

以 XML 標準傳遞訊息的開放式多人虛擬環境系統

劉怡麟

國立政治大學 資訊科學系
台北市文山區指南路二段 64 號
g8901@cs.nccu.edu.tw

李蔡彥

國立政治大學 資訊科學系
台北市文山區指南路二段 64 號
li@nccu.edu.tw

摘要

多人虛擬環境系統結合虛擬實境與分散式系統，讓使用者藉由 3D 瀏覽器瀏覽虛擬場景，並控制虛擬人物(avatar)的動作，與他人產生互動。此類系統特色在於使用者可藉由瀏覽近乎真實的虛擬場景，在虛擬環境中達到娛樂、商業或學術上的目的。然而，目前虛擬環境系統在虛擬人物的動作多為罐裝動作，且在訊息傳輸上缺乏彈性。因此在這篇論文中，我們改寫了一個開放原始碼的多人虛擬環境系統(VNet)，將XML(eXtensible Markup Language)標準納入此系統的訊息傳輸協定中。希望藉由XML標準格式的彈性，能提供多人虛擬環境系統更具擴充性的訊息傳輸方式。我們以兩類功能為例，說明此傳輸格式的彈性。第一是使用XML格式中標籤與屬性的關係包裝虛擬人物動作的設定，達到同步或非同步的方式播放。第二是使用XML標籤的樹狀結構整合虛擬人物不同的罐裝動作，使動作能以串流或是組裝的方式播放。我們相信本研究的成果將有助於XML與多人虛擬環境系統結合的必然趨勢。

關鍵詞：多人虛擬環境系統、電腦動畫、VNet、XML、VRML。

一、概論

多人虛擬環境(Multi-User Virtual Environment, MUVE)系統經過許多年的研究與應用，目前已有許多成功且多元發展的例子。在應用方面，依據虛擬場景及輔助介面的設計，此類系統能給使用者一個教學、娛樂或商業的環境，讓使用者不必走出戶外也能在虛擬環境中完成真實世界裡的應用；例如瀏覽博物館的展出文物、與他人共同合作完成3D模型的設計等。在研究方面，多人虛擬環境的系統架構、彈性的訊息傳輸模式、使用者端的即時互動，及有效的動畫傳輸方式，一直都是多人虛擬環境系統研究的重點。配合網路技術和電腦軟體設備的長足進步，多人虛擬環境系統已經不再是具備高階電腦的使用者才可參與的；一般

使用者以桌上型個人電腦都即可如瀏覽一般網站般地參與多人虛擬環境，完成商業、娛樂或學術等方面的目的。

目前網路上有許多開放原始碼的多人虛擬環境系統(如VNet[11]、DeepMatrix[5])，及會員式的多人虛擬環境系統(如Blaxxun[2]、ActiveWorlds[1])。其中VNet提供一個完全由Java[8]語言實作的主從式系統架構，讓程式設計師可藉由修改其原始碼，以設計符合自己所需的訊息傳輸協定。在虛擬環境中為了盡量模擬真實世界的情況，例如人物有所謂的肢體語言、人與人之間有交談行為，所以使用虛擬人物與他人互動是此類系統發展過程需要具備的功能。使用者間的互動可藉由與他人文字方式的交談、互動行為(例如握手、擁抱)拉近使用者彼此之間的距離，讓使用者在視覺與文字中能更真切地感到此系統比一般平面網站的優點所在。

然而目前大多數多人虛擬環境系統在虛擬人物動作的處理上是屬於循序(sequential)播放。也就是使用者端在接收到播放動作的指令時，會馬上將指令反應到所指定的虛擬人物上播放。這樣的設計無法反應真實世界中一些簡單的互動行為，如握手、擁抱；也因而會讓使用者在瀏覽虛擬環境的過程中感到不夠真實。因此，我們希望系統除了要在架構設計上更具擴充性，以提供應用上多元發展的能力；同時對虛擬人物的動畫應有更多動態或同步化的設計，以符合現實世界的情況。

在這篇論文中，我們藉由改寫VNet多人虛擬環境系統，將目前在業界與學術界被廣為應用的XML[16]標準納入VNet系統的訊息傳輸中，提供更具擴充性的訊息傳輸模式。我們將這種訊息傳輸模式的好處實際應用在兩類虛擬人物的動畫呈現方式上。第一，VNet系統在虛擬人物動作方面不只有原本循序播放的方式，也能有與其他虛擬人物同步化的互動，例如：兩個人的握手動作要同步化播放才能達成；第二是整合既有虛擬人物的動作，從幾個原本一成不變的罐裝動作中組合出新的動作，並使用XML標籤的格式傳送給使用者播放。此兩類應用目的在使虛擬環境中的虛擬人物的

