

# 跨平台虛擬場景瀏覽器之設計與實作

## A Design and Development of a Platform-Independent VRML Browser

沈宗正  
Tzong-Jenq Sheen

紀政宏  
Cheng-Hung Chi

王宗銘  
Chung-Ming Wang

中興大學資訊科學研究所  
圖學、多媒體暨虛擬實境實驗室  
Graphics, Multimedia and VR Laboratory  
Institute of Computer Science, National Chung Hsing University

### 摘要

我們使用 Java 程式語言，自行設計發展一套 VRML 瀏覽器系統，並將該系統命名為 JBVB (Java-Based VRML Browser)。另一方面，Dimension X 公司發展一套名為 Liquid Reality 的應用程式介面(API)，使得設計者可以利用此 API 自行設計發展應用程式，來瀏覽及操作以 VRML 2.0 語法所建構出的虛擬場景。為求縮短 JBVB 系統設計與實作的時程，故我們以此 API 為基礎，來發展 JBVB 系統。本論文敘述系統的設計製作過程並提出辦法解決所發現的缺失，而主要的貢獻點為設計並製作一個 VRML 瀏覽器的物件，再結合物件導向重覆使用及繼承的概念，讓人可以很快的擴充功能及設計自己所需的 VRML 瀏覽器，且達到跨平台的特性。

We use Java programming language to develop a VRML browser system, which is called "JBVB" (Java Based VRML Browser). On the other way, Dimension X Company has developed an API, which is called "Liquid Reality". With this API, programmers can develop their own applications to browse and manipulate the virtual scenes that are constructed by VRML 2.0. In order to short the design and development time of JBVB system, we choose this API as the foundation of JBVB system. This thesis describes the design and development process, and proposes solutions to solve the problems we have found. The main contribution of this thesis is to provide a VRML browser object. Combine with the concept of reuse and inheritance from OOP. Programmers can extend this object to develop their own VRML browser and make it cross platform in a very short time.

### 關鍵字

VRML、Virtual Reality、Java、World Wide Web。

### 一、簡介

電腦技術的蓬勃發展，使得人們朝向高計算能

力的技術推進。虛擬實境 (Virtual Reality) 的提出更帶來了人機介面革命性的改變，對電腦的操作將走向實體化，以人為出發點來設計三度空間的環境 [1,3,4,21,22]。如利用：三度空間滑鼠 (3D Mouse)、空間球 (Space Ball)、追蹤器 (Tracker)、資料手套 (Data Gloves)、資料衣 (Data Suit)、立體眼鏡 (Crystal Eyes) 或是頭戴顯示器 (Head-Mounted Display) 等，讓人更融入電腦所創造的虛擬世界，彼此交談來操作和使用電腦 [9,13,20,26]。

VRML 是虛擬實境描述模型語言的簡稱，它是一種用來描述多使用者互動模擬的語言 (a language for describing multi-user interactive simulations)。虛擬場景 (Virtual Scenes) 可經由此語言建構並經由網際網路 (Internet) 相互連結後，可在 WWW 中作超鏈結瀏覽。VRML 是一種類似 HTML (Hypertext Markup Language) 的概念，以純文字檔案格式來描述三度立體空間，藉由 HTTP (Hypertext Transfer Protocol) 協定在 WWW 相互傳遞，使得三度立體空間的虛擬世界得以共享。VRML 結合了虛擬實境和網際網路的概念，且似乎將成為在 WWW 上傳遞三度立體空間圖形新標準的趨勢。

以 VRML 設計出的虛擬世界須利用瀏覽器來瀏覽，雖然目前有許多的瀏覽器被研究開發出來，但是絕大多數只是支援 VRML 1.0 的語法，而且自家在不同的平台上發展一套瀏覽器，無法跨平台使用相同的程式，程式碼的重複使用機率很低，並且為了對瀏覽器有更進一步的了解，於是決定製作一套符合 VRML 2.0 的瀏覽器—JBVB。

