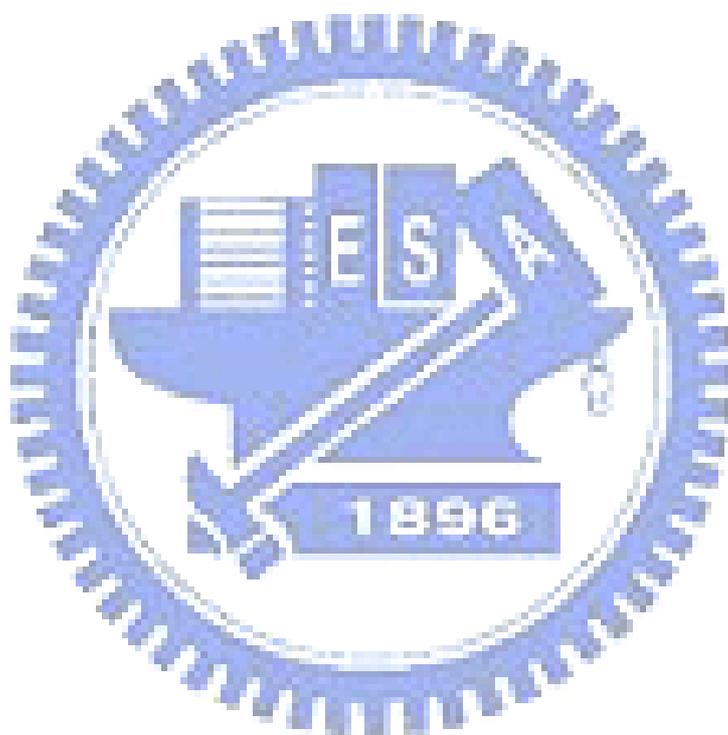
由於在設計之初，考慮到本系統除了可以自行運作外，亦希望本系統可以與其他系統共同服務，因此我們需要提供一個方便的介面讓外部系統存取，使其能夠完美的與其他系統結合。

3.3 系統應用實例示意

在這邊，我們再次利用本章開頭的例子，並更詳細的說明本系統的規劃及運作過程。

- ◆ 班級：一年一班的級任導師建立一個通訊群組，此級任導師即為此群組的管理者，並替該通訊群組設定偏好，「班級為一年一班的學生及家長」、「成員的加入需經過管理者同意」，接下來導師就可以透過這個通訊群組傳送訊息給一年一班內的學生及家長。
- ◆ 社團：網球社的社長希望建立兩個通訊群組，一個為內部社員使用，因此可以將群組偏好設定為「社團為網球社」、「成員的加入需經過管理者同意」，並且發出邀請，邀請社團的指導老師加入，此群組可能為社團行政、社員訓練等可以使用；另一個則為公開的通訊群組，讓對網球社有興趣的使用者自行加入，因此可以設定為「要求加入群組時自動同意加入」，若有比賽消息等，可以透過此群組發送訊息，讓希望取得消息的人有個獲知的管道。

- ◆ 閉館通知：以圖書館為例，每天晚上要閉館時，除了有廣播或是閉館音樂外，我們還可以透過本系統，利用即時通訊軟體、手機簡訊的方式通知館內的人員，因此通訊群組的設定可以是「使用者位置在圖書館內的使用者」，利用系統提供的機制(4.3.2.2 群組成員管理)，自動選擇群組成員，在圖書館將要關閉前，發送群組訊息，通知閉館。



第四章 系統設計與實作

4.1 系統架構概觀

本系統為一個基於網頁的系統，使用者透過瀏覽器和網際網路，即可使用本系統的服務。而系統在傳送訊息時，則會不同的情況選擇不同的傳訊方法，如此一來，訊息接收者可以使用多種不同的管道接收訊息，只要我們的系統支援它們。圖 4 為本系統的整體架構圖。

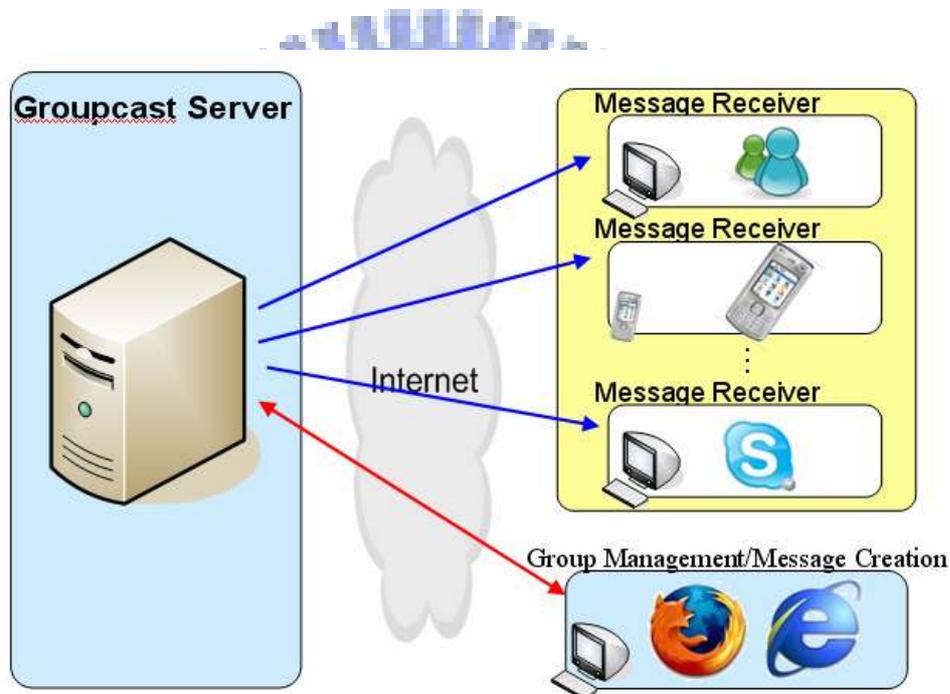


圖 4 Global Architecture

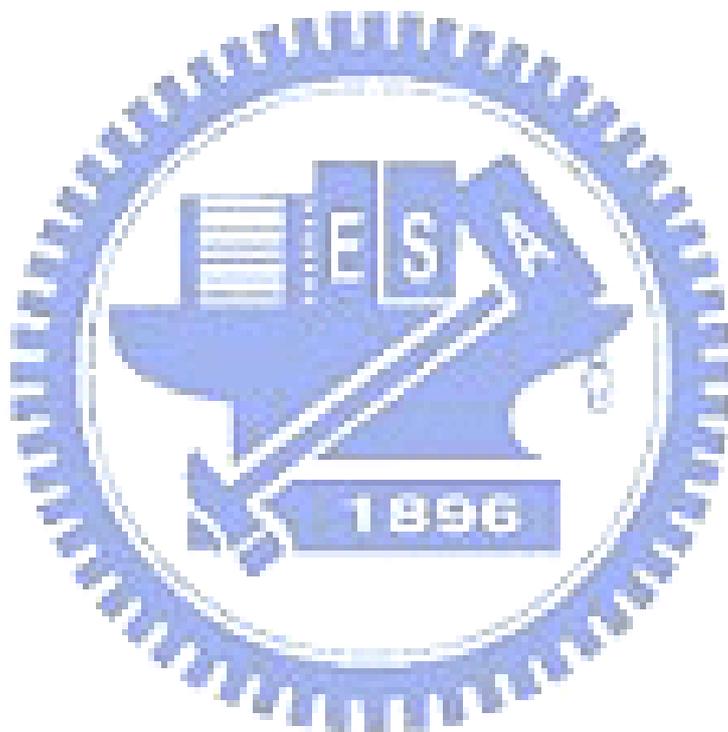
伺服器端以一資料庫做為儲存媒介，儲存使用者資料、通訊群組資料、群組成員。網頁伺服器則提供 web 的使用者介面。

我們亦開發出一組遠端程序呼叫(remote procedure call)，包裝了大部份系統的動作，如登入、登出、通訊群組管理(建立、刪除)、群組成員管理(手動邀請、context-aware autogroup)、群組通訊、等，在往後的維護、擴充等，只需針對此組 remote procedure call 下手即可。減少系統維護的成本。此外，在設計時，「透

過瀏覽器存取」這項功能我們亦考慮進去，以方便我們的網頁使用者介面使用，同時在與其他系統合作時，亦可使用 HTTP 的協定來呼叫我們的 remote procedure call。

系統使用者可以簡單的透過瀏覽器，就如同在瀏覽網頁一般，進行本系統的各项操作。諸如登入、登出、建立通訊群組、設定更改 user profile、group profile、群組通訊等。在實作上，仍以透過呼叫同一組的 remote procedure call 實現這些功能。