動畫播放方式更加彈性。

在第二節相關研究中，我們會介紹多人虛擬環境系統，其中包括分散式系統、虛擬實境和電腦動畫；第三節說明我們的系統與 VNet 系統架構上的差異；在第四節中，我們以實際應用說明使用 XML 標準來傳遞訊息的優點；第五節是將第四節中的應用說明實際應用到我們改寫過的 VNet 系統；最後是我們的結論與未來發展。

二、相關研究

多人虛擬環境系統是虛擬實境在分散式系統中的應用。伺服器端會負責管理虛擬場景，將場景中的變化傳送至使用者端；而使用者端則透過 3D 瀏覽器及輔助介面，讓使用者能見到場景中物件的改變，進而與其互動。虛擬實境本身的優點在於利用其 3D 模型配合聲音與動畫，讓瀏覽者能藉由 3D 場景的輔助，瞭解一些 2D 圖片或文字所不能清楚表達的現象(如，鳥類群聚行為)或物體(天體模型)[6]。

過去在多人虛擬環境系統有許多關於系統架構，訊息傳輸協定的研究[14][4][13][3]。其中系統架構的設計多著重在當使用者數量增加時所應具有的擴充性；例如運用資料過濾及 Dead Reckoning 的技術減少伺服器端與使用者端的資料傳輸量。而訊息傳遞方面的研究，則著重在能依照多媒體的特性作有效地資料傳輸。

VRML[12]標準的建立，對 3D 繪圖在網路上的應用有極大的影響。近期推出的多人虛擬環境系統，亦多以此建模格式做為前端 3D 瀏覽器的標準。然而，在使用 VRML 設計 3D 應用(如虛擬人物動作播放，支援互動的虛擬場景)時，由於 VRML 複雜的 PROTO 設計[9]、及外掛 VRML 瀏覽器不便利性等原因，造成了 3D 應用無法如預期的在網路上普遍應用。近幾年來，XML 此一在網路上被廣泛使用的標準，由於文件格式上的嚴謹定義，和其本身結構上與 VRML 的高相似度，造成 VRML 與 XML 自然地結合而產生出 X3D 標準[15]。利用 X3D 設計而成的 API 與瀏覽器，改變了過去 EAI 外部介面[7]的溝通模式，讓程式可對 VRML 虛擬場景或物件的內容作動態的操控。

三、系統架構

我們在這節中將進一步說明 VNet 系統的訊息傳遞協定，以了解其缺乏彈性的地方。我們將以 VNet 系統原本的架構與我們修改 VNet 系統後的架構做一新舊的比較，並將比較重點放在系統的訊息傳輸模式上。

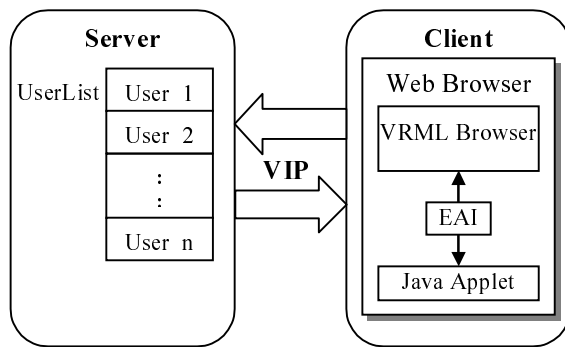
3.1 VNet 的系統架構

VNet 系統的架構如圖一所示。伺服器端負責管理虛擬場景中的物件與使用者間訊息的派送；使用者端有一個 Java 的 applet 程式透過 EAI(External Authoring Interface)與外掛在網頁瀏覽器的 VRML 瀏覽器互動，使用者藉由一開始的登入介面選擇自己在虛擬場景中的虛擬人物造型，並透過輔助介面控制虛擬人物動作與他人互動。伺服器端透過 VRML Interchange Protocol (VIP)[10]傳遞訊息給使用者，維持每一位使用者虛擬場景的一致性，並正確地接收使用者送出的對話訊息與動作。

