

# 發展一個協同科學視覺化系統之技術

## Techniques of Developing a Collaborative Scientific Visualization System

王宗銘、李俊賢  
國立中興大學資訊科學所  
台中市國光路 250 號  
cmwang@cs.nchu.edu.tw  
jslee@cs.nchu.edu.tw

林尚毅、朱賢良、張宏生  
國家高速電腦中心  
新竹市科學園區研發六路七號  
lsi@cs.nchu.edu.tw  
[c00chu00@nchc.gov.tw](mailto:c00chu00@nchc.gov.tw)  
[a00chc00@nchc.gov.tw](mailto:a00chc00@nchc.gov.tw)

### 摘要

本論文敘述發展一個協同科學視覺化系統的技巧。我們提出一個協同視覺的系統架構，並以 Server-Client 方式、配合網際網路瀏覽器實現之。系統的 Member Server 負責提供成員註冊、分群等之管理資訊服務；CV Server 則提供各種視覺、同步等的協同互動服務。協助成員首先完成註冊並經系統編碼確認後，透過網路傳送系列的協同訊息給 CV Server，其他成員則下載該協同訊息，並藉由 VTK 公用軟體將科學資料在所屬平台計算、即時顯示來以完成一個 Cycle 的同步、協同、互動。我們將 Server 置於國家高速電腦中心，六位成員分別位於中興大學、中央大學及中山大學。經測試後，結果顯示系統可以達到幾近即時的優異表現，驗證我們所規劃系統架構之可行性。**關鍵詞：**協同視覺化顯示、網際網路、電腦輔助合作工作

### Abstract

Current visualization systems are designed around a single user model, making it awkward for large research teams to collectively analyze large data sets. This paper presents our investigation for the potential of making collaborative scientific visualization across the Internet. We propose an architecture and implementation details for server-side visualization system, together with authentication and encryption to support both asynchronous and multi-user collaboration. These ideas are implementation on platforms with internet-based computing resources. Experimental results illustrate that this system can achieve nearly real-time performance when visualize scientific data with moderate model complexity under 6 collaborative participants, regardless of the traffic loads encountered in the networks.

**Keywords:** Collaborative Scientific Visualization, Internet, Computer-Supported Cooperative Work.

### 一、簡介

近幾年來由於上網人口的普及，造成網路應用延伸，加上網路頻寬的成長，使得協同軟體的發展日漸蓬勃，許多 2D 或 3D 的 Groupware 如雨後春筍般出現在市面。這些協同軟體的共通特性就是利用網路的連結來跨越應用軟體使用空間的限制[6, 11, 14, 19, 21, 23]。

發展協同軟體其因有三：首先是工作效能的考量；因為團結產生力量，通常一件工作由十個人同時貢獻智慧進行比一個人獨自進行來得有效；其次是群體的創造力與思考力通常遠勝於一人的能力；最後，協同軟體可以成就資訊的分享；共同透過網路完成一件工作，參與的成員可以分享工作成果，得到工作完成的喜悅與滿足感。上述三項原因促使協同軟體開創了另軟體發展的另一個生存空間[1, 3, 6, 8, 11, 14, 19, 21, 23]。

科學視覺化(Scientific Visualization)的目的乃是將大量的資料(Data)透過視覺化的技巧以 3D 圖形的方式展現，俾使人們能據此對問題作判斷、分析，或對求解過程作修正、改進，進而能進一步剖析資料的含意、洞悉問題的本質[9]。科學視覺化起源於 1987 年 McCormick 所發表的報告「Visualization in Scientific Computing」，通稱 ViSC report [17]。該報告將計算機圖學(computer graphics)、影像處理(image processing)、使用者介面(user interface studies)、計算機輔助設計(computer aided design)等領域串成科學視覺化學科(Disciplines of Scientific Visualization)。目前科學視算的應用領域涵蓋極廣，例如大氣科學的氣象分析、預測，流體力學的流場設計、流速分析，地球科學的環境工程與環保評估，醫學工程的醫學影像應用，生物工程的藥品分析與設計等，均仰賴不同程度的科學視視覺化技巧[9, 15, 16]。

早期所發展的科學視覺化，由於計算速度

表 1. 相關的協同科學視覺化系統

編號	型態	系統	應用領域	文獻
1		RIVERS 系統	空氣動力學(Aerodynamics)	[13]
2	影像共享	Tempus Fugit 系統	計算流體力學(Computational Fluid Dynamics)	[11]
3		HIGHEND 系統	空氣動力學(Aerodynamics)	[18]
4		COVISE 系統	高頻寬歐洲結合網路 (PAGEIN Project)	[25]
5		SHASTRA 系統	多媒體計算(Multimedia Computing)	[2]
6		TeleInVivo 系統	網路、科學計算(Network or Scientific Computing)	[7]
7	控制共享	Cspray 系統	科學計算(Scientific Computing)	[20]
8		MICE 系統	分子模擬(Molecular Collaborative Environment)	[10]
9		COVISA 系統	科學計算(Scientific Computing)	[4]

的考量，作業平台大多選擇工作站、大型主機、或超級電腦；單一使用者(a single user)透過系統所提供的介面使用科學視算系統並顯示計算結果。若一群科學家想要合作解決複雜的計算問題時，通常合作的方式限定他們的研究及討論必須同時集中在平台所屬的地理位置(geographical location)。換言之，他們無法以各自的方式、在各自的地方、對問題及計算所得之資料進行剖析、探討、研究並互相交換研究心得。協同科學視覺化概念(Collaborative Scientific Visualization, CSV)乃因應而興起[5, 10]。CSV 能有效的跨越地理位置的限制，使的資訊的傳遞、研究心得的交換、僅在於彈指間。因此，無論是舊有的或是規劃中的系統，莫不將協同(Collaboration)列為更新舊系統或發展新系統所必須支援的首要項目。