在群組通訊時，若接收的一方亦為本系統的使用者，系統在發送訊息時，會依照該名使用者所設定，使用設定為第一優先的通訊軟體作為通訊管道。



4.2 伺服器端之系統設計

4.2.1 伺服器端架構

根據前一節的整體架構來實作伺服器端的系統。在底層的資料中，儲存 profile(包含 user profile 和 group profile)、群組管理訊息、聯絡人資料、群組成員等，資料庫部份採用 mysql 實作，網頁伺服器為 apache。網頁使用者介面和 remote procedure calls 則使用 PHP 來實作。因此，若是第三方系統希望使用我們的服務，如發送群組訊息等，只須懂得 HTTP 協定就可以與我們的伺服器溝通。

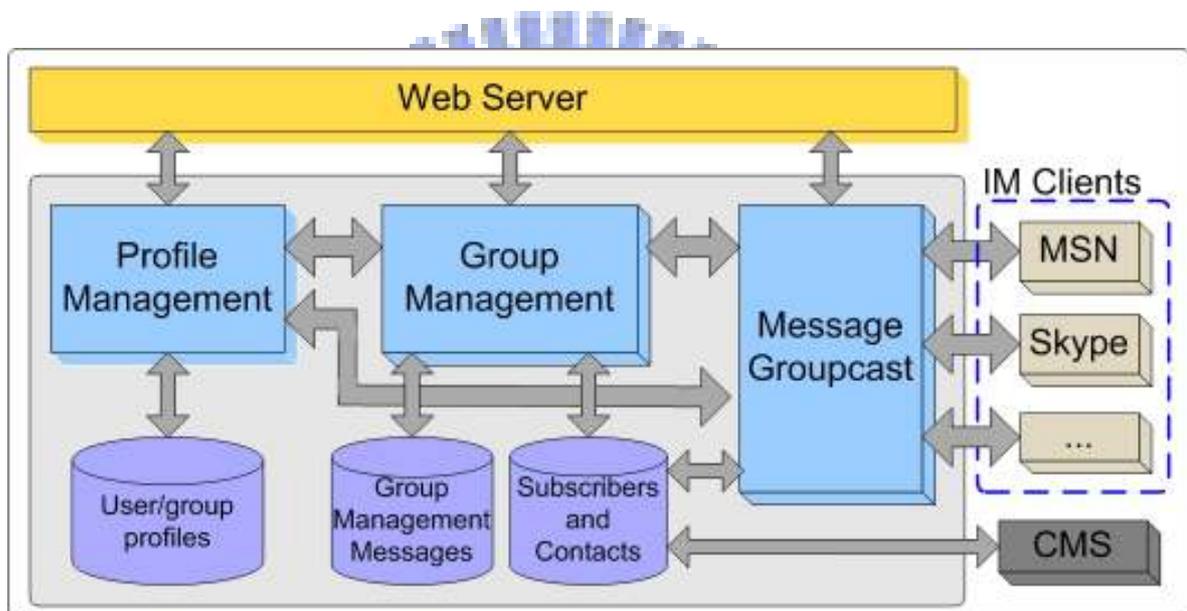


圖 5 伺服器端架構

圖 5 為我們伺服器端的架構，在有顏色的底框中的 components 是本系統實作的部份，為模組化的系統元件，分工完成各項動作並相互合作，下層則是儲存資料的資料庫系統。CMS 系統的聯絡人資料則是整合進我們的資料庫系統中，Message Groupcast 是系統中與 IM networks 合作的窗口，整合了多種不同的 IM networks，最後使用者可以透過瀏覽器使用、操作本系統。

4.2.2 User and Group - The Data on Server

在伺服器端，有兩項主要的資料：User 和 Group。

這邊的 User 指的是本系統的使用者，一名使用者，他會有一個他專屬的 user profile，並且允許建立一個或多個通訊群組，也可以加入一個或多個通訊群組。Group 在本系統中，指的皆為通訊群組(Communication Group)，最主要的作用便是發送群組訊息，在圖 6 中，更細分為「建立的通訊群組」和「加入的通訊群組」兩項。

圖 6 為 User 和 Group 的關係圖。我們可以看到，每位使用者會有四項基本的資料：聯絡人(Contacts)、User profile、建立的通訊群組(Created groups)、加入的通訊群組(Joined groups)。每個使用者所建立的通訊群組，包含了 group profile 以及其群組成員。而每個通訊群組主要有兩項基本資料：群組成員、group profile。

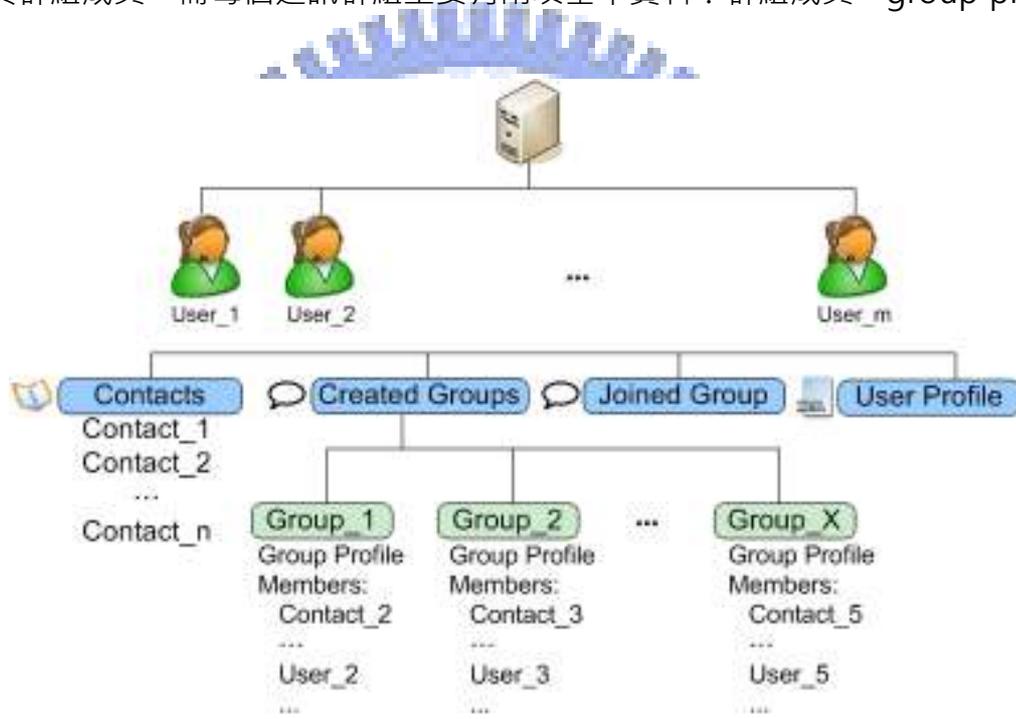


圖 6 User 和 Group 關係圖

聯絡人(Contacts)：這部份的資料是由 CMS 系統所維護。使用者可以利用這部份的資料同步 Address Book，無論出門在外，使用公家電腦，都可以正確無誤取得自己的聯絡人。在我們的系統中，聯絡人可以被加入通訊群組，成為群組的成員，在群組發訊時，聯絡人亦可收到我們所發送的訊息。

建立的通訊群組(Created groups)：使用者可以建立一個或多個通訊群組，使用者可以替屬於這分類的群組設立、修改 group profile，可以手動邀請成員，甚至可

以踢除已加入的群組成員，換句話說，使用者對於這個分類的通訊群組，有著完全的掌控權。

加入的通訊群組(Joined groups)：使用者所加入的通訊群組，可能是某個通訊群組管理者邀請的，也可能是自己要求加入的。因為並非是自己所建立，因此對於這個分類的通訊群組，只有「退出群組」可供使用者操作。

User Profile：主要儲存了使用者基本資料，及偏好設定。User profile 在系統中，伴演著重要的角色，所有跟使用者有關的動作，包含加入群組、群組通訊等，在做進一步的動作之前，系統都是先參考 user profile，根據 user profile 中使用者的設定來進行符合使用者期望的程序。詳細的 user profile 可參照 4.3.1。

Group Profile：群組的管理者替通訊群組設定偏好，如成員選擇條件等。在 group profile 中，主要包含了群組的基本資料及其偏好設定。詳細的 group profile 可能照 4.3.1。



4.3 系統元件介紹

我們將系統切成數個 module, 各 module 有其負責的工作, 亦相互溝通。如圖 5 所示, 有三個主要的 module : (1)Profile Management、(2)Group Management、(3)Message Groupcast, 並且與 CMS 系統彼此合作。

4.3.1 Profile Management

系統中, 主要有兩種 profile, 分別為 user profile 和 group profile。User profile 主要是儲存使用者的基本資料和偏好, 以及該名使用者對於群組通訊的相關設定, 如即時通訊軟體的優先權設定、有人將該使用者加為群組成員時的處理方式等。同理, group profile 儲存 group 的基本資料和其偏好設定, 如選擇成員的條件等。User profile 主要由該名使用者維護, 而 group profile 則由群組的管理者所維護。