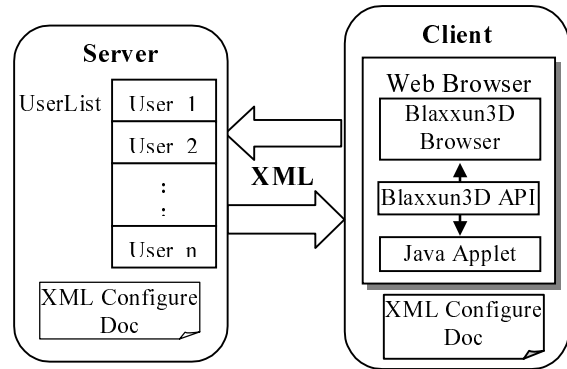
VNet 系統的訊息傳輸協定(VIP)中，訊息分為兩種：物件(虛擬人物)訊息與系統訊息，訊息的格式包含三個部分：物件、訊息種類與資料值，物件用一個 32-bit 的數字表示一個物件的 ID；訊息種類用一個 16-bit 的數字表示，並利用數字正負值分別表示物件與系統訊息；資料值的內容是根據訊息種類而決定，其資料型態是根據 VRML Node 中的 Field 資料型態(SFString、SFVec3f 等)而來。物件訊息的內容是關於虛擬人物的基本資料(位置、方向、大小與名字)，在基本資料之後的是虛擬人物的動作項目，因此訊息種類數會根據物件模型檔中動作數不同而變化；系統訊息的內容是有關 VNet 系統資料與使用者場景中的資料，其訊息內容的目的在保持伺服器端與使用者端物件(虛擬人物)數量的一致，例如加入或刪除物件。在伺服器端與使用者端可由表示訊息種類的值得知訊息內容中值的資料型態，進而對其做相對應的接收或傳遞動作。VIP 的訊息傳遞模式的缺點在於一個訊息只能傳送一種資料型態的資料，在未來發展系統新功能時會因為要傳遞不同資料型態的值而增加許多的訊息種類，對於系統發展會顯得缺乏彈性。

3.2 以 XML 標準傳遞訊息的 VNet 系統

為了讓 VNet 系統的訊息傳輸模式能以 XML 標準做訊息傳遞，我們需要修改 VNet 系統本身 VIP 的設計，讓原本一個訊息種類對應到一種資料型態的訊息傳輸模式變成能以 XML 標準傳輸標籤與屬性的結構。在使用者端，我們為了能完全操控虛擬人物的動作，且讓播放不再是循序播放的方式，我們將虛擬人物動作播放改以設定時間的方式讓動作在指定的時間播放。在 Java 程式與 VRML 瀏覽器的溝通介面部分，我們也採用了另一種不同於 EAI 的溝通介面。我們希望藉由結合 XML 與 VRML 的 X3D 而實作出的 VRML 瀏覽器，能讓 Java 程式對 VRML 的場景內容做完全控制。另外，為了讓我們的系統在未來能更有擴



圖一.VNet 原本的系統架構



圖二. 修改過後新的 VNet 架構

充性，我們使用 XML 文件做為系統的組態設定文件，藉由其標準格式提高系統未來在其它應用方面更高的相容度。基於上述幾項理由，我們修改了 VNet 系統如圖二所示。這些修改包括了 VNet 系統訊息傳輸協定、使用者端與虛擬環境溝通介面、及加入 XML 文件作為伺服器端與使用者端的組態設定。

訊息傳輸協定：為了伺服器端與使用者端間能傳輸多種資料型態，我們使用 Java API 中的 `ObjectOutputStream` 與 `ObjectInputStream` 傳送與讀取 `JDOM` API 中的 `Element` 物件，並使用 `JDOM` 的 API 包裝與剖析訊息。目前我們已將 VNet 系統的 VIP 完全修改為 XML 標籤的形式，每個標籤只有標籤名對應到 VIP 中的欄位名稱，不需像 VIP 要考慮欄位值與其正負值所表達的意思；標籤中使用 `id` 屬性表示送出訊息的使用者 ID，取代 VIP 中訊息的物件 ID。我們以一個 `ID=3` 的使用者送出位置變換至座標 (1, 2, 3) 給所有除了他自己以外的使用者的訊息為例子，說明 VIP 與 XML 標籤間的對應關係：

VIP : (3, 0, [1, 2, 3])

XML tag : `<POSITION id="3" x="1" y="2" z="3"/>`

使用者端介面：我們選擇 Blaxxun 公司以 Java 語言設計而成的 Blaxxun3D 瀏覽器，做為使用者端 VRML 的瀏覽器，並以其 API 取代 EAI 溝通介面，希望能藉由 Blaxxun3D API 的輔助對虛擬場景與虛擬人物動作做更好的控制。EAI 在設計上需要使用 VRML 的 PROTO 設計一個可讓 Java 程式溝通的介面，程式藉由存取 PROTO 中的欄位間接地對 VRML 檔案做動態的變化，進而影響到虛擬場景的變化。Blaxxun3D 的 API 可完全存取 VRML 中經過 DEF 定義的 Node，並提供多個節點同步更新的功能。對於將來要使用 Java 程式操控虛擬環境，並加入多樣化的動畫，將有很大的幫助。