### 二、背景知識

VRML 1.0 所建構出的虛擬世界完全是靜態的，本身物件並不具有任何自主性的行為，而且背景是完全漆黑的環境，人們可以在此環境中瀏覽及點取物件作資料鏈結。VRML 1.0 的檔案是 7-bit Ascii，無法支援非羅馬語系語言，如：中文、阿拉伯語、韓語、日語等，檔案開頭必須是『#VRML V1.0 ascii』，副檔名是『.wrl』或『.wrl.gz』，MIME 型式是『x-world/x-vrml』 [14]。VRML 2.0 所建構出的環境可以是動態的，物件本身可以具有自主性的行為，背景可

以利用貼圖而增加真實性，並增加行為設定、感應器的設定、3D 立體聲音及物體碰撞偵測等節點。VRML 2.0 的檔案是 16 bits UTF-8 編碼方式 ISO 10646 標準，可以支援非羅馬語系的語言，檔案開頭必須是『#VRML V2.0 utf8』，『#』表示註解，大小寫有區別，副檔名是『.wrl』，MIME 型式是『model/vrml』，不支援 URN's (Uniform Resource Name) 的格式。

VRML 以節點 (Node) 來表示三度空間圖形所定義的物件，以欄位 (Field) 描述節點內容，以事件 (Event) 當作節點訊息，以路徑 (Route) 記錄訊息走向，節點以階層方式所排列成的結構稱之為場景圖 (Scene Graphs)。VRML 2.0 依節點的特性大致上可分為六大類型：

(1) 群體節點 (Grouping node)：可以包含其他節點的節點，如：Anchor、Transform、Collision、Group 等。

(2) 感應節點 (Sensors)：感應滑鼠位置和使用作者互動，如：TimeSensor、PlaneSensor、TouchSensor、CylinderSensor 等。

(3) 造型節點 (Geometry)：建構物件的基本節點，如：Box、Cone、Text、ElevationGrid、IndexedFaceSet、IndexedLineSet 等。

(4) 外表節點 (Appearance)：改善物件表面顯示方式，如：Appearance、Material、TextureTransform、ImageTexture 等。

(5) 插值節點 (Interpolators)：設定 Key 值以線性內插法產生中間數值來代表物件的運動路徑，如：PositionInterpolator、ScalarInterpolator 等。

(6) 一般節點 (Common Nodes)：增加虛擬世界的真實性，如：AudioClip、Script、Sound、PointLight、SpotLight、WorldInfo 等 [8,15,24]。

VRML 是虛擬實境描述模型語言，著重在虛擬世界的建構，而無法設定物件完整的行為能力，VRML 2.0 只是擴充，並不能用以代表真實世界，因此必須要有描述語言來輔助描述其物件行為。WWW 的四通八達更顯示存在著各種危機，要確保自身的安全性問題，防止各種病毒入侵。其他語言如：Perl、C++ 等，也可以當作 VRML 的描述語言，選擇 Java 語言當作描述語言並非是唯一選擇，而是最佳選擇，因為 Java 是一套物件導向程式語言，針對網際網路程式設計所開發的，具有簡單性、安全性、架構中立性、強韌性、可移植性、多重線串 (Multithreading) 等多重優點，且也是在 Web 上加入動作及行為，所以較快被廣泛的接受。Dimension X Inc. 以純 Java 語言所發展的 Liquid Reality API，符合 VRML 2.0 規格，提供語法剖析 (parsing) 和成圖計算 (rendering)，也提供 VRML 2.0 節點的擴充功能，使得應用程式更具擴充性、可攜性和重複使用等特性。

VRML 1.0 瀏覽器依判斷資料的不同大致可分為三種：1. Stand Alone 如：WebSpace、WorldView 等，2. Helper Application 如：VR Scout、WebView 等，3.