這個元件主要負責的任務有二。(1) user profile 和 group profile 的管理, (2) profile 的比對(matching)。

4.3.1.1 Profile 管理

User profile 和 group profile 是這個元件的核心, 也在整個系統中, 占有極大的地位。系統中, 一有份文件, 稱之為「user profile metadata」, metadata 有點 server configuration file 的成份在, 但在我們的系統中, metadata 主要的作用是在描述 user profile 中, 所需的資料, 並且在 group profile 中, 提供設定自動群組時能夠設定的條件。如圖 7 user profile metadata 經過我們的 parser, 產生使用者介面以提使用者設定。這邊的使用者介面, 主要有二部份, 一個是使用者偏好設定, 二是 group profile 設定。因此, 若臨時希望在 user profile 中多一些欄位資料, 或是 context-aware autogroup 時多一些條件可供選擇, 我們只需透過修改 metadata 即可輕易達到, 而不須大符改動程式。基於整個系統的統一及國際化考量, 所有的 profile 及 metadata 以 Unicode UTF-8 作為編碼。

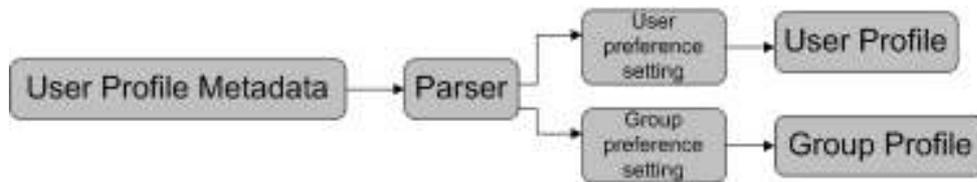


圖 7 User profile metadata 產生 user interface 和 group profile

圖 8 為 user profile metadata 的一個例子。主要分成三個大部份，分別由 info、communication 和 group 標籤所描述：

(1)info：描述在 user profile 中我們希望使用者提供的資料，以圖 8 為例，我們希望使用者提供姓名和地址，同時 info 這部份的資訊，也提供了 context-aware autogroup 的條件設定來源，舉例來說，這邊有姓名和地址，因此在設定選擇條件時，可以設成「姓名包含陳」，將姓陳的使用者，自動組成群組。「地址包含新竹」把新竹地區的使用者組在一個群組。當然，亦可以同時設定多個條件，如「姓名包含陳」AND「地址包含新竹」，新竹地區且姓陳的使用者。每個 label 標籤指明一個需使用者提供的資訊，裡面的標籤 datatype 說明這項資訊的資料型態，目前有 string(字串)和 int(數字)兩種，在設定 context-aware autogroup 時，若資料型態為 string，則會有「包含」、「不包含」、「等於」這三個運算子，若資料型態為 int，那麼就有會「等於」、「大於」、「小於」這三個運算子。

(2)communication：希望使用者提供的通訊方式，亦是目前系統所支援的通訊管道。使用者在這邊可以設定每個通訊軟體(管道)的帳號及優先權，系統會使用優先權最高的通訊管道傳送訊息。

(3)group：系統在群組方面所參考的依據，其結構與 info 十分類似，一個 label 標籤表示一個選項，透過這些選項的設定，系統會以使用者的期望來運作。以圖 8 為例，ReceiveInvite 是設定「收到某人邀請你加入群組」時的動作，有三個選項可供使用者選擇：accept、ask me、reject，當使用者選擇 accept 時，一旦收到邀請，系統會自動幫你接受邀請；同理，若是選擇 reject，系統則會幫你拒絕；若選擇的是 ask

me，收到邀請時，系統會通知使用者，使用者再根據自己的意願選擇接受或拒絕。

```
<?xml version="1.0" encoding="utf-8"?>
<profile type="userprofilemetadata">
  <info>
    <label>
      <text>Name</text>
      <description>Name</description>
      <datatype>string</datatype>
    </label>
    <label>
      <text>Address</text>
      <description>Address</description>
      <datatype>string</datatype>
    </label>
    .....
  </info>
  <communication>
    <im>
      <text>MSN</text>
    </im>
    .....
  </communication>
  <group>
    <label>
      <text>Receiveinvite</text>
      <description>Action when receive "Invite": accept/ask me/reject</description>
      <select>
        <option value="accept">Accept</option>
        <option value="askme" selected="yes">Ask me</option>
        <option value="reject">Reject</option>
      </select>
    </label>
    .....
  </group>
</metadata>
```

圖 8 User profile metadata

圖 9 為一個 user profile 的例子。根據 metadata，user profile 有三個主要部份，分別由 info、communication 及 group 標籤所描述，其內容主要就是由使用者設定後的數值，將其記錄在 user profile 中。

圖 10 為一個 group profile 的例子。同樣的也是記錄通訊群組的設定。如 join_request 標籤記錄有人要求加入群組時的對應動作，圖 10 中的 join_request 值為 accept，表示有人要求加入時，系統即接受該名使用者，將該名使用者加入此群組。而 Condition 標籤是 context-aware autogroup 時重要的參考，每個 group profile 可以擁有零至多個 condition 標籤，有多少 condition 標籤即表示這個 group 在 context-aware autogroup 時的條件有多少，condition 中的 field 標籤，其值必定是 metadata 中的某個 label。

4.3.1.2 Profile 比對

Profile matching 所要做的，就是將 group profile 和 user profile 拿來做比對，並回傳結果。至於要比對什麼，則視情況而定，倘若是 context-aware autogroup 時的成員選擇，那麼 profile matching 所要比對的，就是將 group profile 中成員的選擇條件與 user profile 中相關的資料逐一比較，以決定是否加入群組。

```
<?xml version="1.0" encoding="utf-8"?>
<profile type="user">
  <info>
    <Name>Davy Jones</Name>
    <Address>1001 Ta Hsueh Road, Hsinchu, Taiwan 300, ROC</Address>
    <Relation>Student</Relation>
    <Class>G1-C2</Class>
    <Club>Tennis</Club>
    <UserLevel>87</UserLevel>
  </info>
  <communication>
    <im>
      <type>MSN</type>
      <account>Davy_jones@msn.com</account>
      <priority>3</priority>
    </im>
    <im>
      <type>Skype</type>
      <account>davy_jones</account>
      <priority>3</priority>
    </im>
  </communication>
  <group>
    <AutoGroup>accept</AutoGroup>
    <ReceiveInvite>askme</ReceiveInvite>
    <Visible>yes</Visible>
  </group>
</profile>
```

圖 9 User profile

```
<?xml version="1.0" encoding="utf-8"?>
<profile type="group">
  <gLeaderUID>Uab73e2c673af9d13744ac2066a242a07</gLeaderUID>
  <gLeader>Jack Sparrow</gLeader>
  <gName>Tennis</gName>
  <gDescription>Tennis Club</gDescription>
  <autogroup_type>always</autogroup_type>
  <autogroup_condition_operator>or</autogroup_condition_operator>
  <condition>
    <field>Club</field>
    <operator>contain</operator>
    <value>tennis</value>
  </condition>
  <condition>
    <field>Club</field>
    <operator>contain</operator>
    <value>Tennis</value>
  </condition>
  <condition>
    <field>Club</field>
    <operator>contain</operator>
    <value>網</value>
  </condition>
  <auth_method>none</auth_method>
  <join_request>accept</join_request>
  <cgid>CGa18af37db43902d9d971Dc3696923002</cgid>
</profile>
```

圖 10 Group profile

4.3.2 Group Management

Group Management 主要負責兩個部份，一為通訊群組管理，包含通訊群組的建立與刪除等，二為群組成員管理，包括群組成員的選取等。

整個 group management 的運作，環繞在「profile」和「user level」這兩項。Profile 意指在 group management component 有所動作之前，皆需比對 profile 後才能動作，以確保所產生的結果是符合使用者所期待的。而 user level 的設定，是替系統的使用者做一個等級的劃分，不同的 user level 會有不同的權限和限制，一個最明顯的例子，就是 user level 高的使用者，可以對 user level 比他低的使用者進行 context-aware autogrouping 的動作。以圖 11 為例，照 user level 排序，由高到低依次為校長、老師、學生，所以在 context-aware autogroup 時，校長的對象可以是校長(自己)老師及學生，但老師的對象有老師及學生。