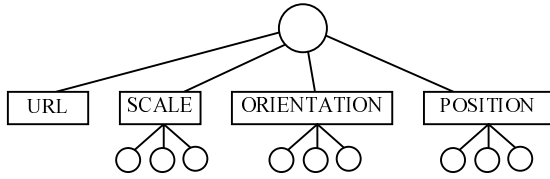
XML 組態設定文件：附錄 A 各別描述了我們系統中伺服器端與使用者端的組態設定文件。使用者端與伺服器端各自剖析 `ClientInfo.xml`

與 `ServerInfo.xml` 兩份 XML 文件，並從中得到所需的資訊。例如在使用者端，使用者會從 XML 文件中得到自己虛擬人物的 URL 與虛擬人物的基本位置、方向與大小，這些資訊成一個樹狀結構，如圖三所示。每一個新加入的使用者也會有一個這樣的虛擬人物基本資料。伺服器端為了維護虛擬場景的狀態，與 VNet 原本的架構一樣是以 List 的方式儲存使用者，不同的是每一位使用者的基本資料變成了以上述樹狀形式存在。

四、虛擬人物動作同步化與動態整合虛擬人物動作

目前在大多數的多人虛擬環境系統中，虛擬人物的動作大多沒有時間上必然的順序關係，使用者藉由輔助介面觸發一個動作給系統時，系統會馬上將訊息傳給其他使用者，使用者在接收到播放動作的要求時就會馬上將訊息中所指定的虛擬人物動作播放出來。這種方式對於一來一往非同步的互動(如揮手再見)是可以被使用者接受的；但對於要一起合作而成的動作(如握手)，這種需要同步進行的動作就無法在這種播放模式上完成。為了在播放虛擬人物動作上能有同步與非同步的選擇，進而增加使用者在虛擬環境中與他人的互動性，我們使用 XML 標準中標籤與屬性間的關係，包裝同步化的動作，並以實際應用的例子說明我們使用 XML 標準做傳遞訊息的好處。

另外，為了使虛擬人物的動作播放上更具彈性，我們設計一套虛擬人物的動作播放模式，讓虛擬人物的動作在一般罐裝動作之外能藉由動態整合虛擬人物既有的部分罐裝動作創造出新的虛擬人物動作，例如將揮手的動作整合走路的下半身動作，創造出一個一邊走路一邊揮手的動作。我們在設計原則上依然使用的是 XML 標準的標籤與屬性間的關係，再搭配上 XML 標籤中的樹狀結構，藉由標籤樹狀的



圖三 使用者基本資料的樹狀結構

關係將虛擬人物的罐裝動作做整合的設計。

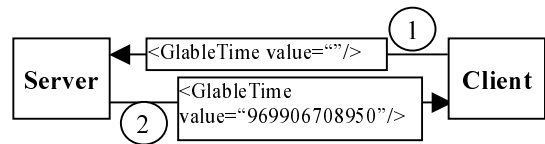
上述兩項應用在 VNet 原本的架構中是很難被實現的，除了要額外設計許多欄位與其相對應的資料型態，還要因為每個欄位表示的意義不同，將訊息分多次傳給使用者端。同理，使用者端在觸發一個動作時也會用相同的訊息協定傳輸給伺服器。這樣的設計對於接收的使用者端而言，要能夠完整的接收到所有相關的訊息才能正確的播放出一個虛擬人物動作，在設計上不僅會顯得不夠彈性，也會造成使用者端在接收或傳遞上程式設計的複雜度。

在 4.1 節中，我們會說明使用者如何使用我們的訊息傳輸模式獲得一個共同遵守的全域時間與如何使用全域時間播放虛擬人物的動作；在 4.2 節中，我們會說明在使用 XML 標準包裝訊息後，如何使用其標籤、屬性與標籤的樹狀結構整合虛擬人物的動作並增加動作的變化。

4.1 虛擬人物動作同步化

原本 VNet 系統的 VIP 由於沒有時間參數的設計，對於虛擬人物的播放是在接收到播放要求的訊息後，藉由觸發一個虛擬人物 VRML 檔案中對應到指定動作的 TimeSensor 做即時播放。因此我們使用 XML 格式將每個動作的要求都加入一個時間屬性。時間屬性的值為全域時間，每一位使用者將全域時間作為在播放動作時的時間參考點。

首先要說明的是使用者取得全域時間的方式。由於每一位使用者電腦的本地時間都不盡相同，所以我們以伺服器端電腦時間為全域時間，而在使用者端進行時間校正。傳遞動作訊息時如需加入時間屬性時，皆以全域時間為標準。但是由於將伺服器端電腦時間傳送給使用者的過程中，會有網路傳輸的延遲問題，每位使用者都會因為延遲時間(Latency)的不同，而得到一個與伺服器端電腦時間有誤差的全域時間。為了計算此延遲時間，我們在使用者端傳送一個訊息給伺服器端時，讓伺服器端在訊息中填上其電腦的時間並傳回給使用者(如圖四)。使用者端將訊息來回的時間除以二，便可計算得到粗略的單向傳遞延遲時間。我們在扣除此延遲時間後，便可計算全域時間與本地時間的差值。