九十年代起，微處理器技術的大幅進步，各種圖形加速卡相繼誕生，個人電腦之計算速度已大幅提昇。許多發展的系統逐漸將執行平台由八零年代的工作站、大型主機、甚或超級電腦轉移至個人電腦平台。網際網路的長足發展更推波助瀾這一個趨勢。全球資訊網(World-Wide Web)兼具使用上的便利性及瀏覽時的互動性，加上能夠動態地呈現出各種豐富的媒體效果，使得瀏覽器(Browser)成為網際網路程式設計的新寵，廠商也紛紛發展以 Web 為基礎的多層式(N-tiers)網路應用程式，就連作業系統廠商也都將瀏覽器納入為作業系統的核心。可以預見未來的作業系統將會是一個以瀏覽器為使用者操作界面的視窗時代[22]。

有鑑於這些主流因素，我們構思建構出一個以 Web 為基礎的協同視覺顯示系統。這個系統以提供 3D 視覺化的運作環境為其架構主軸。我們採用了一套在單機上廣為使用認可的科學視算工具庫 VTK (Visualization Toolkit)來作為我們視覺化運算的處理工具[24]。我們規劃出來的系統架構及協同機制，設計並實作出以 Web 為基礎的協同視覺化系統(Collaborative Scientific Visualization System, CSVS)。本文編排如下：第二節回顧相關系

統；第三節敘述 CSV 系統；第四節分析系統的測試數據。最後，我們總結本文並提出未來可能的工作。

## 二、相關工作

1989 年全球首次公開展示協同視覺化系統[13]。隨著時間相關系統相繼被發展出來[2, 4, 7, 12, 18, 20, 25]。其中，僅有 COVISA 系統是以 World-Wide Web 為基礎，架構在 Silicon Graphics 工作站的 IRIS 作業系統下[4]。文獻回顧顯示協同視覺化系統歸類為三種不同的協同合作方式：螢幕共享(Shared-Screen)、影像共享 (Shared-Image)、控制共享 (Shared-Control) [4, 10, 26]。表 1 依照文獻年代出現的順序，依序列出目前較為著名的系統名稱、應用領域及相關參考文件。由於螢幕共享為早期老舊的作法，無法獲得相關的系統資料，故並未列在表中。

螢幕共享就是協同成員分享同一個類似虛擬的螢幕，此類方法大都適用處理例如文字等資料量小的媒體，此方式溯至 Unix 時代的 X-Window，近期 Microsoft 公司發展的 NetMeeting 及 Norton 公司推出的 PC-Anywhere 等也是隸屬此類方式。此作法有兩個缺失，其一為引起控制權的競爭(Race)狀況，此乃由於每個 Client 都可以操控 Server 的程式畫面所導致。其二為每當 Server 有重繪的動作時就必須送出資料給所有遠端的 Client 進行更新顯示，如此將會花費很多時間在網路傳輸，尤其當使用者持續對較複雜的場景作協同工作時，再大的網路頻寬也將顯得捉襟見窘。

影像共享是協同成員以影像分享的方式來進行協同工作[12, 13, 18]。例如：醫院的遠端診療、遠距教學、視訊會議等。此種方式通常會在遠端先準備圖片或影像，等到預覽結果大致滿意後，再把整個圖片傳送給其他端的使用者瀏覽。雖然這種方式優於螢幕共享，但也有兩個缺點：其一是影像資料量的需求通常都不僅是單張就可滿足，因此經過多張影像資料

的累積後，資料通常都膨脹變成相當龐大；其二，若是網路傳輸速度較慢時，上述龐大的資料量可能造成畫面延遲或是跳格，甚至可能發生畫面僅出現部分就因等待頻寬來傳送資料而暫時停止。

控制共享則是採用控制分享的方式來進行協同工作[2, 4, 7, 20, 25]。此法通常利用傳送控制參數來達到協同工作的目的。一些較為先進的協同軟體，如支援 Computer-Supported Cooperative Work (CSCW) 的各種群組軟體 (Groupware) 皆使用這種方式來進行協同工作 [6, 11, 14, 19, 21, 23]。此種方法有三點好處：其一是所需的資料量相對相當小；這是因為通常每筆控制參數的資料量都不會超過 3 KB，所以縱使一次傳送好幾筆控制參數，總和也不過數十個 KB，遠小於影像動輒數個 MB 的資料量。其二為網路頻寬對資料傳輸的影響較不明顯：此乃因資料量小，故僅需極少的頻寬即可完成傳輸的動作，減少網路傳輸所潛在的影響與衝擊。其三為協同工作的分界明確：這是因為成員接獲控制參數後，即可完全掌控協同工作的後續動作，直至下一次的同步為止，此大幅避免共享影像所可能產生的影像中斷或畫面跳格的情形。然而，此方法最大的缺點為模型檔案的大小將受到相對的限制；這是因為由於僅傳送控制參數，因此，每位協同成員的本地端機器應至少需能儲存協同成員共用的模型檔。由於目前單位記憶體的價格更加便宜，故未來可以預期每台電腦均會使用比目前更多的記憶體，因此，該項缺點未來可以藉由記憶體而克服部分問題。

綜合上述，部分軟體的確能提供協同，例如支援 CSCW (Computer Supported Co-operative Works) 的各種群組軟體 (Groupware)，然而這些軟體均無提供視覺化 (Visualization) 的環境，充其量僅止於支援到“同步”協同而已。因此，我們認為有迫切的必要來發展以網際網路為基底，可以支援科學視算的協同視覺顯示系統。這個系統應能在 Web 環境上執行，也應採用共享控制的方式來作為同步顯示的傳送法則，以充分減少對網路頻寬的需求。

我們經過全盤的考量後決定採用科學視算公用軟體 Visualization Tool Kit (VTK) 來作為 CSV 系統發展的基礎[24]。我們考量的原因有四項：1. VTK 是一套以物件導向 (Object-Oriented) 的觀念所發展的函式庫，發展系統較為容易；2. VTK 會直接呼叫 OpenGL 來進行在成圖計算，此舉使的成圖程式碼的複雜度相對降低許多；3. 使用 VTK 可以受惠於物件導向程式的優點，亦即節省重複開發程式碼及物件循環再利用 4. VTK 透過 OpenGL 來做成圖計算，故成圖可以受惠於價廉物美的硬