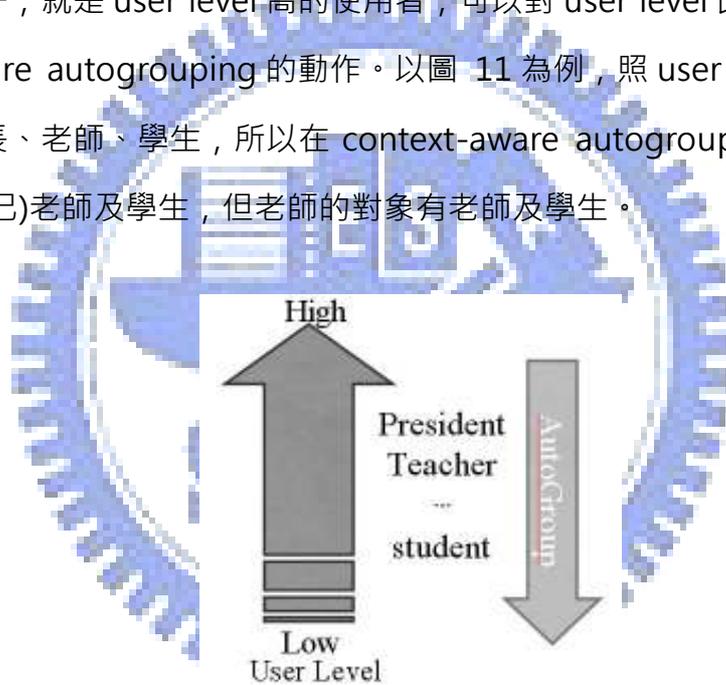


圖 11 User Level

4.3.2.1 通訊群組管理

這邊主要負責建立與刪除通訊群組，並且替每個通訊群組指定一個專屬的識別碼。在本系統中，每一個通訊群組會有一個識別碼，我們稱之為 CGID，CGID 為一 34 字元的字串，以“CG”開頭，取其 CGID 的前兩個字母。CGID 在通訊群組建立時，由系統產生並指定，CGID 具有唯一性，在系統中是獨一無二的，不會與其他通訊群組共享相同的 CGID。每個 CGID 代表著一個通訊群組，群組成員可能會有所更動、

群組設定也會被修改，但 CGID 自始至終永遠保持一樣。

4.3.2.2 群組成員管理

系統在選擇群組成員方面，主要分為二種方式：

1、Context-aware autogroup

根據通訊群組的 profile (group profile)中所指明的選擇成員的條件，由系統替通訊群組新增或刪除群組成員，context-aware autogroup 所新增的群組成員都會有一個註記，標明該使用者是透過 context-aware autogroup 機制所加入群組的。在此機制運作的過程中，群組成員可能是新增，但也可能是刪除，為了確保此機制的運作不會影響到手動邀請、要求所加入的成員，所以需要替經由 context-aware autogroup 所加入的成員做註記，一旦需要刪除群組成員時，系統會先確認該使用者加入群組的管道為何，接下來才決定下一步的刪除使用者是否執行。

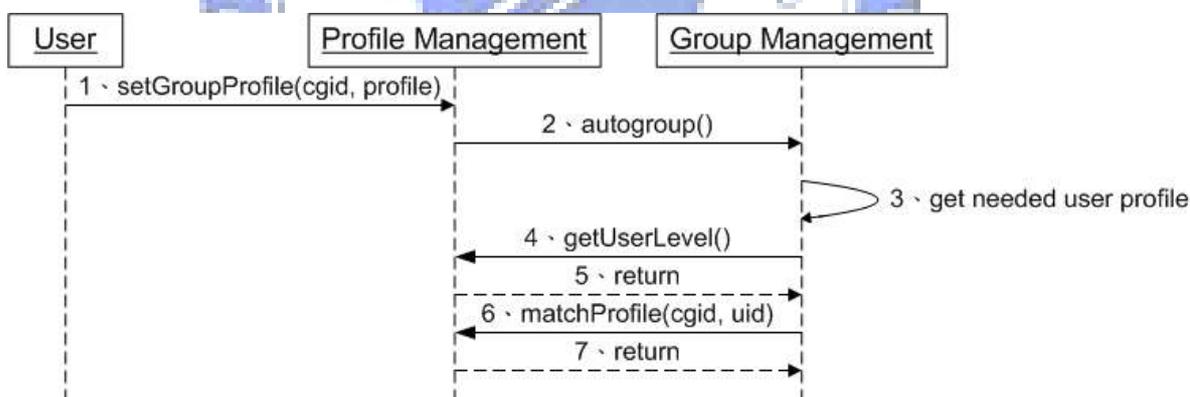


圖 12 Context-aware autogroup 流程

在本系統中，只要 user profile 或 group profile 有所更動，為了隨時保持群組成員符合 group profile 的條件設定，系統便會執行 context-aware autogroup。圖 12 為 group profile 改變，context-aware autogroup 的執行流程，在第三個步驟時，系統會選取需要的 user profile，然後送到 profile management 做比對，決定是新增群組成員或是刪除群組成員。

但因為我們有「user level」的設定，有的使用者可能沒辦法透過 autogroup 的方式加入群組，但同時又希望那些人有機會加入某些通訊群組中，因此我們的系統還需要額外的選擇成員機制。

2、手動邀請、要求加入群組

除了上述的 context-aware autogroup 外，我們另外提供了手動邀請、要求加入群組的設定。在圖 11 中我們知道老師的 user level 大於學生的 user level，假若學生社團的成員除了學生外，也希望社團的指導老師也是成員之一，倘只有 context-aware autogroup 單一機制，便無法達成此一目標。因此，手動邀請、要求加入群組的機制，主要就是根據使用者的意願、個人的意志掌控群組的成員。

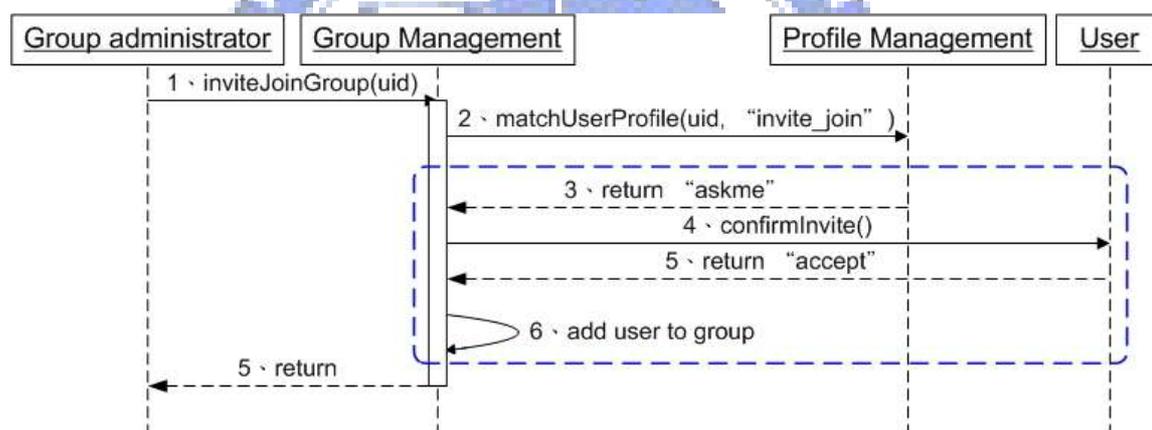


圖 13 手動邀請加入通訊群組的流程

圖 13 是通訊群組管理者手動邀請某位使用者加入群組的流程圖。第二個步驟便是向 profile management 詢問使用者所設定的偏好，在第三個步驟得到回覆，知道該名使用者希望當有人邀請他加入群組時，能夠向使用者本人確認是否加入群組，因此系統會向使用者本人詢問，若是同意加入，系統就將該名使用者加入群組，反之，若不同意則不加入。圖 13 中由虛線框起來的部份，就是會跟著使用者偏好設定而隨之改變的流程，若是在步驟三得到的回答是 accept 或是 reject，那麼系統就不會再次向使用者做確認的動作，而是直接將之同意或是回絕加入群組的邀請。

4.3.3 Groupcasting

這個部份主要負責的功能就是將訊息傳送出去，並且儘可能的支援多種的即時通訊系統，因為每個群組成員所希望使用的即時通訊網路都不太一樣，當使用希望使用 MSN 的通訊網路傳送、接收訊息時，系統便將訊息送往下層的 MSN Controller，採用 MSN 的 protocol 送出訊息，同樣的，需送到 Skype 的通訊網路時，便送往 Skype Controller 走 Skype 的 protocol[13]。