Netscape plug-ins 如：Live3D、Vream。Stand Alone 的瀏覽器不須借助其他瀏覽器 (如：Netscape) 來抓取 VRML 檔案格式的檔案，而是自己來抓取且不經過 MIME 的設定。Helper Application 須借助其他的瀏覽器來抓取檔案，經由 MIME 的設定來選擇輔助應用程式來顯示，Netscape plug-ins 不須經由 MIME 設定直接動態執行。VRML 2.0 瀏覽器目前以 Stand Alone 和 Netscape plug-ins 為主，如：CosmoPlayer、Community Place、WorldView、Vream 是 Netscape plug-ins，Liquid Reality 可以成為 Stand Alone 和 Netscape plug-ins。

### 三、JBVB 系統之設計

基於 WWW 的盛行，而 Homepage 也成為各個學校、公司行號、各類機關的對外窗口，可以將各類欲傳達的訊息，一一在 Homepage 裡面作展示，而 Applet 指的是一種很小、附屬的、內嵌式的應用程式。所謂的內嵌式，指的是它可以在一個大系統中執行，在此以 Java 所寫作的 Applet 是內嵌在 Homepage 上，由網路瀏覽器來執行，Applet 可以獨立的執行自己程式而顯示結果給瀏覽者，或是跟瀏覽者作交談，將資料傳入程式而改變結果，若使用者端並沒有此 Applet 程式的程式碼，則遠端機器會將程式碼下傳至使用者端並執行起程式，為了加強 JBVB 程式碼的共享性，所以整個 JBVB 系統完全是一個 Applet [11,16,17]。

#### 3.1 Liquid Reality API 之架構圖

Liquid Reality API 發展平台是以 Java 為基底，讓程式設計者可由自己的 Java 應用程式來存取 3D 物件或是相互作交談的能力，主要可分成 5 部分 (圖 3-1)：

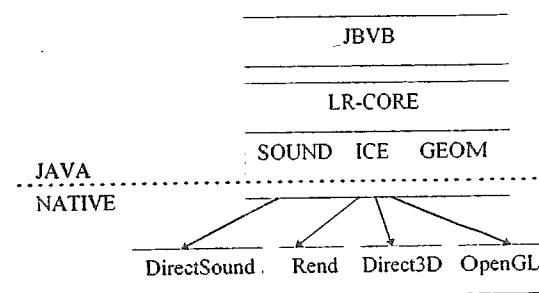


圖 3-1 Liquid Reality 階層圖

1. LR-CORE：此模組提供 Java Class 來表示 VRML 2.0 的節點、欄位、事件，一般來說，可以設計相當複雜和跨平台的 VRML 2.0 相容應用程式。
2. SOUND：此模組允許設計者來建立和產生 3D 立體聲音，在此模組的底層可以和不同的聲音程式庫相連接，而驅動不同的聲音硬體。
3. ICE：由 Java 語言所寫出來屬於低層的 3D API，獨立於 LR-CORE 完全不認識 VRML，可以直接設計和

VRML 不相干的 3D 應用程式，在底層和基本的成圖計算程式庫相連接，進行快速計算。

4. GEOM：數學程式庫和公用程式，提供 2D、3D 圖形基本的幾何物件，如：點、向量、多邊形、矩陣、透視法、邊界矩形(bounding box)等。

5. DirectSound、Rend、Direct3D、OpenGL：這些程式庫可以依其平台特性，用不同的語言來撰寫出最具效率的程式，是和機器相依的。SOUND 模組和 ICE 模組將不直接驅動硬體裝置[5]。

### 3.2 JBVB 設計目標

VRML 的瀏覽器不同於一般文字和圖形的瀏覽器，操作方式也略有不同，針對不同點我們訂定以下的設計目標：

#### 1. 相容性：

- 支援大部分的 VRML 2.0 的語法。
- 製作出類似 Stand Alone 的瀏覽器。
- 能夠使用 HTTP 連接 WWW 讀取檔案。

#### 2. 完整操作介面：

- 動態更換不同的 Camera 來觀看場景。
- 提供不同的瀏覽方式，如：Fly、Walk、Examiner。
- 提供不同的成圖計算，如：Wireframe、Point、Flatshaded、Smoothshaded [7,18,12]。
- 觀看來源檔案的內容及設定 Bookmark。
- 具 Forward、Back 和 Reload 場景的功能。