體加速卡，獲致即時互動的效能。然而使用 VTK 亦有其必須克服之困難之處：1. VTK 僅提供非常簡單、非瀏覽器型態的使用者界面，故使用者介面必須重新設計、發展；2. VTK 僅限定在單一個平台上使用，因此我們必須將之修正成為支援協同工作，這包含發展協同工作的定義、發展適合的物件、開發原始程式碼等工作；3. VTK 並不支援網路通訊，因此網路資料傳輸的部分及協同工作的同步等必須重新設計、發展。因此，雖然以 VTK 為系統的基礎，仍有極大部分仍必須重新規劃、設計，方能在網際網路上實現 CSV 系統。

### 三、系統設計與實現

#### 3.1 設計目標

CSCW 的分類可藉由地點的異同與時間的非同步而發展成為不同特色的系統[11, 19, 21]。同樣的，我們在規劃 CSV 系統時，依據地點、時間、成員等三項因素，將系統定位為能支援擁有「異地」(Distinct Location)、「異時」(Asynchronous Time)、「多重使用者」(Multi-User) 等三種協同模式的系統。換言之，系統提供不同時間、不同地理位置、並藉由記錄檔 (Record) 提供多重成員的協同工作。

CSV 有六項設計目標，分別為 1. 跨平台；2. 跨區域網路：系統架構無需任何巨幅變更，可以透過 Internet 做跨國的延伸；3. 非同步連線：其他成員可以下載或上傳預先錄製好的記錄檔 (Record File) 來進行非同步的協同工作；4. 嚴密安全性：系統對每次的連線進行嚴格的過濾，對每位使用者做嚴密的認證，以杜絕未經許可的非法連線，防止他人惡意破壞；5. 即時的反應：Client 與 Server 間所傳送的内容應是資料量小的「控制參數」，而非資料量龐大的「圖形影像」。6. 便利的使用者界面：系統必須結合 Web 特性，並提供完善、易使用的圖形化使用者界面 (GUI)。

#### 3.2 協同工作之函數表示

CSV 系統設計能提供多群體，多使用者動態進入使用並離開系統。以下我們以集合的方式來定義多群體、多使用者，並利用運作函數來描述多群體、多使用者之行為。此運作函數即為吾人於實現時，需利用程式語言完成之函數之部分。

**定義：**

1. 集合  $U$  為一具有有限成員 ( $G_i$ ) 的有限集合 (Ordered Set)  $U = \{ G_i \mid G_i \in U, i = 1, 2, \dots, n \}$
2. 集合  $G_i$  亦為一具有有限成員 ( $X_j$ ) 的有限集

- 合  $G = \{ X_j | X_j \in G_i, j = 1, 2, \dots, n \}$
- 當集合  $G$  內沒有成員時稱此集合  $G$  為空集合 (Empty Set), 以  $\{\}$  表示。
  - 我們稱兩個集合  $G_1$  與  $G_2$  相等, 當  $G_1$  與  $G_2$  具有完全相同的成員
  - 當  $G_1 \cap G_2 = \phi$ , 我們稱  $G_1$  與  $G_2$  為互斥 (Mutual Exclusive), 即  $G_1$  與  $G_2$  沒有任何相同的成員。

我們以運作函數描述群組 (Group) 與成員 (User) 的行為。運作函數如下所示：

- Empty ( $G$ ): 當一個群組  $G$  其內無任何成員時, 我們稱此群組  $G$  為空群。此時, 系統會移去此空的群組。
- Participant ( $G_i$ ):  $Participant(G_i) = \{ X_j | X_j \in G_i, j = 1, 2, \dots, n \}$ , 透過此函數可找出群組  $G_i$  的所有成員。
- Create ( $G_i, X$ ): 成員  $X$  可在系統中自組一個工作群組  $G_i$ 。
- Append ( $G_i, X$ ):  $G_i \cup X$ : 成員  $X$  可加入群組  $G_i$  中, 系統會將成員  $X$  的相關資料如暱稱 (Nickname) 與密碼 (Password) 記錄於系統中。
- Participant-information ( $X_j$ ): 成員  $X_j$  的基本資料包括姓名 (Name)、年齡 (Age)、電子郵件位址 (E-mail Address)、所屬群組名稱 (Group Name)、使用暱稱 (Nickname) 及密碼 ( P a s s w o r d ) 。
- Leave ( $G_i, X_j$ ):  $G_i - \{ X_j \}$ : 當群組成員  $X_j$  要離開系統時, 系統會自動通知同群組其他成員。例如  $G_1 = \{ X_4, X_2, X_1, X_3 \}$ , 當  $X_1$  想要離開, 則  $G_1 = G_1 - \{ X_1 \} = \{ X_4, X_2, X_3 \}$ 。

### 3.3 系統架構

系統的主體架構大致分成二大類別、四大部分, 如圖 1 所示。其中 (1) Member Service: 提供成員資訊服務, 包含 Member Server 與 Client; (2) CV Service: 提供協同視覺化服務, 包含 CV Server 與 Client。為能兼顧系統的彈性運用與日後的擴充便利, 此二大類別可單獨執行, 提供服務。各類別之功能如下：

