

Web-Cow: 全球資訊網上的電腦輔助合作工作系統

Web-Cow : A CSCW system on WWW

黃耀賢 林志勝 王宗銘

Yao-Hsien Huang, Chih-Sheng Leen and Chung-Ming Wang

圖學多媒體暨虛擬實境實驗室

國立中興大學資訊科學研究所

Graphics, Multimedia and Virtual Reality Lab.

Institute of Computer Science, National Chung Hsing University

Taichung, Taiwan 402, R.O.C.

摘要

電腦輔助合作工作全名為 Computer Supported Cooperative Work (CSCW)，其意義為藉由電腦的輔助來達成團隊合作的工作，如共同編輯系統(Co-Editing System)及共同決策系統(Co-Decision System)等。此篇論文的目的在於提出一套架構在全球資訊網路上，採行雙重 Server 的 Client/Server 架構，將現有的全球資訊網的低互動的特性，提出一種改良的方法，使得我們可以在不更改現行全球資訊網架構的前提下，及原有的全球資訊網架構設計一套多群多人的共同圖文編輯系統 (World Wide Web Co-Operative Work System，縮寫為 WebCow System)。

關鍵字: CSCW, WWW, Multimedia

Abstract

We want to develop a system that is combined with CSCW and WWW. The system is named by World-Wide Web Co-operative Work System (WebCow system). Features of WebCow system is not to modified the web server and system function will aggrandized in the future.

Keywords: CSCW, WWW, Multimedia

一、簡介

由於電腦軟硬體的急速發展，個人電腦的應用已從傳統的單一主機運作，發展到網路資源分享的多人多工環境，而網路的環境也從純文字模式如 Gopher、Telnet 等發展成具有多媒體及超文件 (Hypertext) 的全球資訊網 (World Wide Web)。

WWW (World-Wide Web) 由 CERN (the European Laboratory for Particle Physics) 歐洲量子物理實驗室在一九八九年提出，原本是為了內部傳輸查詢多種形態資料之用，如今已搖身一變成爲 Internet 世界的新寵，其目的在於建立一個分散式多媒體的網路資訊系統。全球資訊網的基本架構是採開放式主從架構 (client-server) 的方法，其中：

- 伺服器端架構規定了伺服器的傳輸協定、伺服器本身的基本開放架構及資料的傳輸格式。

- 客戶端架構規劃出資訊接收的格式以建構適當的資訊資料接收工具如瀏覽器 (browser) 等。

目前的全球資訊網，提供的資訊有：文字 (Text)、聲音 (Sound)、圖形 (Graphic) 及影像 (Video)，而運作的方式除了一般從伺服器端 download 資料到客戶端的傳統單向展示外，也可透過通用閘道界面 CGI (Common Gateway Interface) 來達成伺服器端與客戶端的互動。

綜觀當今世界上許多的 CSCW 系統像是視訊會議系統及圖文共同編輯系統等，均是架構在區域網路 (LAN, Local Area Network) 上，這樣在實用性上會大打折扣，因為唯有架構在廣域網路 (WAN, Wide Area Network) 上的 CSCW 系統才会有其真正的實用性。現今世界上最大眾化的廣域網路非全球資訊網莫屬，可是由於先天的架構加上一些如檔案上傳 (upload) 等牽涉到資訊安全 (Information Security) 等的問題，使得現在的全球資訊網還是停留在資訊提供者 (Information Supporter)，只能透過 CGI (Common Gateway Interface) 及 Java 等做有限的互動，此外，全球資訊網的伺服器，會因為發展公司基於一些考量而做其特殊的修改，例如 Micro Soft 的伺服器 IIS (Internet Information Server) 的 ISAPI (Internet Server Application Program Interface) 等，因此若改寫全球資訊網的伺服器可能會導致在未來只要有新的伺服器標準出來就要改寫的問題，因此我們希望能發展出一套架構在現今全球資訊網上，不更改全球資訊網的伺服器端，提供多群 (Group) 多人 (Multi-User) 共同線上編輯的 CSCW 應用系統 (World Wide Web Co-Operative Work System，縮寫為 WebCow System)，期望透過 WebCow 系統來實現 CSCW 在全球資訊網上的應用。

二、背景探討

所謂的電腦輔助合作工作 CSCW，主要目的是藉由電腦的輔助來達成團隊合作的工作例如新聞討論群 (News Group)、共同編輯系統 (Co-Editor System) 及電子視訊會議系統 (Video Conferencing System)