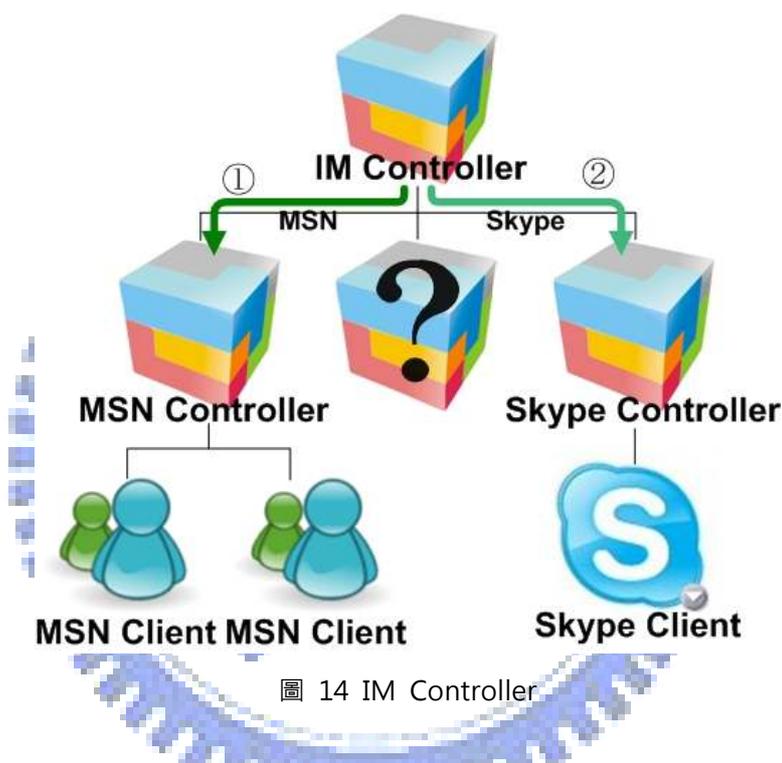


圖 14 IM Controller

如圖 14，這個 component 採用階層式的架構來設計，主要分為三層，最上層為 IM Controller，第二為各即時通訊網路的 controller(如：MSN Controller、Skype Controller 等)，最下面則是即時通訊的用戶端程式(如：MSN Client、Skype Client 等)。最上層的 IM Controller，主要負責接收參數，並將屬於 MSN 的帳號，往 MSN Controller 送，如圖 14 左邊編號 1 號的線，同理，屬於 Skype 的帳號往 Skype Controller 送，圖 14 右邊編號 2 號的線。此外，IM Controller 亦規範群組通訊的基本功能，定出下層單一通訊軟體 controller 的介面，如 `sendMessage()`，下層每個單一通訊軟體的 controller 需根據不同的即時通訊協定來實作這些功能。

針對單一 instant messenger 的 controller，除了基本的功能外，可以針對不同的 instant messenger，做最佳化，或是 implement 不同的功能。以 Skype 來說，因為 Skype 開放出來 API[13]的限制，因此在 Skype controller 下只能 implement 一個 Skype client，不過因為 Skype 有提供手機簡訊 SMS (Short Message Service) 等的其他增值服務，因此，Skype controller 可以針對這些增值服務，開發介面，以供我們的系統使用，提供系統手機簡訊 SMS 的通訊管道。

隨著我們持續擴充這個元件，不斷的支援更多的即時通訊系統，我們可以透過各式的管道傳送訊息。



4.3.4 與其他系統合作

我們將伺服器的動作、功能，利用遠端程序呼叫(Remote procedure call)的方式，提供給我們的系統及外部系統使用。遠端程序呼叫採用 HTTP 的模式運作，以 URL 的方式呼叫，程序運行後的回傳及輸出資料，我們統一採用 XML 的格式，且文字編碼統一為 UTF-8。

➤ 使用者認證、登入階段

這部份，我們主要是延伸 CMS 的功能，並且加以擴充以符合需要。

- **login**

參數：username，password

系統根據使用者之帳號密碼來確認使用者身份，成功確認後，會啟動一個登入階段(session)，往後系統會利用此 session 來確保隱私資料存取的合法性。

- **logout**

參數：無

使用者登出，成功後，將會結束由 login 所建立的 session，以避免非法使用者利用 session 來存取資料。

- **keep_alive**

參數：無

維護目前的登入階段。為了避免逾時而被伺服器判定為使用者已離線，因此在登入後。呼叫 keep_alive 時，都會隱隱進行此動作，讓 session 保持在 active 的狀態。由於 session 是在成功登入後才產生的，因此這邊便順便對 session 做檢查，避免未經認證過的使用者、惡意程式存取，以保護資料及確保資料存取的合法性。

- **gcs_checkIdPwd**

參數：username，password

Gcs_checkIdPwd 主要是用在 group administrator 要求群組成員再一次認證時使用，其功能與 login 非常類似，只差在 login 會產生出 session，而 gcs_checkIdPwd 不會，而是單純的做檢查使用者 id 跟密碼這個動作而已。

➤ **Group management 通訊群組管理部份**

包含通訊群組管理及群組成員管理這兩部份的動作。下面列出系統所提供的 remote procedure call。

- **gcs_addCommGroup**

參數：無

新增一個通訊群組時使用。成功後，**gcs_addCommGroup** 會回傳一個唯一且不重覆的辨識碼：**CGID**，通訊群組被指定 **CGID** 後，便不可更改，因此 **CGID** 為辨識最有效率的法法，無論該群組的基本資料(如：群組名稱、群組描述、…)是如何的改動。

- **gcs_addContactToGroup**

參數：cid

把自己的聯絡人，加入通訊群組。聯絡人是經由同步個人電腦上的通訊錄而來，因此在系統上並沒有 **user profile** 等的偏好設定，所以 **gcs_addContactToGroup** 所負責的工作便是將聯絡人加入群組，不需要其他額外的偏好設定檢查或意願詢問等。

- **gcs_autogroup_group**

參數：uid

若使用者的 **user profile** 有改更動，系統便會呼叫 **gcs_autogroup_group**。系統會把所需要重新比對的通訊群組之 **group profile** 拿來與該名使用者的 **user profile** 做比對的動作，檢查這該使用者是否要加入當中的某一些群組。

- **gcs_autogroup_user**

參數：cgid

一旦 **group profile** 有所改變，系統便會呼叫 **gcs_autogroup_user**。系統會將需重新比對的使用者他們的 **user profile** 拿來與該通訊群組的 **group profile** 做比對，新增或刪除該群組中經由 **context-aware autogroup** 管道而來的使用者。

- **gcs_getGroupList**

參數：無

利用 **login** 所建立的 **session**，得知正確的的使用者，回傳該名使用者所「建立」及「加入」的群組。之所以要分成使用者所建立的通訊群組及使用者所加入的通訊群組兩個部份，主要是在功能上，這兩個部份有所區別。使用者所建立的通訊群組這部份，系統需提供使用者完整的功能，包含群組的偏好

設定、群組成員的增減、群組訊息的發送、...；但對於使用者所加入的通訊群組這部份，使用者無法操作通訊群組管理及群組成員管理這兩部份的功能，因此只提供發送群組訊息的功能，讓使用者可以發送訊息給其他的群組成員，分享訊息。

- **gcs_getGroupMembers**

參數：cgidd

取得該群組的所有群組成員。除了回傳有哪些使用者之外，**gcs_getGroupMembers** 亦會連同使用者的一些基本資料一起回傳。**gcs_getGroupMembers** 會回傳的資料包含有成員的姓名、uid、msn 的帳號、Skype 的帳號。

- **gcs_inviteJoinGroup**

參數：uid, cgid

邀請使用者(uid)加入通訊群組(cgid)。

- **gcs_requestJoinGroup**

參數：cgid

請求加入群組。

- **gcs_removeFromGroup**

參數：uid, cgid

把使用者從群組中移除。若呼叫 **gcs_removeFromGroup** 的人是群組的管理者，那麼，他可以移除所有的群組成員，除了他自己。若呼叫 **gcs_removeFromGroup** 的人只是群組裡的成員，並非管理者，那麼就只能把自己移除讓群組，意即自己退出群組。

- **gcs_deleteGroup**

參數：cgid

刪除通訊群組。在真正刪除通訊群組前，系統會先確認該使用者是否可以刪除 CGID 這個通訊群組，因為只有群組的管理者才有資格刪除群組。在確認可以刪除後，**gcs_deleteGroup** 會刪除所有與該通訊群組有關的資料，包括 group profile 和在資料庫系統中的資料。

➤ **Profile management 部份**

這個部份的動作，主要是維護 user profile 和 group profile。

- **gcs_getGroupProfile**

參數：cgid

取得通訊群組的 profile。

- **gcs_getUserProfile**

參數：uid

取得使用者的 profile。回傳先前設定好、已儲在系統的 user profile。

- **gcs_getUserMetadata**

參數：無

取得 user profile metadata。gcs_getUserMetadata 主要用於在設定 group profile 和 user profile 之前，要先向系統取得 user profile metadata，以便 parser 解析，產生使用者介面以供使用者設定。

- **gcs_setGroupProfile**

參數：group profile

新增、修改 group profile。輸入的參數實際上會根據 user profile metadata 的改變而有所不同。User profile metadata 所影響的部份，主要是在 Context-aware autogroup 上，選擇成員的條件。系統在讀取 metadata 後，會 parse 出可供使用者設定的屬性，使用者再根據這些屬性設定其值，再後，系統再依照使用者設定好的條件，替該群組維護成員(新增或刪除)。