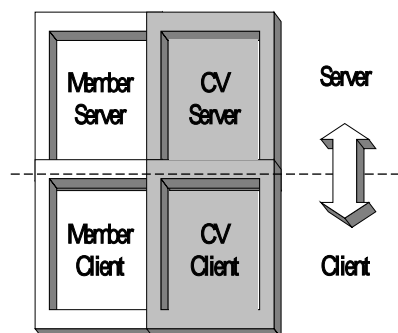


圖 1. 系統主體架構的四大部分

**1. Member Server:** 專門負責使用者的註冊以及使用者的登入。這個 Server 是由 WWW Server 來擔任, 利用 Server 端語言 (如 CGI、ASP、JSP、PHP 等) 的控制與 Database 的連結, 來處理 Member Client 在瀏覽器上的登入驗證並執行使用者所選擇的服務。Member Server 主要是由五個單元組合而成。(1) System Kernel: 為 Member Server 的主要核心, 負責與各個單元進行協調溝通的工作。(2) Control Unit: 負責使用者登入的條件檢查與成員的連線控制。(3) Access Unit: 專門用來儲存使用者以及系統的相關資訊。(4) Process Unit: 用來處理使用者所要求的服務以及提供一個 GUI 界面並執行管理者透過此界面所下的命令。(5) Communication Unit: 專門負責 Member Server 對外的通訊, 例如利用 http 協定與 Member Client 進行連線, 或是透過 Java Socket 來與 CV Server 相互溝通。

**2. Member Client:** 提供瀏覽器界面給使用者登入以及提供相關的服務, 並藉由 Dynamic HTML (DHTML 如 JavaScript、ActiveX 等) 的控制, 提供一些使用者登入的訊息 (如成功、失敗或是人數過多等) 以及提供相關服務的連結。Member Client 也是由五個單元組合而成: (1) Browser: 係主要的運作核心, 所有的單元皆架構其上。(2) Control Unit: 負責回應 Member Server 的一些控制命令。(3) Access Unit: 對於使用者選擇的服務進行相對應的程式呼叫。(4) Display Unit: 提供一系列的使用者界面。(5) Communication Unit: 專門負責與 Member Server 進行溝通協調的任務。

**3. CV Server:** 專門負責協同視覺化合作的溝通工作以及相關的認證動作, 如圖 2 所示。

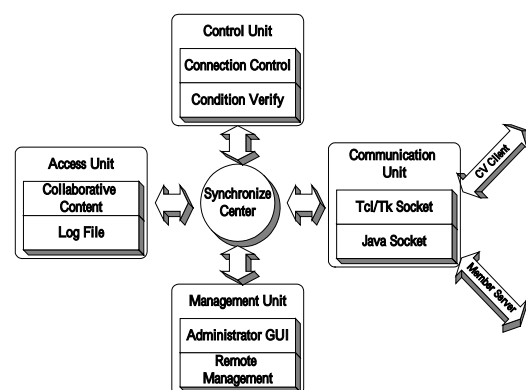


圖 2. CV Server 架構

此部份是以 Tcl/Tk 程式為主軸, 再配合 Java 的運用來監控管理每一個連線與每一個使用者的行為。首先它會等待 Member Server 的通知, 將 Member Client 的資訊加入其成員資料庫, 然後再等待 CV Client 的連線以進行

“協同視覺化”工作。另外，CV Server 還提供了一系列的管理界面以利管理者能夠充分管理。CV Server 也由五個單元所組合而成：(1) Synchronize Center：負責所有 CV Client 的協同動作與保持視覺化 Model 的一致性。(2) Control Unit：負責 CV Client 的登入與驗證連線的動作。(3) Access Unit：用來作為協同工作時的共用儲存所(Work Space)，除了可以存取每一位成員協同動作的內容外，還可以記錄所有網路連線的相關資訊。(4) Management Unit：可以讓本地以及遠端的管理者進行 CV Server 的負載調整與線上成員管理。(5) Communication Unit：專門負責從 CV Client 接收與傳送協同工作的內容以及接受遠端管理者與 Member Server 的控制命令。

4. CV Client：提供協同視覺化工作的界面給使用者，如圖 3 所示。這個部份是以 Tcl/Tk 程式為主軸，輔以 Java 的配合，提供使用者進行互動式的“協同視覺化”環境。程式通常是由 Member Client 從 Member Server 收到 CV Server 的資訊後，自動地被呼叫起來並連接到 CV Server 去進行使用者的登入動作，等到 CV Server 驗證身分無誤後就開始進行“協同視覺化”工作。

CV Client 則由七個單元組成：(1) Message Parser：負責接受位於前景使用者的所有動作以及接收在背景由 CV Server 送來的協同資料，然後經過分析再交由其他單元來進行處理。(2) Control Unit：負責控制每一筆協同工作的同步接收時間，以及調整每一個播放動作的更新間隔時間。

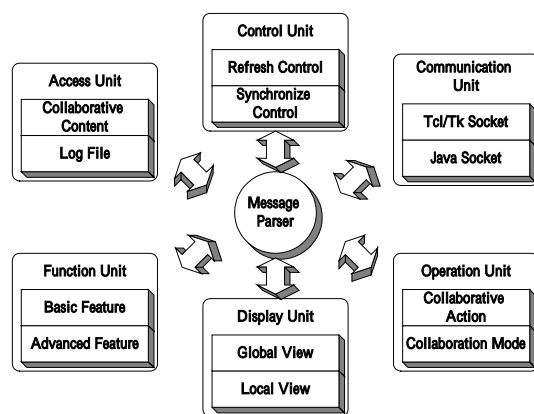


圖 3. CV Client 架構

(3) Access Unit：提供一個協同工作的作業緩衝區(Work Buffer)，負責協同工作內容的記載以及相關資訊的記錄。(4) Function Unit：負責提供一些協同運作的基本功能以及一些 3D 視覺化方面的進階功能。(5) Display Unit：負責提供協同視覺化工作的使用者界面，以及顯示一些相關狀態的提示訊息。(6) Operation

Unit：提供一些協同運作的機制(如：Single 單機模式、Interactive 互動模式、Lock 鎖住模式)，並對相對應的工作模式進行後置處理。(7) Communication Unit：負責與 CV Server 進行同步連線及傳送、接收協同工作資料。