等。

在電腦輔助合作工作的第一個特徵「電腦輔助」方面，是比較偏重於關於電腦技術性方面的研究，其目的在於電腦如何輔助團隊合作工作，電腦與人的互動研究多屬於人機介面 (Human Computer Interface) 方面，主要研究內容為如何能滿足團隊中多數使用者需求之友善使用者介面 (Friendly User Interface)，而其另一個特徵「團隊合作」則主要研究人與人之間的互動作用，依互動型態的不同有同步 (Synchronous) 與非同步 (Asynchronous) 兩種模式，由地域性差異而有同域式 (Co-located) 與分散式 (Distributed) 的不同。

根據互動型態與地域性差異的不同，可將電腦輔助合作工作分成四種模式：同步 (Synchronous)、分散式同步 (Distributed Synchronous)、非同步 (Asynchronous) 與分散式非同步 (Distributed Asynchronous) [5]，這四種模式涵蓋了電腦輔助合作工作的所有領域，因此針對這四種模式來了解其需求，進一步解決其癥結問題便是電腦輔助合作工作的研究重點。

三、WebCow 系統設計目標

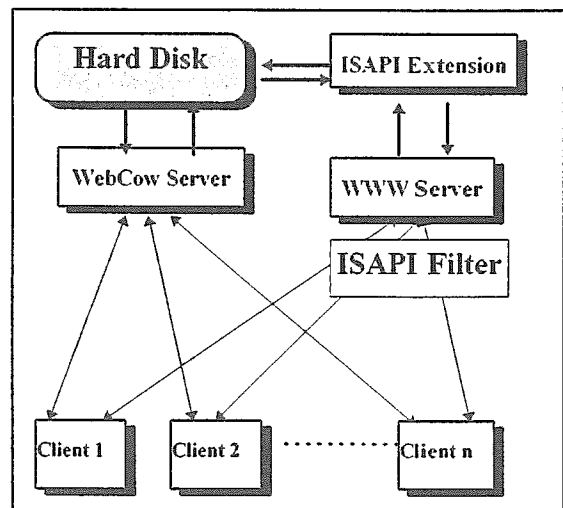
WebCow 系統主要的設計目標在於不修改現行全球資訊網的架構，且不改寫全球資訊網伺服器端的前提下，實做一套架構在全球資訊網上的多群多人共同編輯系統。WebCow 系統，WebCow 系統除了要達到電腦輔助合作工作的五種功能需求外，還具有以下特點：

1. 多群多人共同線上編輯
要能夠讓多個群，每個群多位使用者同時上線編輯，且要能辨別使用者的來源。
2. 資料的完整性與一致性
資料的內容必須保證其完整性與一致性，且要避免在傳輸過程中遺漏部分內容，以達到 WISIWYS。
3. 使用者的互動
使用者之間要有溝通對話的管道，且對於互動事件必須有協調能力。
4. 圖文整合
WebCow 系統除了可共同編輯文件外，還提供可繪圖的電子白板功能。
5. 提供硬碟空間 (Virtual Workspace) 讓使用者與群體可以暫存檔案
在伺服器端提供部分硬碟空間來暫存使用者資料。
6. 具有資料壓縮與資訊安全的考量
利用現有的壓縮工具來達成減少資料量與達到資訊安全的目的。
7. 利用 Netscape Plug-Ins 擴充現有的 WWW 架構
對於多群多人共同線上編輯這個目標，WebCow 系統必須具備管理使用者的能力，對於資料完整一致性與安全性，系統需提供資料壓縮與管理來辨識與管理使用者的資料，對於提供硬碟空間方面系統需對於使用者有硬碟容量 (Quota) 的限制，對於使用者互

相討論的的管道，需具備有決策能力，此外提供電子白板以提供更多元的文件內容。為了實現這些目標，WebCow 系統採用 Client/Server 中央集權式架構，且在全球資訊網的伺服器外另外多加一個 WebCow 伺服器以達成使用者的管理與互動。