- **gcs_setUserProfile**

參數：user profile

新增、修改 user profile。在更動 user profile 後，系統會呼叫 gcs_autogroup_group，讓系統自動維護通訊群組的成員。

➤ **Groupcast 部份**

- **gcs_groupcast**

參數：cgid, message

將 message 傳送給該通訊群組中的每位成員。在傳送時，系統會依據使用者的設定，發送至使用者所設定第一優先的通訊管道。

4.3.5 資料庫結構

在伺服器端，我們以一資料庫做為最底層的資料儲存，基於國際化考量，資料庫中所有的資料、文字皆以 Unicode UTF-8 格式儲存。

由於我們的系統，有部份是延伸自 CMS 系統，因此會使用到 CMS 資料庫中的資料。表 1 為在本系統中會用到的 CMS 資料表。表 2 為我們系統中會用到且較重要的資料欄位。

表 1 CMS 資料庫

Table Name	Description
<u>Contacts</u>	聯絡人資訊

表 2 為 CMS 系統中，關於聯絡人的資料。我們的系統在 groupcast 時，若發送訊息的對象為聯絡人時，聯絡人的即時通訊帳號、線上狀態等資料，皆是來自這個表格。

表 2 聯絡人資訊

Contacts		
Field	Data type	Description
CID	Char(34)	聯絡人編號 (Contact ID, CID)
UID	Char(33)	使用者編號
GIVEN_NAME	Char(32)	聯絡人名
SURNAME	Char(32)	聯絡人姓
MSN	Char(255)	MSN Messenger 帳號
SKYPE	Char(255)	Skype 帳號

MSN_STATUS	Char(16)	聯絡人的 MSN Messenger 上線狀態
SKYPE_STATUS	Char(16)	聯絡人的 Skype 上線狀態

除了上述引用到 CMS 系統現存的資料表格外，我們亦根據需求，規劃出下列的表格，來記錄儲存我們系統的資料。

表 3 為記錄通訊群組的管理者(administrator)。

表 3 Communication Group Administrator

CGS_CGLEADER		
Field	Data type	Description
<u>CGROUP</u>	Char(34)	同 CGID，每個通訊群組的識別碼
<u>CGLEADER</u>	Char(33)	該群組的 administrator

表 4 記錄著群組的成員，以及成員是透過何種管道加入該群組的。其中 MODE 這個欄位，就是在記錄使用者是經由何種管道加入群組的，該欄位有以下幾種值：autogroup、invite、request、contact。

Autogroup：表示這位使用者是經由 context-aware autogroup 機制所加入群組的，這位使用者是否在群組內，完全由 context-aware autogroup 機制所控制。

Invite：表示這位使用者，是由群組管理者所邀請加入。Context-aware autogroup 機制不會影響到經由此管道加入群組的使用者。

Request : 表示該名使用者,是自己要求加入群組的。Context-aware autogroup 機制不會影響到經由此管道加入群組的使用者。

Contact : 表示這位群組成員是使用者的聯絡人。聯絡人是否在群組內,完全由使用者所控制。

表 4 通訊群組成員

CGS_CGMEMBERS		
Field	Data type	Description
<u>CGROUP</u>	Char(34)	通訊群組
<u>MEMBER</u>	Char(34)	群組成員
MODE	Char(10)	記錄是何種方式加入通訊群組

表 5 記錄所有需要 context-aware autogroup 的通訊群組。當某位使用者的 user profile 有所更動時,系統會將 user profile 與在表格中的通訊群組其 group profile 做比對,以決定使用者是否加入這些通訊群組。

表 5 需要 autogroup 的通訊群組

CGS_CGAUTOGROUP		
Field	Data type	Description
<u>CGROUP</u>	Char(34)	需要做 autogroup 的通訊群組

根據 user profile 和 group profile 中的設定,有很多系統的動作是需經過使用

者確認或決定的，因此系統需要發訊息向使用者詢問，但使用者並非隨時隨地都會在線上，因此表 6 便是一個表格，記錄著系統準備向使用者發送的訊息。表 7 為通訊群組管理者邀請某位使用者加入群組的例子。表 8 為使用者要求加入通訊群組的例子。

表 6 等待回覆的要求

CGS_WAIT_RESPONSE		
Field	Data type	Description
SEQ	Bigint(20)	自動增加的序號
TYPE	Char(10)	邀請或者是要求
UID	Char(34)	發起邀請/要求的使用者
TARGETUID	Char(34)	想要邀請加入群組的使用者 想要加入通訊群組的 administrator
CGROUP	Char(34)	邀請加入的通訊群組 所要加入的通訊群組
TIMPSTAMP	timestamp	時間戳記

表 7 中，TYPE 欄位的值為 invite，表示這筆資料是「邀請加入群組」，user1 發送邀請，邀請 user2 加入 group1 這個群組。

表 7 等待回覆的要求-邀請加入群組 (例子)

SEQ	TYPE	UID	TARGETUID	CGROUP	TIMESTAMP
1	invite	User1	User2	Group1	...

表 8 中,TYPE 欄位為 request,表示這筆資料是「要求加入群組」,user1 向 user2 詢問是否可以加入 group1。

表 8 等待回覆的要求-要求加入群組 (例子)

SEQ	TYPE	UID	TARGETUID	CGROUP	TIMESTAMP
2	request	User1	User2	Group1	...

表 9 記錄著系統運作過程中的訊息、系統運作後的結果。其中包括有：邀請加入群組的結果、要求加入群組的結果等。

表 9 狀態訊息

CGS_STATUSMESSAGE		
Field	Data type	Description
SEQ	Char(34)	自動增加的序號
UID	Char(33)	接收訊息的使用者
MESSAGE	text	訊息內容
TIMPSTAMP	Timestamp	時間戳記

表 10 記錄著每個通訊群組的成員數量。在資料庫系統中，表 10 是一個 view，系統並不直接操作這個表格的內容，而是由資料庫系統自動維護，每當通訊群組成員有所變動，即表 4 中的資料有所增減時，資料庫系統會同步更新這個表格。

表 10 通訊群組成員數量

CGS_MEMBERCOUNT		
Field	Data type	Description
<u>CGROUP</u>	Char(34)	通訊群組
COUNT	Bigint(21)	通訊群組的成員數量

4.3.6 AJAX 和 Remote procedure call 的使用

在傳統的網頁瀏覽上，瀏覽器透過 POST 或 GET 傳送參數給網頁伺服器，再接收由伺服器端傳回來的新網頁，此時瀏覽器需要重新載入整個頁面，好讓接收到的新網頁呈現在瀏覽器上以供使用者瀏覽閱讀。亦由於重新載入的關係，即使使用 JavaScript 增加網頁的互動性，有許多在操作過程中會產生的變數都無法保存下來，因此不太適合做太複雜的系統。

AJAX 全名為 Asynchronous JavaScript and XML [14][15][16]。AJAX 是透過 Javascript[17]中的類別 XMLHttpRequest 在背景悄悄地與伺服器通訊，亦可以在有需要時才向伺服器索取資料，並且將回傳的資料用 XML DOM(Document Object Model)解析，方便我們在 XML 的回傳中快速取得想要的資料，動態地修改局部的網頁內容。如此一來，不必重新載入網頁即可更新網頁內容，呈現最新資訊，伺服器也不必每次都要傳輸整個頁面給瀏覽器，減輕伺服器的負擔，也減輕資料的傳輸量，而使用者可以得到更佳的使用體驗。

在本系統的網頁介面，便是採用 AJAX 技術來實作。由於我們在設計 API 時有將 HTTP 的功能考慮進去，因此這邊可以利用 XMLHttpRequest 來呼叫伺服器端的 produces，並將回傳的 XML 格式資料以 DOM 解析，動態更新網頁內容。再加上不會每次取得資料就必須重新載入網頁，因此 javascript 可以儲存更多與使用者相關的資訊，對於使用者介面的體驗，有加分的作用。