### 3.4 安全架構

系統架構在 Internet 上，故安全性課題顯得份外重要。一般而言，安全性不離認證(Authentication)與保密(Confidentiality)兩種議題。Server 與 Client 間的通訊具有三項安全性考量：1.主機認證(Host Authentication)：針對每一個連線 IP 與資料庫中合法的主機位址做確認；2.使用者認證(User Authentication)：針對每一個合法 IP 上的使用者與資料庫裡的成員名稱做比對；3.使用者名稱加密(User Encryption)：對於每一個使用者的登入名稱在傳送前先作加密處理，如此可避免他人的偷窺及修改，且也可利用此加密字串當作兩者間傳輸資料加解密使用的對稱鑰匙(Session Key)。對稱鑰匙的所有人才可以解開兩者間傳送的內容，倘若訊息被攔截，所顯視的也是一串無意義的密文。

CSV 系統的安全架構如圖 4 所示。首先 CV Server 會對每一個 CV Client 做 IP 與 Username 的認證，其認證的依據是 Member Table，另外為了避免使用者的資訊被盜用以及被冒充，我們利用使用者所在的 IP 與 Username 加密成一段密文，此密文相當於個人的身份證，用來作辨識 CV Client 的真偽性，然後再將密文作為 Server 與 Client 間傳接資料加解密用的對稱鑰匙。至於 Member Table 僅允許由 (1) Config File (2) Local Administrator (3) Member Server (4) Remote Administrator 來進行變更。這四個人的認證資訊置於 Allow Table 內，其認證的步驟與 CV Client 類似。為求安全性起見，Allow Table 只能由 (1) Config File (2) Local Administrator 來變更內容，免去被冒充的可能性。

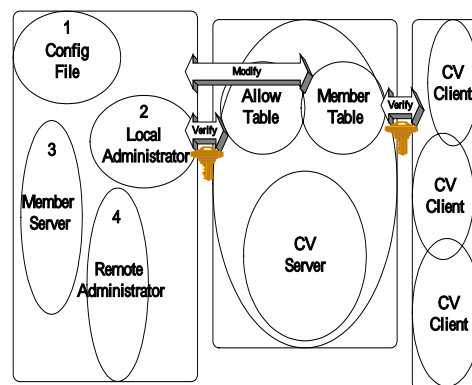


圖 4. CSV 系統安全架構圖



### 3.5 協同工作界面

協同工作之 CV Client 界面如圖 5 所示。由於 Global View 視窗與 Local View 視窗之介面相似，以下依照編號順序分別介紹 Local View 視窗中的各種選單或按鍵之功能。

**1. 系統選單列：**總共有 File、Record、Edit、View、Action、NPR、Option、Help 等八項。File 是載入與儲存 Model 檔用的；Record 是匯入與匯出記錄檔用的；Edit 是對 Model 進行一些特殊的處理；View 是用來改變使用者的視點位址與 Model 的顯示方式(如：Feature Edges、Boundary Edges、Non-manifold Edges 等等)；Action 則是用來變更系統成圖方式(Wireframe、Surface、NPR 等)與協同工作模式(Lock、Single、Inter-Active)；NPR 則是用來選擇 NPR 的顯示方式(OBJ(object)、SIL(silhouette)、OBJ\_AND\_SIL)、以及 NPR 的種類(SIL\_4Pass、SIL\_WireF、HighLight)；Option 則是提供一些進階的功能(如：Model Edit、Compare Results、改變前景與背景的颜色、立即同步送出等)；Help 則會顯示本程式的一些相關資訊。

**2. 協同模式列：**可以讓成員選擇目前的協同模式；其中，Lock 模式代表目前是處於鎖住狀態，此時使用者不能對 Model 進行任何的動作(此為 Global View 視窗的預設工作模式)；Inter-Active 模式則是一般正常的協同互動模式，如圖 5 所示。通常協同工作皆會在此模式下進行(此為 Local View 視窗的預設工作模式)；Single 模式則是表示目前處於單機狀態，所有的動作只在本機進行而不會被送到 CV Server 去，通常這是當使用者要暫時退出協同群組來進行私人工作時才會採用，此外使用者可以此模式做練習來求得較好的工作成果。

**3. 系統狀態列：**用來顯示系統的相關狀態，例如 Local View 或是 Global View，以及目前的協同工作模式等等。

**4. 訊息提示列：**用來顯示目前的工作訊息，例如連線成功、連線失敗、登入失敗、協同 Model 來源、協同工作來源、目前同步記錄的來源等。

**5. 成圖視窗：**這個視窗是用來顯示目前這個 Model 在經過成圖計算後的輸出結果。

**6. Model 狀態列：**用來顯示目前這個 Model 的相關資訊，例如其 Vertices、Cells 個數以及目前的 View Point 等。

**7. 連線狀態列：**顯示 CV Server 的位址、CV Server 的連線 Port 編號、目前連線狀態、以及使用者的登入名稱等。

**8. 成圖方式選擇列：**可以選擇 Wireframe、Surface、NPR 等三種不同的成圖方式，如圖 5 所示。其中，NPR 的成圖方式在 VTK 中並不支援，此部分完全是我們所獨立自行研發。Wireframe Silhouette 與 4-Pass Silhouette 都是用來描繪物體的外框，而 NPR HighLight 則是用來顯示物體的高亮點。

**9. 屬性設定列：**可分別設定前景(foreground)及背景(background)的屬性，例如 Specular、Specular Power、Ambient、Diffuse、Color。

**10. 協同工作列：**總共有 Send、Stop、Rept、Preview、Discard、Review、Rest、Last、Recv 等九項功能。其中 Send 負責傳送本次的協同動作到 CV Server 去；Recv 則負責將 CV Server 送過來的協同資料放映出來；Rept 會重播上一次從 CV Server 所送過來的動作；Rest 則會重播所有 CV Server 送過來的協同動作；Preview 可以讓使用者在送出本次動作前先進行預覽；Discard 則是當使用者對預覽的結果不甚滿意時，便可將這次動作與予刪除；Review 會將使用者所有曾經送出過的動作進行重播；Stop 可以讓使用者停止正在播放中的動作；Last 則可以讓使用者在任何情況下返回到最新的動作。