圖四 使用者取得全域時間的步驟

藉由此差值可以很簡單地在全域時間與本地時間之間做轉換：當接收到一個包含全域時間屬性的動作時，只需將此全域時間加上此時間差便可得到本地時間。同理，使用者在傳送播放動作的訊息至其他使用者時，只要將本地時間減掉此時間差就能得到全域時間。

在指定虛擬人物動作播放的方式上，我們以 XML 標準在訊息中包裝播放動作的相關資訊，訊息中包括自己的 ID、與某位使用者同步化的要求、播放動作的時間以及播放的動作名稱。例如，使用者 ID=5 的使用者要在送出訊息的全域時間 10 秒後(969905354600，單位：ms)對 ID=3 的使用者做一個握手的動作同步化要求，訊息的模式就例如：`<BEHAVIOR id="5" name="ShakeHands" time="969905354600" user="3"/>`。以這種模式設計訊息的好處在於我們可以在標籤中設計許多屬性對應到動作的播放上，例如：當我們不想要虛擬人物播完整套動作，只需要播放整套動作的 20%~70%，就只需要在播放動作的標籤中加入 `start="0.2"` 與 `end="0.7"` 兩個屬性，告訴接收到此播放動作訊息的使用者只要播放動作的 20%~70%。

在處理虛擬人物的動作播放上，為了能更精確地控制不同物件的動畫播放，我們設計了一個 Timer 物件負責處理虛擬人物在虛擬場景中的動畫播放。此 Timer 物件的功能在於能根據每個動作訊息所指定的播放時間、動作的關聯物件(如手、腳)、播放次數等各項動作屬性，精確地完成每個動作的播放。藉由我們自行設計的 Timer 搭配上動作標籤中各項設定動作的屬性值，我們可以為虛擬人物的動作做更多的細部設定。且每個動作的播放方式不再是接收到動作訊息就要馬上播放的方式，而是使用者可以選擇在未來的某個時間點上播放一特定的動作或是跟某位使用者在相同時間做相同的動作以達到某些動作同步化的目的。在 4.2 節，我們會說明在一個播放動作的訊息中，此 Timer 如何處理訊息中的標籤樹狀結構並精確地將其播放。

4.2 動態整合虛擬人物動作

由於目前 VNet 系統中虛擬人物的動作都是罐裝動作，在動作的種類與數目是根據使用者在登入系統時所選擇的虛擬人物造型而變化，因而限制了使用者間虛擬人物的互動。為

了能動態整合虛擬人物的動作並播放，我們使用 XML 標準中標籤的樹狀結構與 4.1 節描述的指定動作變化的方式，整合虛擬人物既有的罐裝動作，創造出新的動作。

首先我們對動作標籤設定兩個基本的子標籤：`<position x="" y="" z="">`與`<orientation x="" y="" z="" angle="">`，目的在指定動作播放時的位置與方向，藉由這兩個標籤我們可以讓動作播放時有 teleport（移至指定地點）的功能。接著從虛擬人物的罐裝動作中取出一些組裝新動作所需的資訊，例如：由揮手的罐裝動作中取出揮手動作的上半身運動與走路罐裝動作的下半身動作。將兩個動作分別包裝成兩個動作的標籤，例如：`<BEHAVIOR name="Wave" affect="upper_body">`與`<BEHAVIOR name="Walk" affect="lower_body">`。最後將四個標籤群組為一個動作名為“WalkWave”的新動作，如下：

```
<COMPOUND name="WalkWave">
  <position x="" y="" z="">
  <orientation x="" y="" z="" angle="">
  <BEHAVIOR name="Wave" affect="hand">
  <BEHAVIOR name="Walk" affect="lower_body">
</COMPOUND>
```

※ COMPOUND 標籤指示接收到此訊息的使用者要根據其子標籤的內容組合一個名為 name 屬性值的新動作。

※ BEHAVIOR 標籤的 name 屬性值在指定罐裝動作的名稱，affect 屬性值在指定罐裝動作中的部分動作。

如果動作要加上播放時間、只播放某個動作的一部分片段或同步化，只要在 COMPOUND 或是 BEHAVIOR 標籤中加入 4.1 節說明的一些屬性 (time、user) 到動作標籤中，讓接收到訊息的使用者端的 Timer 能剖析這個樹狀結構，根據 COMPOUND 其子標籤下的各個 BEHAVIOR 標籤將一個虛擬人物的新動作所需的關聯物件組合起來。若標籤中有對動作做細部設定的屬性，亦將其屬性值加入播放動作時的設定，例如，揮手的動作只需播放 30%~80% (`start="0.3"、end="0.8"`)，走路的動作要連續播放五次 (`loop="5"`)。