#### 3. 擴充性：

- 預視 VRML 檔案的場景，以利選擇。
- 一個場景以不同的成圖方式顯示。
- 程式碼跨越不同的平台使用。
- 達成物件導向 (Object-Oriented) 重複使用程式碼的特性。
- 提供讓人容易製作 Browser 基本功能的物件[6][17][18]。

### 3.3 JBVB 系統架構

為了區分文字瀏覽器和 VRML 瀏覽器的不同，我們將 JBVB 系統設計成類似於 Stand Alone 瀏覽器的操作介面，故可分為 5 個模組，如圖 3-2 所示：

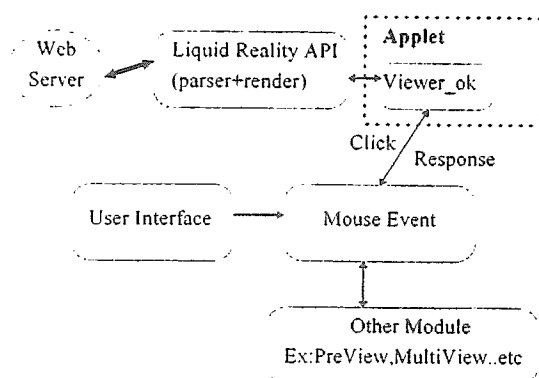


圖 3-2 JBVB 架構圖

1. Liquid Reality API：此模組係以 Java 為基底而開發出的，故內含有網路程式，自動從 Web 抓下程式和資料，並可進行 VRML 檔案的剖析而成具有階層狀的場景圖(Scene Graph)，再進行成圖計算。

2. Viewer\_ok：此模組係以 Liquid Reality API 為基底而再開發的 subclass，繼承 Applet Class，在此 class 內先讀取檔案剖析後，再將場景圖成圖計算後的場景顯示在所建立的 Scene Object，即可以提供瀏覽的黑色區域。此模組為系統的核心，並可重複的使用此模組，而製作成其他的應用程式，如：製作預覽 VRML 檔案的應用程式等。

3. User Interface：建立 GUI 的人機介面，製作可供下拉式的選單、可輸入文字的文字欄、對話盒和具圖形的按鈕，將這些功能自成一組時，可以任意的加入和刪除所需的介面，而不會影響 Viewer\_ok 模組。

4. Mouse Event：由 User Interface 接受使用者所作的事件，將事件送入此模組，再進行工作分配。

5. Other Module：此模組係為 JBVB 系統的縮影，如：預覽 VRML 檔案和顯示不同成圖方式等應用程式，亦是以 Viewer\_ok 模組當作核心，再擴充成導向不同的應用程式，顯示核心模組的重複使用性高。

### 3.4 JBVB 之實作與工作流程

JBVB 系統發展平台與環境，圖 3-3：

| 項 目                | 版 本               |
|--------------------|-------------------|
| Machine            | Personal Computer |
| Operation System   | Windows95         |
| Language           | Java JDK 1.0.2    |
| Liauid Reality API | 1.0 beta 11       |
| Complier           | Javac             |