四、系統架構

WebCow 系統是採用兩個 Server 的 Client/Server 架構 (如圖一)，WebCow Server 端的主要工作為處理來自各個群的 Client 端的使用者訊息、接受 Client 端文件與白板資料的上傳並將之做適當的處理放在 Server 端硬碟 (Virtual Workspace) 中適當的地方、判斷資料的來源的合法性與負責決策的進行與結果等四項；而 WWW Server 端的工作與一般的全球資訊網伺服器端的工作一樣，為資訊上傳 (Down-load) 的提供者；ISAPI Extension 部分主要的工作為處理客戶端的登入、決定下傳哪些資料給哪些 Client 端與判別現在系統網路的負載以決定是否允許新的 Client 端進入使用此系統；ISAPI Filter 部分主要的工作為攔截非法想取用系統資源的非法使用者；而 Client 端則是實際與使用者互動的應用程式介面，主要為文件段落的增刪與編修，以及電子白板繪圖元件的增刪，還有傳達與接收溝通訊息，此外還要處理發出與接收系統使用的決策與結果。因為 Client 端的文件編輯子系統資料包含了段落與元件，因此必須定義段落與元件的資料結構，而電子白板的資料則包括有畫筆 (Pen)、直線 (Line)、圓 (Circle)、橢圓 (Ellipse) 及矩形 (Rectangle)，元件屬性則標明了元件型態的屬性值，如元件的顏色等。



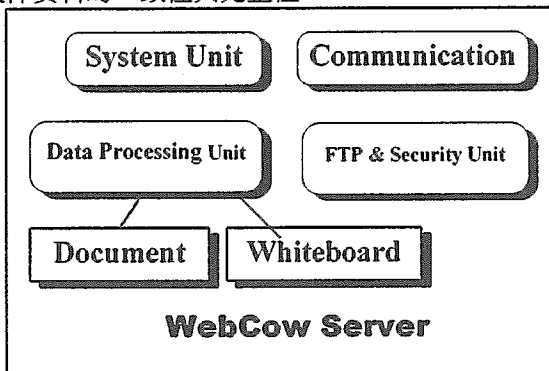
圖一 WebCow System 架構圖

WebCow Server 端有四個單元模組 (如圖二)：系統單元 (System Unit)、溝通單元 (Communication Unit)、檔案傳送與資訊安全單元 (FTP & Security Unit) 與資料處理單元 (Data Processing Unit)。其中系統單元處理一些系統訊息例如現在有哪些

群，每個群有哪些人在使用，處理文件與白板的投票和鎖定訊息，判斷使用者是否有超過其硬碟使用空間大小的限制，對於該更新資料的客戶端發出 Netscape Plug-In reload 的訊息，以及處理客戶端離開系統時的一些工作。

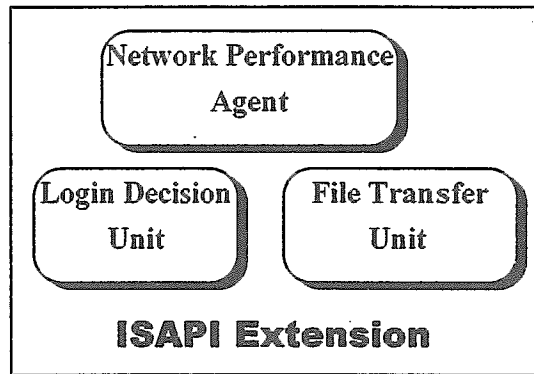
檔案傳送與資訊安全單元處理客戶端上傳的文件與白板資料，判斷其是否為合法的客戶端所上傳的資料且其資料大小有無超過限制，若均沒錯就將此資料交給資料處理單元。

資料處理單元接收檔案傳送與資訊安全單元所傳來的資料，判斷其是文件資料或是白板資料將其交由文件或是白板子單元處理。此兩個子單元會將此資料處理然後分成兩檔案存在硬碟 (Virtual Workspace) 中，其中一個檔案是只含最新修改的部分，另一個檔案則是完整的整份資料，然後呼叫系統單元傳送更新資料的訊息給其同一群的其他客戶端使其透過 Netscape Plug-Ins 的 reload 動作來更新白板及文件的資料內容，在伺服器硬碟中的完整資料是為了避免若客戶端發生網路資料遺失而產生資料不一致的情形時，仍能透過下載這份完整的資料來維持白板與文件資料的一致性與完整性。



圖二 WebCow Server 單元模組

在 ISAPI Extension 部份有三個單元模組 (如圖三)：網路效能管理者 (Network Performance Agent)、登入處理單元 (Login Decision Unit)、檔案傳輸單元 (File Transfer Unit)。網路效能管理者負責計算最近單元時間內整個伺服器端的網路效能，以判別決定是否要給新的使用者登入上線使用。登入處理單元負責處理使用者的登入，密碼的判定等並結合網路效能管理者來判斷是否要給要求登入的使用者登入使用。檔案傳輸單元負責將最新的文件或白板資料交給 WWW Server 以傳送給適當的客戶端。