對於動作標籤的屬性，我們可以設計關於動作細部設定的屬性 (如 start、end、time、loop 等) 與跟樹狀結構有關的屬性 (如 next)。樹狀結構的屬性主要是在設定一個動作與其它動作的關係，例如：`<BEHAVIOR user="2" name="Walk" next="Jump">` 中的 next 屬性表示在播放完 Walk 動作後要接著播放 Jump 動作。這兩個動作以樹狀的觀點來看就是兄弟的關係，而當一個動作標籤中沒有 next 屬性時就表示沒有下一個動作要播放。將此概念再擴大時，整個動作標籤的樹狀結構就有更多的變化，且此變化亦反應至虛擬人物的動作上。例如要在一個邊走路邊揮手的動作播放完畢後緊

接著一個跑步動作，我們不用傳送兩個動作訊息，而是只需要利用標籤的樹狀結構將兩個動作包裝成一個動作訊息，並在第一個動作中設定設定一個屬性值即可。對於一連串的动作而言，不需要修改虛擬人物模型檔中的動作，而只需就既有的動作中組合即可。因此，以樹狀結構包裝動作訊息，除了可達到動態地組合虛擬人物既有的動作，對於循序的播放方式也可藉由各項屬性完成。

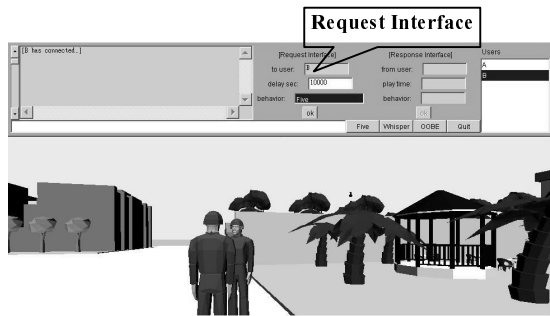
在 Timer 物件剖析一個樹狀標籤結構的過程中，會由上至下根據每個動作標籤 (BEHAVIOR) 的播放時間屬性 (time) 以及延遲播放時間屬性 (delay) 兩項屬性決定此動作的正確播放時間，對於有順序關係的動作，其播放時間是由最上層開始影響至最下層。在剖析的過程中，Timer 物件會將每個動作的資訊 (如 start 屬性值、end 屬性值、next 屬性值等) 包裝為動作物件並儲存在 Timer 物件的一個時間序列中。接下來 Timer 物件所做的便是使用 Blaxxun3D API 中的 beginUpdate() 與 endUpdate() 控制畫面的更新，在每次更新虛擬場景畫面時從時間序列中依序抓取出在此時間點中需更新的動作物件，並根據物件中對動作所做的各項細部設定的屬性值設定在新畫面中此動作所應呈現出的樣子。

五、實驗成果與分析

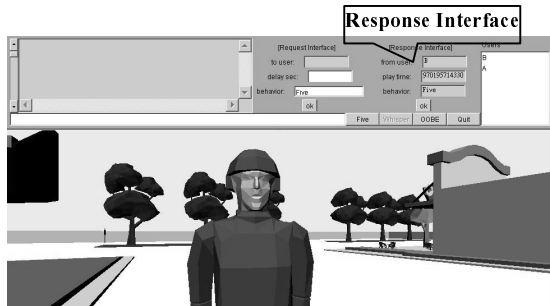
這一節將以我們修改過的 VNet 系統為實驗平台，為第四節描述的虛擬人物動作同步化與動態整合虛擬人物動作；設計實際應用的例子，說明我們實際將 XML 標準應用到多人虛擬環境中的訊息傳輸協定，能真正為訊息傳輸模式所帶來的彈性。