**11. 記錄檔工作列：**共有 Play、Pause、Reset、Trace、Loop 等五項，如圖 5 所示。這個工作列通常只有在匯出或匯入記錄檔之後才有作用，首先 Play 會將記錄檔的內容與予播放；Pause 則是用來暫停正在放映中的動作；Reset 則會重置記錄檔的放映位置，使其重新位於檔案的起始位址；Trace 可以讓使用者進行記錄檔的單步追蹤，也就是按下一次 Trace 只會放映一個動作；Loop 則會一直不斷地重複播放整個記錄檔的動作，通常可在 Demo 時使用。

**12. 立體成圖選擇列：**在這裡使用者可以選擇是要以正常的或是以立體的成圖方式來進行瀏覽，其中立體的成圖方式共有 CrystalEyes、RedBlue 兩種可供選擇。當使用者以 CrystalEyes 的方式觀賞時須配戴 CrystalEyes 的立體眼鏡方可正常運作；選擇“RedBlue”的方式只需要配戴一般紅藍分光鏡製成的眼



圖 5. 協同工作之 CV Client 界面

鏡即可得到立體的效果。

**13. 同步時間設定列：**在這裡使用者可以設定從 CV Server 送來協同資料的反應顯示時間。一般而言，當 CV Client 收到 CV Server 所送來的資料時，會將其內容存放在作業緩衝區 (Work Buffer) 裡面，等到設定的同步時間計時器一到，就會先檢查作業緩衝區內是否有資料，若有則馬上進行協同動作的放映，然後再啟動下一次的同步時間計時器；若無則直接啟動下一次的同步時間計時器。本計時器共分為 0~10 個級數，其中 0 代表 CV Client 與 CV Server 保持最快的同步狀態，意即 CV Client 在收到 CV Server 送來的資料後會馬上播放協同動作的內容；而 1~9 級則會依序地增加 CV Client 與 CV Server 的同步時間；至於 10 則代表 CV Client 與 CV Server 之間的同步只有在按下 Recv 的按鈕後才會發生 (通常這個等級會用在 Local View 上面)，此外，CV Client 本身不會自己進行同步的動作。一般當使用者在改變級數後，計時器就會重新設定使得同步效果可以立即反應。

**14. 更新時間設定列：**這裡是用來設定每個播放動作的時間間隔用的，使用者可以經由變更時間間隔的長短來達到各種慢動作的播放，其中每一個間隔單位為 0.2 秒，同樣地當使用者在改變級數後計時器也會重新設定使得更新效果可以立即反應。

**15. 瀏覽動作選擇列：**在這裡可以讓使用者選擇三種不同的瀏覽方式。(1)放大縮小 (Zoom)：可以放大與縮小使用者的視野。(2)旋轉 (Rotate)：可以讓使用者的視野對 Model 進行旋轉。(3)移動 (Pan)：可以上下左右地移動使用者的視野。

## 四、系統測試與分析

### 4.1. 測試環境

我們將上述自行發展的 CSV 系統在網際網路環境中實際操作，並對系統做測試，將數據加以分析，以便瞭解 CSV 系統之整體效能。影響 CSV 系統效能共有六個要素 (Performance Factors) 分別為 (1) 網路傳輸速度 (2) Server 端硬碟存取速度 (此乃由於 Server 端並不執行計算工作，僅執行互動控制訊息的存檔動作之故) (3) Client 端的硬體運算速度 (4) 協同成員人數 (5) 協同模型的複雜度 (6) 協同工作本質。

綜合諸影響因子，我們的測試環境如表 2 所示。其中，Server 為工作站，置於國家高速電腦中心 (Spring)，並由中興資訊所做遠端控制。至於 Client 端則共配置六部異質工作站或個人電腦。其中，南部中山大學的海下技術所 (M-6) 配置一部 Sun 工作站；北部中央大學的

學生宿舍 (M-5) 配置一部個人電腦。其餘四部則全部使用個人電腦 (M-1 M4)，並置於中興大學資訊所的實驗室。此外，我們參照國外相關研究，將協同成員人數訂在 6 人以下 [4]。

表 2. 測試環境硬體資源一覽表

Machine Name	Machine Locations	Network (Mbps)	Memory (MB)	Clock (MHz)
Server	國高中心	100	512	250
M-1	中興大學	100	128	392
M-2	中興大學	100	256	450
M-3	中興大學	100	128	700
M-4	中興大學	100	128	700
M-5	中央大學	10	128	450
M-6	中山大學	10	128	167

文獻上並無標準的測試模型，因此，我們自行建構一個 Fighter 模型。Fighter 是一架戰鬥飛機的模型，共有 590 個 Points，541 個 Polygons。最後，為能客觀測試協同工作本質的變因，我們採用 200 個不同性質的協同工作並顯示加總的整體時間。這些協同工作包含瀏覽動作的選擇；例如旋轉 (Rotate)、移動 (Pan)、放大縮小 (Zoom In/Out)，成圖方式選擇；例如 Wireframe, Surface, NPR，以及協同工作列的選擇；例如 Preview, Repeat, Reset 等。

在協同工作時，我們定義了三種時間，分別代表不同意義。它們是 (1) 反應時間 (Response Time, RT)，(2) 通訊時間 (Communication Time, CT) 及 (3) 成圖時間 (Rendering Time, RET)。反應時間 (RT) 係代表接收端的 GlobalView 完成同步顯示的時間 (T3) 減去傳送端的 LocalView 開始傳送動作的時間 (T1)，亦即 (T3-T1)。此時間間隔是從某位協同成員的 LocalView 送出協同動作後開始，直到自己的 GlobalView 收到這各協同動作、計算並完成同步顯示為止。對於協同成員而言，這個反應時間的長短代表成員彼此間即時協同互動性之優劣程度。

通訊時間 (CT) 係代表接收端的 GlobalView 收到資料的時間 (T2) 減去傳送端的 LocalView 開始傳送資料的時間 (T1)，亦即 (T2-T1)。此時間間隔是從某一成員的 LocalView 上傳協同控制資料到 Server，經 Server 端的處理後，下傳該筆協同控制資料到各個成員接收端的 GlobalView 為止。通常此通訊時間為網路傳輸速度與 Server 端硬碟存取時間的總和。