圖三 ISAPI Extension 單元模組

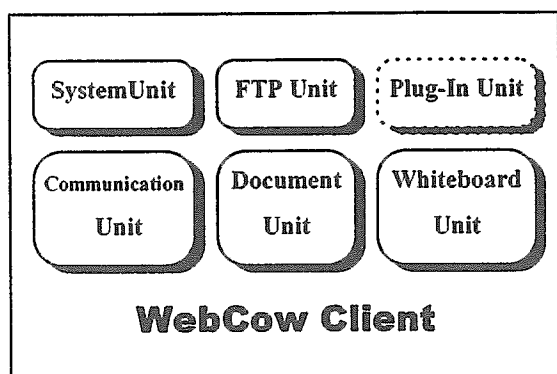
WebCow Client 端有六個單元模組 (如圖四)：系統單元 (System Unit)、檔案傳輸單元 (FTP Unit)、Plug-In 單元 (Plug-In Unit)、溝通單元 (Communication Unit)、文件單元 (Document Unit) 與電子白板單元 (Whiteboard Unit)。系統單元負責處理一些系統的訊息像是現在系統有哪些群體哪些人在使用，接收 WebCow Server 發出更新白板或文件的訊息並將之交給 Plug-In 單元處理，接收與發送文件與白板的鎖定與解除鎖定訊息並交給文件單元或白板單元以鎖定或解除鎖定文件或白板，及在結束系統使用時向 WebCow Server 提出離開的要求。

檔案傳輸單元負責將客戶端所編輯後的文件或白板資料壓縮，然後用 TCP 的方法傳送給 WebCow Server 的檔案傳送與資訊安全單元以傳送給同群其他客戶端做更新白板或文件資料之用。

Plug-In 單元的工作有兩個，一是在一開始時呼叫整個客戶端程式起來執行工作，在程式設計實做時，因為 Netscape 的瀏覽器會將 Plug-In 程式當成一個子行程 (Process) 來處理，為了避免因為客戶端程式過於龐大而導致系統不穩，因此將 Plug-In 單元獨立出來寫成一個 .dll 檔放在 Netscape 瀏覽器的 Plugins 子目錄中以供 Netscape 的瀏覽器呼叫，這也就是為什麼圖四中的 Plug-In 單元其外框是虛線的原因，而 Plug-In 單元的另一項工作是當接到系統單元所發出更新白板或文件資料的訊息時，向 WWW Server 要求最新的白板或文件的更新資料。

溝通單元的主要工作是負責接收由 WebCow Server 所送來的溝通訊息並將之顯示於溝通視窗中，並將客戶端的溝通訊息傳送給 WebCow Server 以傳送給同群的其他客戶端。

文件單元與電子白板單元則是負責文件與白板的段落與元件的增刪、修改與顯示採用動態記憶體配置 (Dynamic Memory Allocation) 與鏈結 (Linked Lists) 來處理新增或修改後的段落與元件，資料的內容以 WWW Server 所傳送出來的資料為準，為了維持資料的完整性與一致性，在同一時間同一群中只能有一個客戶端編輯文件資料，一個客戶端編輯白板，使用者必須先向 WebCow Server 的系統單元要求鎖定 (Lock) 文件或白板來進行編修，編修完後再向 WebCow Server 提出解除鎖定。



圖四 WebCow Client 單元模組

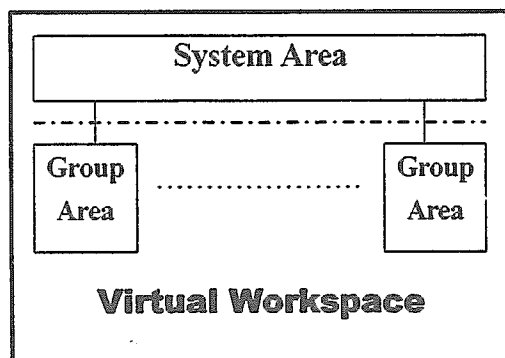
因為 WebCow 系統在伺服器端的硬碟部份在此系統中有一定的重要地位，因此我們特稱此部份為 Virtual Workspace，而 Virtual Workspace 這種共用區間的觀念也是現今世界上許多 CSCW 系統的核心。Virtual Workspace 的架構圖（如圖五）主要由兩個部分所組成：System Area 與 Group Area。在設計實作時，System Area 為伺服器端硬碟中的一個目錄，而每個 Group Area 則為 System Area 的子目錄，System Area 主要放置一些與系統有關的資料，而 Group Area 放置各個 group 自己的資料：