圖 3-3 JBVB 發展環境

JBVB 系統在執行讀檔功能而更換場景時，所進行的工作流程，如圖 3-4。

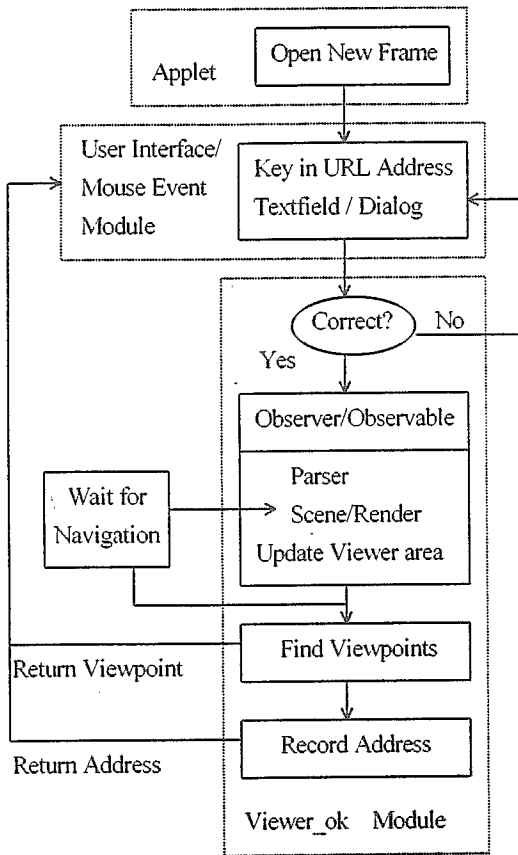


圖 3-4 JBVB 系統工作流程

系統的開始是一個 Applet，表示此系統可以內嵌在 Homepage 上，但是由於 Netscape 瀏覽器的安全性的考量，使得瀏覽器內建的類別提取器(Class Loader)無法提取此類別，故目前無法在 Netscape 瀏覽器內執行此系統。但在未來 Netscape 的版本中，將可以執行此系統。

目前我們可以利用 Java 內建的類別提取器 (appletviewer)來執行，經由操作介面輸入檔案位址，判斷位址的正確性與否後讀檔案。利用觀察者/可觀察者(Observer/Observable)的特性，將 Viewer\_ok 設定為觀察者，而觀察者若發現可觀察者有所變動時，會自動的執行 update() 函式，再進行成圖計算更換瀏覽區域的場景。每次重新讀入新檔時，必須尋找 Viewpoint 點和記錄位址，回傳給 User Interface 模組設定成目前場景的狀態，而場景物件 (Scene Object) 也是以觀察者/可觀察的特性，在 Wait for Navigation 的狀態下，等待使用者的瀏覽來抓取滑鼠位置會再更換場景 [27,28,29,30]。

#### 四、結果及分析

##### 4.1 結果

根據以上的方法，我們製作了一個 VRML 2.0 瀏覽器：JBVB，在個人電腦 (Personal Computer) Windows95 作業系統上的操作介面，如圖 4-1。

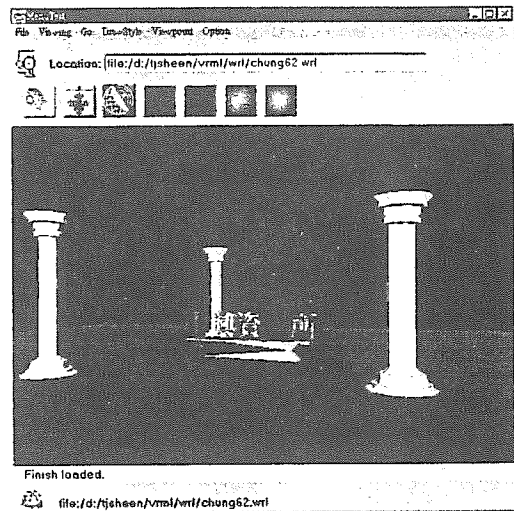


圖 4-1 JBVB 操作介面(PC)

JBVB 系統在 SGI Indigo Irix 5.3 作業系統上的操作介面，如圖 4-2。

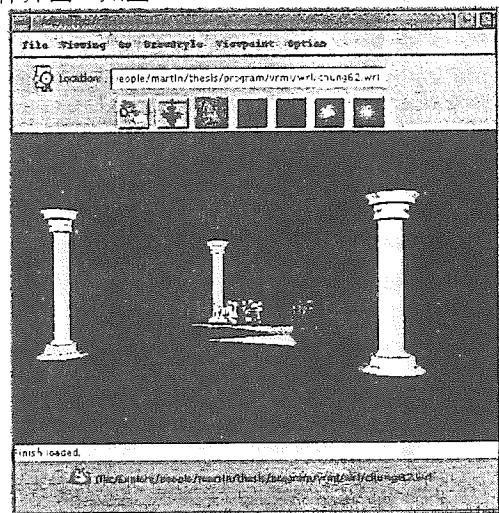


圖 4-2 JBVB 操作介面(SGI)