最後，成圖時間 (RET) 係反應時間 (RT) 與通訊時間 (CT) 之差，也就是 ((T3-T1)-(T2-T1)) = (T3-T2)。這個時間是從接收端的 GlobalView 收到協同控制資料起，直到同步動作顯示完畢為止。成圖時間代表機器處理協同控制資料所需的計算與顯示時間。

表 3. Fighter 模型的協同工作測試時間 單位：秒

網路時刻	工作時間	M-1 (中興)	M-2 (中興)	M-3 (中興)	M-4 (中興)	M-5 (中央)	M-6 (中山)
尖峰	反應時間	11.39	10.55	8.41	8.28	15.03	18.03
	通訊時間	5.53	5.51	5.40	5.39	7.72	7.93
	成圖時間	5.86	5.04	3.01	2.89	7.31	10.10
離峰	反應時間	11.37	10.57	8.43	8.33	14.93	17.95
	通訊時間	5.50	5.47	5.40	5.36	7.66	7.92
	成圖時間	5.87	5.10	3.03	2.97	7.27	10.03

#### 4.2. 測試數據

表 3 列出 Fighter 模型在網路尖峰與離峰時所測得之數據。其中，網路尖峰、離峰之時刻係參照教育部電算中心所提供之資料。由表可知：Fighter 模型在網路尖峰時，最長的反應時間是位於中山大學海下所的 M-6，需時 18.03 秒，最短的是位於中興資訊所的 M-3，需時 8.41 秒。至於通訊時間，由於使用不同頻寬的網路傳輸，故略有差別；使用 100 MBps 的四部硬體，約需時 5.4 秒，而使用 10 MBps 的兩部硬體，約需時 7.72 秒以內。至於成圖時間，以 M-6 的 10.10 秒最多，而以 M-4 的 4.83 秒最少。在離峰時，各項數據並無巨幅的變動。值得一提的是：在網路尖峰與離峰時，通訊時間的改變微乎其微。

我們針對整體時間做分析，由圖 6 可以歸納得知：整體的反應時間最快的為 8.28 秒(使用 700 MHz 之機器)，最慢的為 18.03 秒(使用 167 MHz 之機器)；以最快為例，平均每個協同工作需時 0.0414 秒，此代表成圖能維持每秒至少 24 個 Frame 的互動水準；相對的，最慢的反應時間也至少能達到每秒 11 個 Frame 的水準。因此，整個的協同工作幾乎可以達到即時(Real Time)的顯示。使用者並未感覺系統有延遲(Delay)之現象。

針對成圖時間做分析由圖 6 我們可以歸納得知：通訊時間雖然與協同成員所在位置有關，但差別並不明顯。此外，網路傳輸速度(100/10 MBps)對通訊時間有一定的影響，此點相當直覺。但令人意外的是：我們發現系統執行時，網際網路無論處於尖峰或離峰流量時，通訊時間所受之影響不大。事實上，若使用相同的網路網路傳輸速度，則對於任何地方的任何使用者而言，通訊時間幾乎可以視為一個固定的時間。

綜合以上，我們發現：在上述的六個影響因子中，網際網路的流量並不是影響 CSV 系統的一個主要因素。這是因為 CSV 系統的成員在做相互的通訊時，所傳送的内容是資料量小的控制參數，而非資料量龐大的圖形檔案。此舉的確有效的減少資料傳輸量，縮短需求的

網路頻寬。由於無論網路處於尖峰與離峰情況下，其流量頻寬至少都有幾十個 Kbytes，足供協同工作控制參數的傳輸。

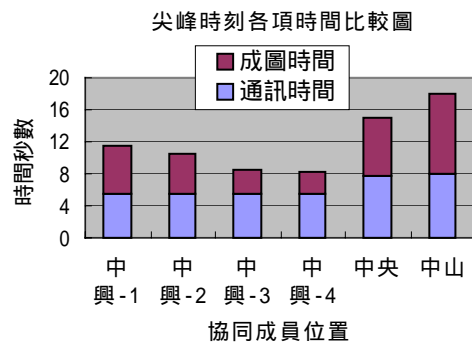


圖 6. 尖峰時刻數據比較圖

此外，Server 端的硬碟儲存速度也非主要的影響因素。在其他的影響因子固定下，我們認為 Client 端的硬體速度才是 CSV 系統的主要影響因素。計算能力越快的機器，所獲得的成圖時間就會越短，而在通訊時間幾乎固定不變的情況下，系統的反應時間就會越短。因此，Client 端的硬體速度具有絕對的影響力。

#### 五、結論與未來工作

本論文敘述發展一個協同科學視覺化系統的技巧。我們提出一個協同視覺的系統架構，並以 Server-Client 方式、配合網際網路瀏覽器實現之。在系統架構方面，為了達到良好的覆載平衡以及兩權分立的優點，我們採用了雙伺服器(Dual-Server)架構。不論是在 Server 端還是 Client 端，我們都會對連線狀態以及協同內容進行完整的事件記載。尤其是 Server 端對於非法的連線和違規的存取更是有詳細記錄，以確保整個協同工作進行時的安全。

對於系統效能方面，由於我們是採用 Shared-Control 傳送參數的方式，其中使用者每一個動作的傳送量大小為 0.3 Kbytes。經實際的系統測試後，發現到如此小的傳輸量不容易受到網路頻寬以及網路流量的影響，因而證明了當初我們採用傳送參數的抉擇是正確的。



總結本文,透過 CSV 系統之發展與實現,我們提供了一個可以讓身在異地的使用者以“同步”或“非同步”方式來進行 3D 視覺化工作的 Web-Based 協同環境,在這個環境裡,使用者可以充分運用我們所提供的服務來完成他們所需要的工作。同時我們也透過實際的系統測試來印證我們當初所提出來的系統架構具有一定的運作效能與安全性水準。