System Area: 放置

- a. 系統資料
- b. 系統訊息
- c. 所有群體與其成員的資料 (password, real name, nick name, e-mail address 等)
- d. 現在系統線上的群體與使用者的資料

Group Area: 放置

- a. group 共用資料
- b. 溝通訊息資料
- c. 電子白板完整資料與最新修改資料
- d. 文件完整資料與最新修改資料



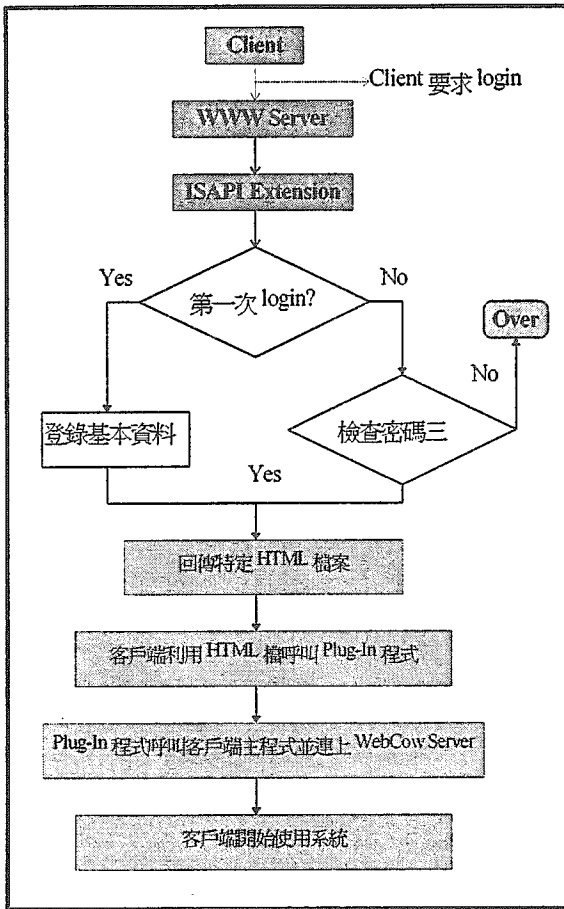
圖五 Virtual Workspace 架構圖

WebCow 系統運作的步驟如下（圖六）：

1. 客戶端利用 Netscape 瀏覽器連到 WebCow 系統的 WWW 伺服器。
2. WWW 伺服器的 ISAPI Extension 程式判斷若是新的使用者要求登入使用系統，則要求其建立基本相關資料(如姓名、暱稱、密碼等)，若是已登記過的使用者則判別其密碼。
3. 若成功則傳回一個含有特殊 MIME Type 檔案的 HTML 檔案，此檔案除了含有此特定的 MIME Type 檔案以呼叫客戶端的 Plug-In 程式外還含有這個客戶端所屬群的最新白板及文件資料。
4. 客戶端利用這個 HTML 檔案呼叫 WebCow 系統客戶端的 Plug-In 程式來處理這個 MIME Type 檔案。
5. Plug-In 程式被呼叫起來後便呼叫整個 WebCow 系統客戶端的程式包含電子白板、文件編輯等客戶端並與 WebCow Server 建立連線便可以開始使用 WebCow 系統。