第五章 實作成果演示

5.1 Web 使用者介面

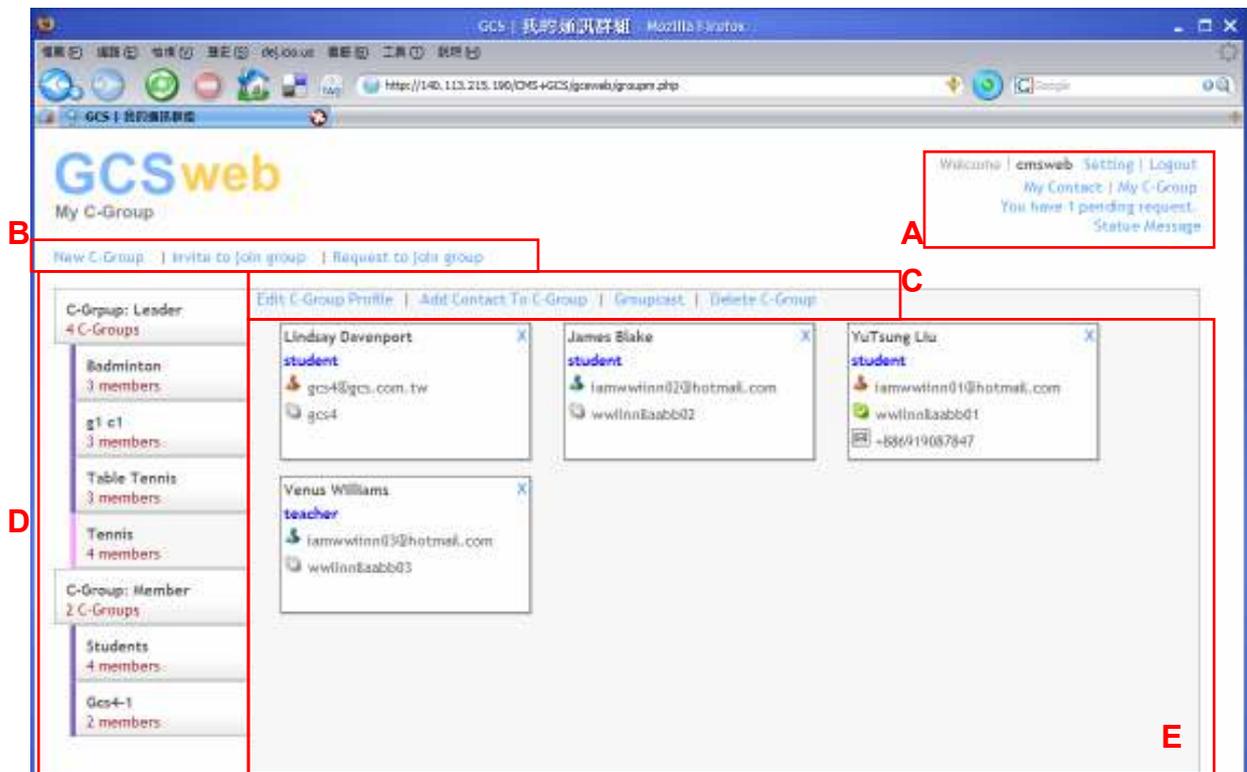


圖 15 網頁使用者介面

圖 15 為系統實作出來的 web 使用者介面，主要可以畫分為五個大區塊，在下面分別介紹。

A、使用者資訊區：使用者可以在這邊設定個人偏好、顯示是否有「尚待回覆的要求」、以及系統運作後的狀態訊息。

B、通訊群組工具列：由左而右，依次為「新增通訊群組」、「邀請加入通訊群組」、「請求加入通訊群組」。

C、通訊群組工具列：目前開啟的通訊群組所能進行的操作。若目前開啟的通訊群組的管理人為使用者本人，那麼可以進行「修改 group profile」、「將聯絡人加入

群組」、「發送群組訊息」、「刪除群組」，如圖 16。若目前開啟的通訊群組屬於「我所加入的通訊群組」，則使用者儘能發送群組訊息，如圖 17。



圖 16 通訊群組工具列-我所建立的通訊群組



圖 17 通訊群組工具列-我所加入的通訊群組

D、通訊群組資訊區。這邊細分為「使用者所建立的通訊群組」和「使用者所加入的通訊群組」。

E、群組成員資訊區。所有的群組成員資訊以資料卡的方式呈現。如圖 18 所示，除了姓名等，還可一目瞭然的清楚知道該名成員即時通訊的線上狀態。

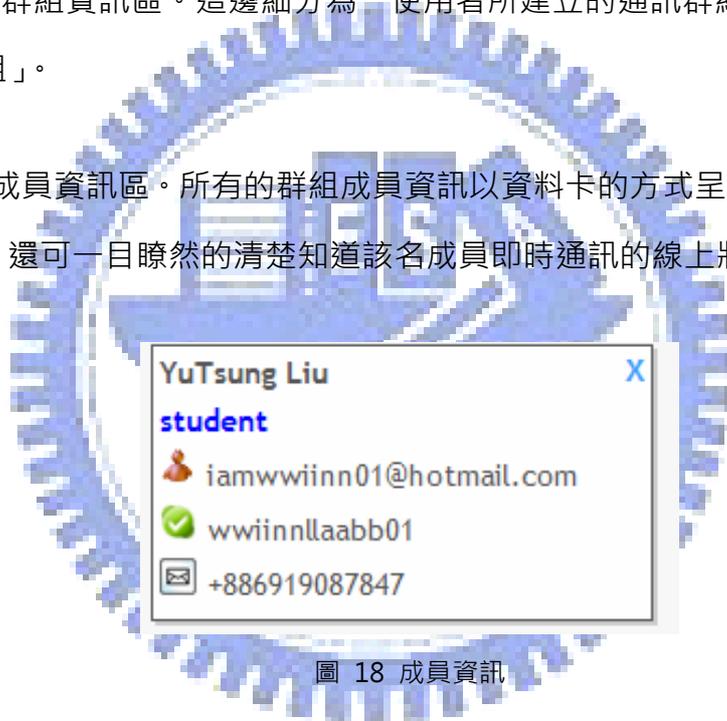


圖 18 成員資訊

5.2 偏好設定

圖 19 為系統中偏好設定的介面。(a)為使用者偏好設定的介面，(b)為通訊群組的偏好設定。設定完成後，將這些資訊分別儲存在 user profile 和 group profile 中，系統日後的行為，便是根據這些設定而來。

The image shows two side-by-side preference setting windows. The left window is titled "Preference" and contains three sections: "Information Setting" with fields for Name, Address, Relation, Class, and Club; "Communication Setting" with fields for MSN, Skype, and SMS, each with a priority dropdown; and "Group Setting" with radio buttons for AutoGroup, Receive Invite, and Visible, each with a description. The right window is titled "C-Group" and contains two sections: "Information" with fields for Group Leader, Group Name, and Description; and "C-Group Setting" with radio buttons for Context-aware Autogroup, Condition Operator, Auth Method, Request to Join group, and Contacts Communication, along with dropdown menus for Club and Relation. Both windows have OK and Cancel buttons at the bottom.

(a) 使用者偏好設定

(b) 通訊群組偏好設定

圖 19 偏好設定

5.3 群組訊息

可以透過系統提供的介面，輸入群組訊息。如圖 20。



圖 20 輸入群組訊息

透過本系統，可以將訊息傳送給使用不同 IM networks 的使用者。如圖 21。

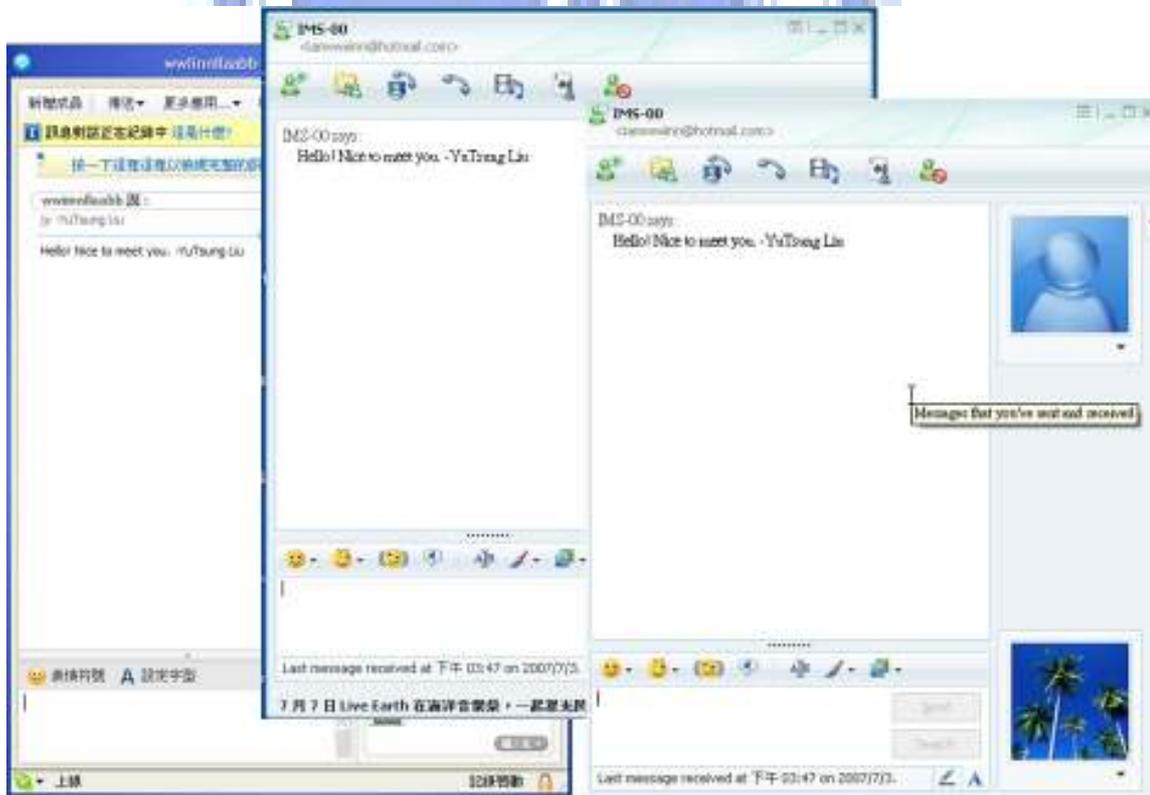


圖 21 接收訊息

5.4 群組管理資訊

有些動作，需要使用者親自確認，像是有人要求加入群組時，需要群組管理者的同意等...。因此系統要個介面給這種情況使用，如圖 22，Venus Williams 請求加入 Table Tennis 這個通訊群組，而該群組的管理者可以透過這個介面，接受或拒絕 Venus Williams 的請求。