CSV 系統未來工作包含允許共有彈性的“線上編輯模型”、更多樣的 NPR 非擬真成圖方式、測試更多的協同成員、擴展跨國協同工作及更完備的系統維護。

## 六、致謝

本研究承蒙國科會專題計畫之經費補助(NSC-89-2218-E-005-008),謹向國科會工程處致謝。本研究成果榮獲 2000 年龍騰論文獎,謹向宏碁電腦公司致謝。最後,系統測試時,承蒙中央大學電機系林城伍先生、中山大學海下技術所陳家偉先生、國家高速電腦中心周朝宜先生之鼎力協助,謹致以作者由衷之謝忱。

## 參考文獻

- [1] M. Abbott and L. K. Jain, “DOVE: Distributed Objects Based Scientific Visualization Environment,” In *ACM 1998 Workshop on Java for High-Performance Network Computing*, pp. 56-68, ACM Press, 1998.
- [2] V. Anupam and C. Bajaj, “Collaborative Multimedia Scientific Design in SHASTRA,” *IEEE MultiMedia*, Vol. 1, No. 2, pp. 39-49, 1994.
- [3] S. Benford, C. Greenhalgh, T. Rodden, and J. Pycok, “Collaborative Virtual Environments,” *Communications of the ACM*, Vol. 44, No. 7, pp. 79-85, 2001.
- [4] K. Brodlie, J. Wood and A. Yeo, “Web-based Visualization: A Client Side Approach,” In *Proceedings of IEEE Workshop on Distributed Visualization Systems*, pp. 1-3, IEEE Press, 1998.
- [5] K. Brodlie, S. Lovegrove, and J. Wood, “Harnessing the Web for Scientific Visualization,” *Computer Graphics*, Vol. 34, No. 1, pp. 10-12, 2000.
- [6] A. Chabert, E. Grossman, L. S. Jackson, S. R. Pietrowiz, and C. Seguin, “Java Object-sharing in Habanero,” *Communications of the ACM*, Vol. 41, No. 6, pp. 69-76, June 1998.
- [7] J. Coleman, A. Goettsch, A. E. Klement, and P. Bono, “TeleInVivo: Towards Collaborative Volume Visualization Environments,” *Computers and Graphics*, Vol. 20, No. 6, pp. 801-811, 1996.
- [8] F. Costantini and C. Toinard, “Collaborative Learning with the Distributed Building Site Metaphor,” *IEEE MultiMedia*, Vol. 8, No. 3, pp. 21-29, 2001.
- [9] R. A. Earnshaw, “Scientific Visualization: the State of the Art,” *Physical World*, pp. 48-51, September 1993.
- [10] T. T. Elvins, “Introduction to Collaborative Visualization,” *Computer Graphics*, Vol. 32, No. 2, pp. 8-11, 1998.
- [11] J. Grudin, “Computer-Supported Cooperative Work: History and Focus,” *IEEE Computer*, Vol. 27, No. 5, pp. 19-26, May, 1994.
- [12] M. J. Gerald-Yamasaki, “Cooperative Visualization of Computational Fluid Dynamics,” *Computer Graphics Forum*, Vol. 12, No. 3, pp. 497-508, 1993.
- [13] R. B. Haber, “Scientific Visualization and the RIVERS Project at the National Center for Supercomputing Applications,” In *Proceedings of IEEE Visualization '90*, IEEE Press, pp. 231-236, 1990.
- [14] R. T. Kousez, J. D. Myers and W. A. Wulf, “Collaboratories: Doing Science on the Internet,” *IEEE Computer*, Vol. 29, No. 8, pp. 40-46, 1996.
- [15] K.-L. Ma, “Large-Scale Data Visualization,” *IEEE Computer Graphics and Applications*, Vol. 21, No. 4, pp. 22-23, 2001.
- [16] K.-L. Ma, S. Parker, “Massively Parallel Software Rendering for Visualizing Large Scale Data Sets,” *IEEE Computer Graphics and Applications*, Vol. 21, No. 4, pp. 72-83, 2001.
- [17] B. H. McCormick, T. A. DeFanti and M. D. Brown, “Visualization in Scientific Computing,” *Computer Graphics*, Vol. 21, No. 6, November 1987.
- [18] H. G. Pagendarm and B. Walter, “A Prototype of a Cooperative Visualization Workplace for the Aerodynamicsyst,” *Computer Graphics Forum*, Vol. 12, No. 3, pp. 485-496, 1993.
- [19] J. D. Palmer and N. A. Fields, “Computer-Supported Cooperative Work,” *IEEE Computer*, Vol. 27, No. 5, pp. 15-17, 1994.
- [20] A. Pang, C. M. Wittenbrink, and T. Goodman, “Collaborative 3D Visualization with CSpray,” *IEEE Computer Graphics and Applications*, Vol. 17, No. 2, pp. 32-41, 1997.
- [21] W. Reinhard, J. Schweitzer, and G. Volksen, “CSCW Tools: Concepts and Architectures,” *IEEE Computer*, Vol. 27, No. 5, pp. 28-36, May 1994.
- [22] T. Rhyne, L. Martin, and D. Brutzman, “Internetworked Graphics and the Web,” *IEEE Computer*, Vol. 30, No. 8, pp. 99-101, 1997.
- [23] A. S. Rogers, “An Introduction to Groupware and CSCW,” *BT Technology Journal*, Vol. 12, No. 3, pp. 7-11, 1994.
- [24] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, Prentice Hall, New York, 1998.
- [25] A. Wierse, U. Lang and R. Ruhle, “Architectures of Distributed Visualization Systems and their Enhancements,” *Eurographics Workshop on Visualization in Scientific Computing*, pp. 125-137, 1993.
- [26] J. D. Wood, H. Wright, and K. W. Brodlie, “Collaborative Visualization,” In *Proceedings of IEEE Visualization '97*, pp. 253-260, IEEE Press, November 1997.