WebCow Server 運作的步驟如下（圖七）

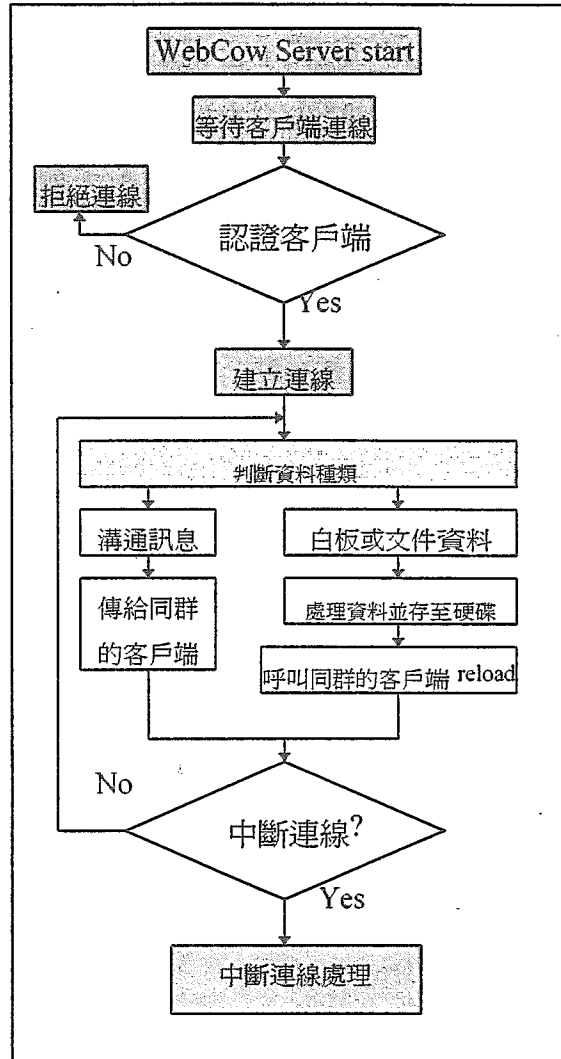
1. 等待客戶端的連線。
2. 認證要求連線的客戶端，判斷此客戶端是否已在 WWW 伺服器的 ISAPI Extension 完成使用登記。
3. 若未完成登記就拒絕其連線的要求，若已完成登記便對此客戶端提供服務。
4. 若客戶端上傳的資料為的溝通訊息則回傳給同群的其他客戶端
5. 若上傳的資料為電子白板或文件資料則將最新更新資料與整合後的完整資料分別存在 Virtual Workspace 並對同群的其他客戶端發出 Plug-In reload 的更新訊息。
6. 若客戶端要求結束使用，則做中斷處理，中斷此客戶端連線並對 ISAPI Extension 發出此客戶端離開系統的訊息。



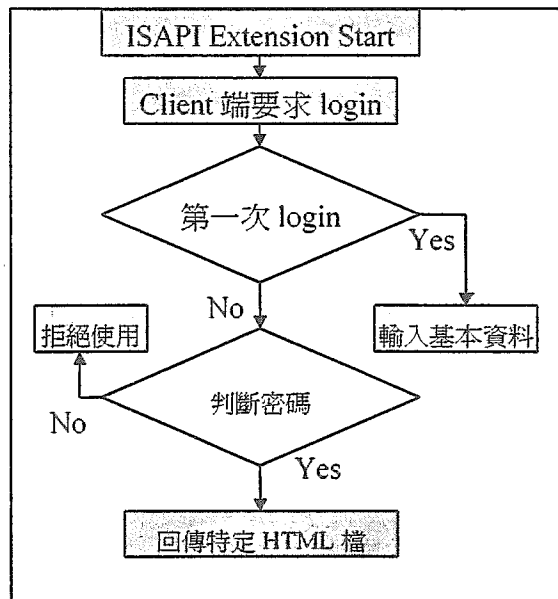
圖六 WebCow 系統運作流程

ISAPI Extension 運作流程如下 (圖八) :

1. 判斷是否第一次登入 (login)，若是則建立使用者基本資料。
2. 若此客戶端之前已建立資料，則檢查其密碼。若不正確就拒絕使用系統。
3. 若正確則回傳內含特定 MIME Type 的 HTML 檔案到 WWW Server 以傳回給客戶端呼叫 Plug-In 程式。