5.1 虛擬人物動作的同步化

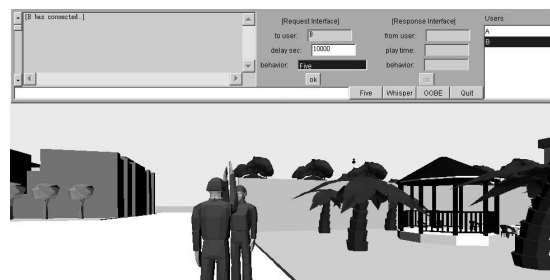
以“擊掌”動作為例說明動作同步化的過程與結果。我們第一步先根據 4.1 節描述取得全域時間的方法，並計算出使用者端與伺服端之間的時間差。接下來當使用者 A (使用者 ID=4) 在瀏覽虛擬場景時遇到他朋友 B (使用者 ID=8)，想跟他的虛擬人物做一個擊掌的動作時，就在使用者端的輔助介面的使用者清單中點選欲與其做同步化動作的使用者名字，以及考慮對方收到訊息以及反應的時間，自行設定要在全域時間幾秒後做一個動作的同步化，並在按下同步化要求介面中 ok 的按鈕後送出一個訊息至伺服端 (如圖五)，伺服端會將此訊息傳送到系統中的所有使用者。訊息的內容是：`<BEHAVIOR id="4" name="GiveMeFive" time="970169197560" user="8">`



圖五 使用者 A 對使用者 B 送出同步化的要求



圖六 使用者 B 收到使用者 A 送出的同步化要求



圖七 使用者 A 與使用者 B 在指定時間作出同步化的擊掌動作

使用者 ID=8 的使用者 B 會藉由輔助介面的一個提示訊息，得知有一位使用者 A 要與他的虛擬人物做一個擊掌動作的同步化(如圖六)，如果同意，就只要按下反應同步化動作的 ok 按鈕，就會送出一個 `<BEHAVIOR id="8" name="GiveMeFive" time="970169197560"/>` 的訊息，訊息中不設定 user 屬性值表示這個訊息不是一個同步化動作的要求。對於場景中其它的使用者而言，他們只是根據訊息中 time 的值與登入時計算出與全域時間的時間差，將動作播放的時間轉換到自己電腦的時間，並藉由 Blaxxun3D 的 API 設定播放動作的時間，其本身並不會知道播放動作的訊息之間有無同步化的過程。

若使用者 B 在使用者 A 同步化動作的要求時間內有做出回覆的動作，則會因兩者的虛擬人物有一樣的動作播放時間，而同步播放出動作(如圖七)。

5.2 動態整合虛擬人物動作

我們首先對虛擬人物動作設計一個 XML 的架構，例如：虛擬人物 Nancy 有四個動作：跑步(Run)、走路(Walk)、揮手(Wave)與跳躍(Jump)。

```
<Avatar name="Nancy">
  <Behavior name="Run">
    <Component name="upper_body">
    <Component name="lower_body">
  </Behavior>
  <Behavior name="Walk">
    <Component name="upper_body">
    <Component name="lower_body">
  </Behavior>
  <Behavior name="Wave">
    <Component name="hand">
  </Behavior>
  <Behavior name="Jump">
    <Component name="upper_body">
    <Component name="lower_body">
  </Behavior>
</Avatar>
```

在設計虛擬人物動作的 XML 架構時我們盡量避免一些動作整合後可能出現不合法的結果，例如：整合走路的下半身動作與跳躍的下半身動作會造成身體的分離結果。但此類動作的檢查，目前非本論文之研究重點，而尚須設計者以人工方式或工具程式完成。

我們在使用者端使用 JDOM 的 API 將有關於虛擬人物動作檔案的內容建立一個樹狀結構，以方便程式整合虛擬人物的動作。在使用者使用輔助介面自行設計出一個新動作時，動作會依據 Behavior 標籤中，Component 標籤的 name 屬性值判斷是否有重覆的 Component 標籤，若有重覆則表示這個整合出的動作是不合法的。因此，由上面 Nancy 的動作架構與整合規則，可以整合出一個合法的新動作 (RunWave)：

```
<COMPOUND name="RunWave">
  <BEHAVIOR name="Run" affect="lower_body"/>
  <BEHAVIOR name="Wave" affect="hand"/>
</COMPOUND>
```

如果要為動作加入播放時的方向與位置位置，可在 COMPOUND 標籤加入 position 與 orientation 兩個子標籤，讓動作播放時可以有 teleport 的功能藉以完成某些需與其他使用者一起完成的動作(如握手)，例如：

```
<COMPOUND name="RunWave">
  <position x="5" y="2" z="2"/>
  <orientation x="0" y="1" z="0" angle="3.14159"/>
  <BEHAVIOR name="Run" affect="lower_body"/>
  <BEHAVIOR name="Wave" affect="hand"/>
</COMPOUND>
```