JBVB 系統在 SUN Sparc Ultra Solaris 2.5.1 作業系統上的操作介面，如圖 4-3。

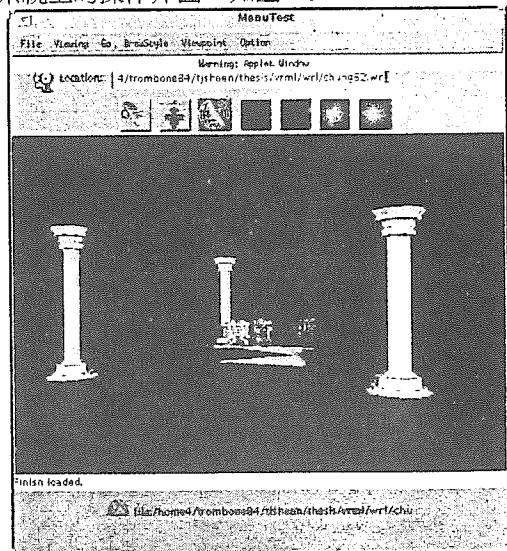


圖 4-3 JBVB 操作介面(SUN)

由圖 4-1、圖 4-2 和圖 4-3 之中可以看見，選單和按鈕的排列順序完全一樣，不同點在於 PC、SGI 和 SUN 工作站視窗標題樣子的不同，且在工作站上的按鈕以置中來區別兩者的不同。表示程式碼不須作任何改變，而可以跨平台的執行。

其主要的功能可分成 6 個選項：

1. File:

New Browser 可以開啓另一個新視窗，作另外一個場景的瀏覽。Open URL 可以抓取遠端的檔案，其格式為 "http://"，Open file 可以抓取當地的檔案，其格式為 "file:"，Preview File 可以呼叫另一個視窗，作預覽檔案的功能。

2. Viewing

Fly 以像是飛翔的方式來瀏覽場景，Walk 以像是人走路的方式來瀏覽場景，Examiner 是假想空間有個虛擬球，而對此虛擬球作旋轉時便同樣旋轉場景，View Source Code 可以觀看此場景檔的 VRML 檔案格式。

3. Go

Back 可以回到前一個場景，而 Forward 可以回到前一個場景，Reload 可以重新讀入此場景。

4. DrawStyle

Wireframe 以線框的成圖計算來顯示，Flat Shaded 以每個平面法向量上照明的密度值當作此平面的值，Smooth Shaded 將每個平面頂點法向量的照明密度值，先以兩兩配對的方式用內插法算出平面邊上的照明密度值，再以掃描線 (Scan Line) 的方式，算出平面內各點的值，Point 只顯示各平面的端點。

5. Viewpoint

Previous 可以回到此前一一個視點，Next 可以到下一個視點，Home 可以回到原來的視點。

6. Option

Help 指定獲得此瀏覽器進一步資料的網路位址，About This Browser 介紹此瀏覽器的開發者姓名和所屬單位。

4.2 分析

我們將計算 Liquid Reality API 內成圖計算程式庫的速度，分成停止瀏覽 (Non-Browsing) 和瀏覽中 (Browsing) 兩種狀態。程式每 500ms 抓取目前場景每秒切換畫面的速度，即每秒對此場景作幾次的成圖計算，對 10 次所得的數值作平均。我們將 VRML 的測試檔分作靜態場景 (Non-Interpolator) 和具有動態場景 (Interpolator) 兩種，而採用的幾何模型有：Box，Cylinder 和 Prims (box + cone + sphere) 三種，結果如圖 4-3：

|                  | Model    | Polygons | Non-Browsing     | Browsing         |
|------------------|----------|----------|------------------|------------------|
| Non-Interpolator | Box      | 12       | 36.468 frame/sec | 11.817 frame/sec |
|                  | Cylinder | 64       | 36.399 frame/sec | 11.762 frame/sec |
|                  | Prims    | 364      | 36.399 frame/sec | 10.115 frame/sec |
| Interpolator     | Box      | 12       | 12.227 frame/sec | 11.517 frame/sec |
|                  | Cylinder | 64       | 11.649 frame/sec | 11.263 frame/sec |
|                  | Prims    | 364      | 9.091 frame/sec  | 8.653 frame/sec  |