圖七 WebCow Server 運作流程

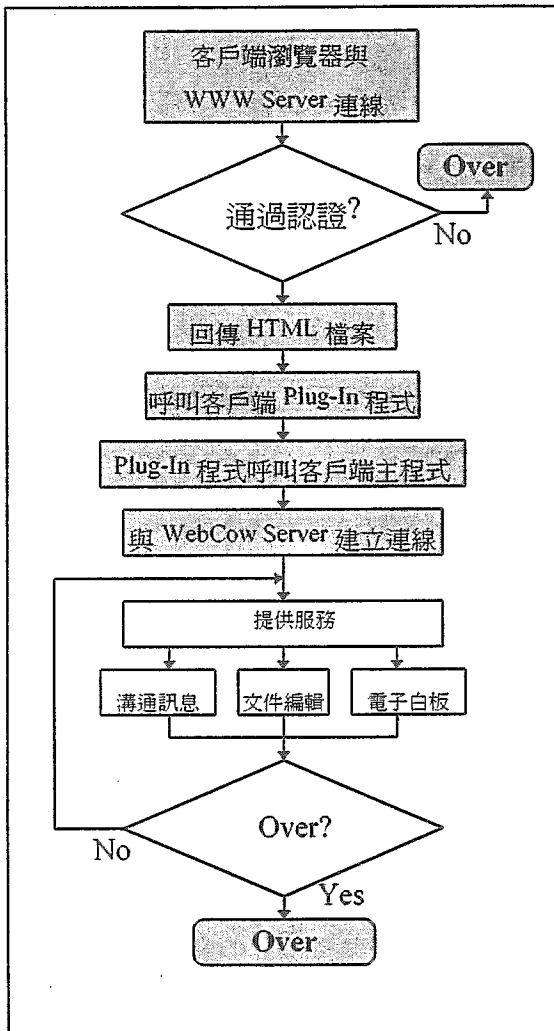


圖八 ISAPI Extension 運作流程

WebCow 系統客戶端運作流程如下 (圖九) :

1. 客戶端瀏覽器與 WebCow 系統的 WWW Server 連線
2. 通過登錄驗證並傳回含有 MIME Type 檔的 HTML 檔案
3. 瀏覽器呼叫客戶端 Plug-In 程式來處理此 MIME Type 檔案
4. Plug-In 程式下載最新白板與文件資料, 並呼叫客戶端主程式
5. 客戶端主程式與 WebCow Server 建立連線
6. 客戶端開始提供服務, 處理溝通訊息、文件與白板資料的上傳、下載與顯示直到結束服務。

- Pentium Pro 200 with 32MB RAM and Enhanced IDE 2.5GB HD (Server 端)
- Pentium 166 with 32MB RAM and SCSI 1GB HD (Client 端)
- Microsoft Windows NT 4.0 中文版+IIS (Internet Information Server) 2.0 (Server 端)
- Microsoft Windows 95 中文版 (Client 端)
- Netscape Navigator 3.0, Netscape Communicator 4.0 Preview Release 2
- Microsoft Visual C++ Version 4.2, MFC (Microsoft Foundation Class) 4.2



圖九 WebCow Client 端運作流程

五、系統實作

WebCow 系統伺服器端的設計環境是架構在 Microsoft Windows NT 上的 32 位元作業系統, 而客戶端則是在 Microsoft Windows 95 作業系統, 整套系統的發展環境與使用語言如下:

現在完成的部分在客戶端有文件編輯、電子白板、溝通訊息處理與檔案傳輸, 在伺服器端有伺服器網路效能計算與 WebCow Server 部分。

六、結論與未來工作

由現在已完成的部分經實際驗證在此套系統架構下確實可行, 希望在未來系統完成後能夠將測試結果與分析做進一步展示

參考資料

- [1] Ing-Chau Chang, Wei-Hsin Tseng, Yuh-Ming Huang, Tzong-Jer Yang, Jau-Hsiung Huang, Ja-Ling Wu, Horng-Ming Shiu, Ya Cheng, Weng-Ruey Deng and Chung-Wei Lee, "The Design and Implementation of a Multimedia CSCW System for Office Environments", *Proc. of Real-time and Media Systems*, 1995.
- [2] K. Gregory, P. Robichaux, B. Merkel and M. Pope, "Building Internet Applications with Visual C++", Que publishing, 1995.
- [3] M. B. Twidale, D. M. Nichols and C. D. Paice, "Browsing Is a Collaborative Process", <http://www.comp.lances.ac.uk/computer/projects/ariadne/docs/bcp.html>. 1996.
- [4] S. Prata, "C++ Primer Plus Second Edition", Waite Group Press, 1996.
- [5] T. Sheldon, "The Windows NT Web Server Handbook", McGraw-Hill, 1996.
- [6] Z. Oliphant, "Programming Netscape Plug-Ins", Sams.net publishing, 1996.
- [7] 李炳中、黃耀賢、王宗銘, "多媒體圖文共同編輯系統之設計與實現", *Proceedings of the second Symposium on Computer and Communication Technology*, Taichung, Taiwan, pp. 283-292, 1996.