另外，若動作要時間延遲播放或是只要播放動作的部分片段，可依 4.1 節中所敘述的方式，在動作中設定全域時間與起始動作百分比與結束時的百分比。

六、結論與未來發展

我們已藉由修改後的 VNet 系統實作兩個應用：同步化虛擬人物動作與動態整合虛擬人物動作，說明使用 XML 標準在訊息傳遞上的好處。標籤與屬性的搭配，讓系統的訊息不再被限定只能傳輸同一種資料型態，我們將資料型態轉為最基本的文字型態在伺服器端與使用者端間傳輸，能很有效地從標籤與屬性的內容設定使用者或伺服器端相關的資訊。此類系統的優點，可以從網際網路上 HTML 語言進化到 XML 語言的過程，看出其必然的發展趨勢。我們相信，採用 XML 標準將有助於多人虛擬環境系統的未來發展。

事實上，使用 XML 標準還有許多的優點；例如，未來使用名稱空間(NameSpace)可以讓系統功能有更多的變化，對於標籤中不同的名稱空間，系統可以做不一樣的應用發展。在 3D 協同建模中，藉由標籤與屬性的關係，以及我們在本論文中描述的同步化過程，都可在未來設計此類應用時提供一種具擴充性的訊息傳輸協定。

參考文獻

- [1] Active Worlds, URL:<<http://www.activeworlds.com>>
- [2] Blaxxun, URL:<<http://www.blaxxun.com>>.
- [3] C. Greenhalgh, "Implementing Multi-user Virtual Worlds: Ideologies and Issues," *Proc. of the Web3D-VRML 2000 Fifth Symp. on Virtual Reality Modeling Language*, February 20-24, CA, USA, 2000.
- [4] C. Bouras and T. Tsiatsos, "pLVE: Suitable Network Protocol Supporting Multi-User Virtual Environments in Education," *Intl. Conf. on Information and Communication Technologies for Education*, Vienna, Austria, Dec. 6-9, 2000.
- [5] Deepmatrix, URL:<<http://www.geometrek.com>>
- [6] D. W. Fellner and A. Hopp, "VR-LAB - A Distributed Multi-User Environment for Educational Purposes and Presentations," *Proc. of the Fourth Symp. on the Virtual Reality Modeling Language*, Feb. 23-26, Germany, pp.121-132, 1999.
- [7] EAIspec, URL:<<http://www.web3d.org/WorkingGroups/vrml-eai/>>
- [8] Java, URL:<<http://java.sun.com>>
- [9] S. Diehl and J. Keller, "VRML with Constraints,"

Proc. of the Web3D-VRML 2000 Fifth Symp. on Virtual Reality Modeling Language, 2000.

- [10] The VRML Interchange Protocol (VIP), URL:<http://ariadne.iz.net/~jeffs/vnet/VRML_Interchange_Protocol.html>.
- [11] Vnet, URL:<<http://ariadne.iz.net/~jeffs/vnet/>>
- [12] VRML97spec, URL:<<http://www.vrml.org>>.
- [13] W. Broll, "DWTP - An Internet Protocol For Shared Virtual Environment," *Proc. of the Third Symp. on Virtual Reality Modeling Language*, pp.49-56, 1998.
- [14] W. Broll, "Distributed Virtual Reality for Everyone - a Framework for Networked VR on the Internet," *Proc. of the IEEE Virtual Reality Annual Intl. Symp. (VRAIS '97)*, pp.121-128, NM, IEEE Computer Society Press, Mar. 1997.
- [15] X3D, URL:<<http://www.web3d.org/x3d/>>
- [16] XML, URL:<<http://www.xml.org>>

附錄 A

Server 端的 XML 組態設定文件：

```
<ServerInfo port="8888">
  <Avatar>
    <BasicFields>
      <POSITION type="SFVec3f" x="0.0" y="0.0"
z="10.0"/>
      <ORIENTATION type="SFRotation" x="0.0" y="1.0"
z="0.0" angle="0.0"/>
      <SCALE type="SFVec3f" x="1.0" y="1.0" z="1.0"/>
    </BasicFields>
  </Avatar>
</ServerInfo>
```

Client 端的 XML 組態設定文件：

```
<ClientInfo port="8888" isConnect="false"
BGCOLOR="A0A0FF" FGColor="000000">
  <AvatarList>
    <BasicFields>
      <POSITION type="SFVec3f" x="0.0" y="0.0" z="10.0"
field="translation"/>
      <ORIENTATION type="SFRotation" x="0.0" y="1.0"
z="0.0" angle="0.0" field="rotation"/>
      <SCALE type="SFVec3f" x="1.0" y="1.0" z="1.0"
field="scale"/>
    </BasicFields>
    <Avatar>
      <name>The Head</name>
      <URL>kiwano.wrl</URL>
      <behavior>Laugh</behavior>
    </Avatar>
    <Avatar>
      <name>Armyman</name>
      <URL>armyman.wrl</URL>
      <behavior>Five</behavior>
    </Avatar>
  </AvatarList>
</ClientInfo>
```