Polygon type : Triangle

圖 4-3 比較 Non-Interpolator/Interpolator

由以上的結果可以看出，在 Non-Interpolator 的狀態下作 Non-Browsing 時，每秒所更換的 frame 數量受 Polygon 個數的影響不大，原因是在第一次讀入檔案時，作第一次的成圖計算後便靜止不動，所以不更改 Camera 的位置便不作成圖計算，frame 個數便不致受 Polygon 個數的影響。

在 Browsing 的狀態下，可以明顯的看出每秒的 frame 個數因 Polygon 個數的增加而減少，所以 Polygon 的個數是會影響瀏覽器在瀏覽場景的效率。

在 Interpolator 的狀態下，不管是在 Non-Browsing 或是在 Browsing 的情況下，每秒的 frame 個數會因 Polygon 數的增加而減少，表示 Polygon 數的多寡是影響效率的主因之一。

而針對在 Non-Interpolator 和 Interpolator 中的 Non-Browsing 狀態下，可以明顯的看出 Interpolator 也是影響效率的另一個主因，因為瀏覽器必須不斷的作成圖計算才能作到物件動態 (Interpolator) 的效果，若是動態的效果作的愈多，效率會變得更差。

五、結論、未來工作與展望

在系統製作之初，十分的缺乏相關的書目及資料，原因是 VRML 2.0 的規格 1996 年 8 月 4 日才完全確定，Java 語言在近幾年才逐漸成熟，而兩者的結合更是一種新的突破，DimensionX Inc. 的 Liquid Reality API 目前只是在 Beta Version，且資料的收集必須完全藉由網路來達成，因此進度相當的緩慢。

依其設計目標，我們製作了一套符合 VRML 2.0 語法類似 Stand Alone 的瀏覽器，針對 VRML 瀏覽的不同特性，而具有改變瀏覽方式、改變成圖計算方式、動態改變 Camera 的位置而實作出一套 JBVB 系統，此系統可以透過 Homepage 而下傳至各個使用者端，達成程式和場景檔的共享性，也利用 Java 語言重要的跨平台特性，使得程式不用修改而得以在不同的平台上執行。而程式碼的重複使用性高，也從我們製作出的預覽 VRML 檔案及顯示不同成圖方式的應用程式中可以證實。而我們也提供了 Viewer\_ok 模組，讓使用者可以利用此模組很快的製作出可以觀看符

合 VRML 2.0 語法的場景，也具備 VRML 瀏覽器的基本特性[6,10,19,23]。

未來的工作將加強對 VRML 場景結構的剖析，依據場景圖 (Scene Graph) 找出隱藏在場景的各種資訊，而將之顯示在使用者操作介面上，如：超鏈結點和幾何形狀的關係、動態的在場景內加入或刪除節點、自己定義新的節點等，也可以對此系統修改成符合微軟公司的 IE (Internet Explorer) 版本，而內嵌在 Homepage 上。