圖 22 等待回覆的要求



第六章 結論與未來發展

6.1 結論

在本篇論文中，我們勾畫出一個以群組為基礎並且整合即時通訊軟體的通訊平台，並搭以網頁的使用者介面，讓使用者以最方便的方式瀏覽管理通訊群組。根據我們的需求，設計具高彈性、高擴充性的系統架構。我們的系統具有下列的特點：

我們的系統使用網頁做為瀏覽的介面，無論是管理或是發送訊息，只要有瀏覽器的地方，就可以使用我們的系統。

允許每個使用者建立屬於他自己的通訊群組，群組內的成員完全由自己掌控，可以發送訊息給所有的群組成員，亦可以接收來自群組成員所發送的訊息。

允許每位使用者、每個通訊群組設定偏好。每個人都有自己的習慣，同樣的每個群組的行為多多少少也會不同，使用者、群組根據不同的需求設定其偏好，讓系統以符合使用者、群組的期望來運行。

在通訊群組內的成員，都可以發送群組訊息。透過系統的轉送，可以將訊息傳送給使用不同即時通訊系統的使用者手上，省去了需要安裝一堆即時通訊軟體的麻煩。

我們設計一個具高度擴充性的架構，可以在將來持續擴充支援更多種類的即時通訊系統，讓我們的系統，在做群組通訊時，可以涵蓋更多使用者、更多的即時通訊網路，有更多的終端使用者，可以不用改變原本使用的通訊網路，即可接收訊息。

我們系統開發出一組通用的 remote procedure calls，提供給第三方系統使用，利用本系統的功能，合作開發服務。由於我們的 remote procedure calls 可以使用 HTTP 做為溝通的媒介，在彼此合作，彼此溝通上，大大減輕開發成本。

6.2 未來發展

使用者位置感知：可以透過其他系統的輔助，得知目前使用者身在何處，隨著使用者身處不同的地方，動態地修改即時通訊系統的優先權，讓系統使用合宜時地的即時通訊網路來發送訊息。舉例來說，感測到使用者進入了辦公室，便將 MSN Messenger 的優先權設高，讓系統以 MSN Messenger 傳送訊息；若感測到使用離開了辦公室，則將 SMS 設為高優先，讓系統以手機簡訊的方式傳送訊息。

線上維護即時通訊系統聯絡人的線上狀態：針對聯絡人的線上狀態感知，目前在每個客戶端安裝一隻小程式，負責讀取並上傳至伺服器更新。在未來，希望可以由伺服器端直接取得這部份資料，省去需另外安裝程式的麻煩。

整合更多的即時通訊網路或其他的通訊管道：目前系統可以支援 MSN Messenger 和 Skype 的即時通訊，但市面上仍有許多即時通訊系統，如 Google Talk 等，都是未來可以考慮新增支援的，隨著擴充各種通訊管道，系統可以將訊息發送至更多的使用者手上。

系統應用：在我們的系統裡包含了一組通用的遠端程序呼叫介面，提供給第三方系統來存取，希望本系統與其他系統相互合作，利用本系統所提供的功能，做為一個通訊平台，進一步開發出不同類型，各式各樣的服務。

參考文獻

- [1] C. Zhang, S. M. Sadjadi, W. Sun, R. Rangaswami, Y. Deng, "User-Centric Communication Middleware," Technical Report FIU-SCIS-2005-11-01. November 2005
- [2] Y. Deng, S. M. Sadjadi, P. Clarke, C. Zhang, V. Hristidis, R. Rangaswami, and N. Prabakar, "A Unified Architectural Model for On-Demand User-Centric Communications," Technical Report FIU-SCIS-2005-09, School of Computing and Information Sciences, Florida International University, 2004.
- [3] V. Poladian, J. P. Sousa, D. Garlan, and M. Shaw, "Dynamic configuration of resource-aware services," in Proceedings of the 26th International Conference on Software Engineering, (Edinburgh, Scotland), May 2004.
- [4] S. L. Tsao, J. C. Chou and Y. C. Hsu, "Inter-working and integration of Messaging Services in a Heterogeneous Wireless Environment," In Proc. of IEEE. 1st International Conference Pervasive Computing and Communications. (PerCom), Mar. 2003.
- [5] Y. F. Chen, H. Huang, R. Jana, T. Jim, M. Hiltunen, R. Muthumanickam, S. John, S. Jora and B. Wei, "iMobile EE – An Enterprise Mobile Service Platform," ACM Journal on Wireless Networks, Volume 9, Issue 4, pp. 283-297, July 2003.
- [6] J. M. Wams and M. van Steen, "Pervasive Messaging," In Proc. Of IEEE 1st International Conference Pervasive Computing and Communications (PerCom), Mar. 2003.

- [7] C. K. Yeo, S. C. Hui, I. Y. Soon, and G. Manik, "Unified Messaging: A System for the Internet," International Journal on Computers, Internet, and Management, September 2002.
- [8] D. Mierla, "SIMPLE-XMPP Interworking,". Internet Draft. 22 October 2003. draft-mierla-simple-xmpp-interworking-01.
- [9] Y. Chen, H. Huang, R. Jana, S. John, S. Jora, A. Reibman and B. Wei, "Personalized multimedia services using a mobile service platform," in Proceedings of the IEEE Wireless Communications Networking Conference, Florida (March 17–21, 2002).
- [10] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging". RFC 3428 (Standards Track). December 2002.
- [11] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," RFC 3428, December 2002.
- [12] M. Nilsson, J. Hjelm and H. Ohto, "Composite capabilities/preference profiles: Requirements and architecture," W3C Working Group <http://www.w3.org/Mobile/CCPP/>
- [13] Skype Developer Zone "Skype4Com Library" <https://developer.skype.com/Docs/Skype4COMLib/>
- [14] W3C "AJAX Tutorial" <http://www.w3schools.com/ajax/default.asp>
- [15] Mozilla Developer Center "AJAX: Getting Started" http://develper.mozilla.org/en/docs/AJAX:Getting_Started
- [16] J. Garrett, "Ajax: A new approach to web applications,"

<http://www.adaptivepath.com/publications/essays/archives/000385.php>,

Feb 2005

[17] W3C "JavaScript Tutorial"

<http://www.w3schools.com/js/default.asp>

[18] H. Alvestrand, "Instant Messaging and Presence on the Internet," ISOC MEMBER BRIEFING #9, isoc.org

[19] Extensible Messaging and Presence Protocol (XMPP) IETF Working Group.
<http://www.ietf.org/html.charters/xmpp-charter.html>

[20] H. Rao, Y. Chen, D. Chang, M. Chen, "iMobile: A proxybased platform for mobile services," *Proc. of the First ACM Workshop on Wireless Mobile Internet (WMI 2001)*, Rome, July 2001.

[21] G. W. Bond, E. Cheung, K. H. Purdy, P. Zave, and J. C. Ramming, "An open architecture for next-generation telecommunication services," *ACM Transactions on Internet Technology* IV(1):83-123, February 2004.

[22] A. K. Dey and G. D. Abowd, "The context toolkit: Aiding the development of context-aware applications," in *Proceedings of the 22nd International Conference on Software Engineering (ICSE): Workshop on Software Engineering for Wearable and Pervasive Computing*, (Limerick, Ireland), June 2000.

[23] H. T. Chu, W. S. Chen, Y. H. Huang and J. Y. Chen, "A novel design of instant messaging service extended from short message service with XMPP," *3G Mobile Communication Technologies, 2004. 3G 2004. Fifth IEE International Conference*, page 504-508,

[24] M. Book, V. Gruhn, and G. Mücke, "An Instant Message-Driven User

Interface Framework for Thin Client Applications," Automated Software Engineering, 2006. ASE '06. 21st IEEE/ACM International Conference, page 257-260.

[25] A. J. H. Peddemors, M. M. Lankhorst, and J. de Heer, "Presence, location, and instant messaging in a context-aware application framework," In MDM' 03 Proceedings of the 4th International Conference on Mobile Data Management, pages 325–330. Springer-Verlag, 2003.

[26] G. Lawton, "Instant Message Puts on a Business Suit," IEEE Computer, March 2003.

[27] J. Peterson, "Common Profile for Instant Messaging (CPIM)," RFC 3860, August 2004.

[28] J. Peterson, "Common Profile for Presence (CPP)," RFC 3859, August 2004.