#### 參考資料

- [1] A. K. Danowitz, Y. Nassef and S. E. Goodman, "Cyberspace across the sahara: computing in North Africa," *Communications of the ACM*, vol. 38, no. 12, pp. 23-28, 1995.
- [2] B. Fox, "Modeling Efficient VRML Worlds," Intervista, Invited speech Notes, World Movers: The VRML 2.0 Developer's Conference, 30th January 1997.
- [3] B. N. Meeks, "Electronic Frontier: The game plane to save cyberspace," *Communications of the ACM*, vol. 39, no. 5, pp. 24-26, 1996.
- [4] D. R. Pratt, M. Zyda and K. Kelleher, "Guest editor's introduction: virtual reality-in the mind of the beholder," *Computer*, vol. 28, no. 7, pp. 17-19, 1995.
- [5] Dimension X Inc, "Liquid Reality White Paper," Dimension X Inc, <http://www.dimensionx.com/products/lr/docs/whitepaper.html>, February 1997.
- [6] G. Taubin, "The IBM/Apple/ParaGraph VRML 2.0 Compressed Binary Format," IBM Research: <http://www.rs6000.ibm.com/vrml/binary/draft-3-bin.html>, 12th May 1996.
- [7] H. Baker, "Computer Graphics," Prentice-Hall International, 1994.
- [8] L. Ames, R. Nadeau and L. Moreland, "VRML 2.0 Sourcebook", Wiley, November 1996.
- [9] L. F. Hodges, R. Kooper, T. C. Meyer, B. O. Rothbaum, D. Opdyke, J. J. Graaff, J. S. Williford and M. M. North, "Virtual environments for treating the fear of the heights," *Computer*, vol. 28, no. 7, pp. 27-34, 1995.
- [10] L. Kitainik, "Building Efficient VRML Worlds," Invited Speech Notes, *World Movers: The VRML 2.0 Developer's Conference*, 30th January 1997.
- [11] L. Vanderburg, "Tricks of the Java programming gurus," Sams.net, 1996.
- [12] M. Green and S. Halliday, "Ageometric modeling and animation system for virtual reality," *Communications of the ACM*, vol. 39, no. 5, pp. 46-53, 1996.
- [13] M. F. Deering, "The holosketch VR sketching system," *Communications of the ACM*, vol. 39, no. 5, pp. 54-61, 1996.
- [14] M. Pesce, "VRML browsing & building cyberspace," New Riders Publishing, June 1995.
- [15] N. Matsuba and B. Roehl, "Special Edition Using VRML," QUE Publishing, 1996.
- [16] P. Chan and R. Lee, "The Java Class Libraries An Annotated Reference," Addison Wesley, September 1996.
- [17] P. Niemeyer and J. Peck, "Exploring Java," O'Reilly Publishing, 1996.
- [18] P. Stucki, J. Bresenham, and R. Earnshaw, "Computer Graphics in RP Technology," *IEEE CG&A*, vol. 16, no. 3, pp. 17-19, 1996.
- [19] R. Myers, "Optimizing VRML Content," Invited Speech Notes, *World Movers: The VRML 2.0 Developer's Conference*, 30 January 1997.
- [20] R. Stytz, "Distributed Virtual Environments," *IEEE CG&A*, vol. 16, no. 3, pp. 19-31, 1996.
- [21] S. Bayarri, M. Fernandez and M. Rerez, "Virtual reality for driving simulation," *Communications of the ACM*, vol. 39, no. 5, pp. 72-76, 1996.
- [22] S. Bryson, "Virtual reality in scientific visualization," *Communications of the ACM*, vol. 39, no. 5, pp. 62-71, 1996.
- [23] S. Chen, "Optimizing VRML," Invited Speech Notes, *World Movers: The VRML 2.0 Developer's Conference*, 30 January.
- [24] Silicon Graphics Inc, Sony and Mitra, "The Virtual Reality Modeling Language Specification Version 2.0. ISO/IEC WD 14772," VRML Architecture Group, 4th August 1996.
- [25] S. Diehl, "VRML++: Adding Classes to VRML," Technical Report, University des Saarlandes, January 1997.
- [26] S. Kalawsky, "The science of Virtual reality and virtual environments," Addison Wesley, 1993.
- [27] T. Poston and L. Serra, "Dextrous virtual work," *Communications of the ACM*, vol. 39, no. 5, pp. 37-45, 1996.
- [28] 莊榮宏, "虛擬實景發展環境與應用：子計畫二模型建構技術與在幾何設計的應用", 國立交通大學資訊工程系, 84年7月。
- [29] 陳錫賢, "虛擬實境在網際網路的應用發展一套VRML瀏覽器的研究", 國立中興大學資訊科學研究所碩士論文, 85年6月。
- [30] 陳錫賢, "虛擬實境在網際網路的應用發展一套VRML瀏覽器的研究", Rams'96 symposium, pp. 383-388, 1996年7月。