

以 Web 2.0 觀念實現可客製化之語意虛擬環境 Realizing Customizable Semantic Virtual Environment with the Concept of Web 2.0

朱鈺琳
國立政治大學資訊科學系
g9505@cs.nccu.edu.tw

李蔡彥
國立政治大學資訊科學系
li@cs.nccu.edu.tw

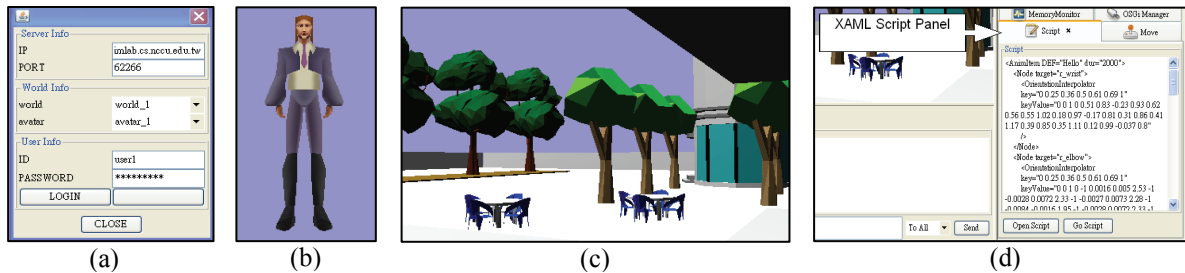


圖 1 多人虛擬環境系統功能概觀

摘要

在多人虛擬環境中，如何透過 Web 2.0 的觀念，讓使用者可以也是設計者，即時和其他使用者分享自己所設計的動畫元件，是 3D 數位內容可否普及的重要關鍵之一。透過語意的描述將有助於動畫元件理解虛擬物件的含意，進而達到元件間的溝通並產生客製化的化身行為。本論文中實現語意虛擬環境的過程可分為三部分：動態載入動畫元件的機制、設計虛擬物件和化身的語意描述、及制訂元件間訊息傳遞的通訊協定。我們以自行開發的 IMNET 做為多人虛擬環境系統的實驗平台，加入動態載入動畫元件的機制，並利用虛擬世界的 Ontology 來建立及存取物件的語意資訊，並以化身的互動元件來說明此語意虛擬環境的運作方式。

關鍵詞：多人虛擬環境、動畫腳本語言、OSGi 架構、Ontology、語意虛擬環境。

1. 前言

近年來多人虛擬環境的應用有逐漸增加的趨勢，不論在學術上或商業上都有可供多人上線的虛擬環境服務，在功能上也不斷地推陳出新。多人虛擬環境(Multi-user Virtual Environment, 簡稱 MUVE)是一個可供多人同時上線，透過文字或 3D 虛擬人物達到互動效果的虛擬世界。如圖 1(a)~(d)為一般多人虛擬環境的功能概觀：利用系統的登入介面登入、選取自己的人物模型、瀏覽 3D 虛擬世界和選擇性的使用腳本語言和虛擬世界中的人物作互動。部分較為人所熟知的多人虛擬環境，在這幾年網路速度的提升和電腦硬體設備的更新之下，線上

參與的人數從幾千人上升到幾百萬人，成長迅速使得各式應用也逐漸發展開來[10]。

因此，在多人虛擬環境中，如何透過虛擬世界中的化身來展現每位個體的獨特性，是人們在遊戲中能否獲得更多成就感和趣味的重要因素。化身(Avatar)意指可供使用者操作，並在虛擬世界裡活動的替身。以多人虛擬環境使用者的角度來看，化身能代替真實世界的使用者，與整個虛擬世界或其他使用者產生互動。因此，不少虛擬環境的使用者相當注重自己所操作的化身，如化身的衣著、飾品、甚至是行為等等。

另外，在虛擬環境和使用者的互動上，大部分虛擬環境系統並不只是單純的顯示 3D 物件，而是提供一些使用者和環境進行互動的方式，如在具有故事劇情的虛擬世界中，使用者到達某個特定地點後，便會觸發某個情節的產生。然而，傳統上這些虛擬環境中的情境資訊，都是以 3D 繪圖及動畫的方式呈現，而使用者也只能透過系統所設定的固定腳本與環境或其他使用者互動。如何設計一個有延展性的虛擬環境系統，讓使用者也能撰寫客製化的動畫程式，動態取得環境及其他使用者的資訊，以做進一步的互動設計，是一個極具挑戰性的問題。

IMNET 是一個以 XML 為基礎並具延展性的主從式虛擬環境系統[8]，透過 XAML(eXtensible Animation Modeling Language)語言的設計讓使用者可以整合不同層次的動畫指令於同一腳本中，以操作虛擬環境中的化身[7]。低階的動畫指令包含化身關節的控制，使用者可以產生較細微的化身動作，但缺點是低階指令內容龐大。因此，在 IMNET 中可以選擇使用高階的動畫指令來避免傳送低階的動畫指令。但這些高階指令是系統中預先設定好

的，因此無法達到使用者的客製化。

在我們先前的研究中，主要就是著重於如何讓化身擁有客製化的行為[3]。我們改以動畫元件的方式，讓使用者能夠設計自己的動畫元件，並即時安裝到虛擬環境中與其他使用者共享。此動態安裝的機制可刺激使用者創作，並在 3D 環境中分享這些創作內容，如一個可協助化身作路徑規劃的動畫元件，可透過簡單 XML 腳本的指定，安裝到虛擬環境中，並為化身規劃出一條不會碰撞到周遭障礙物的路徑。在一般的虛擬環境中，除幾何資訊外，系統對 3D 物件模型所提供的額外資訊十分有限，因此如何讓動畫元件的設計者皆能依循標準的物件資訊描述及傳遞方式運作，是虛擬環境系統需要進一步提供的功能。

綜合以上所述，本論文以 Ontology 來作為在動畫軟體元件在設計上所能依循的虛擬世界描述方式，程式可透過語意的描述取得物件的資訊，如物體的名稱、幾何、高度和 2D 近似值等資訊。在化身的語意上，我們也設計了一套 Ontology 來描述化身所包含的資訊。和一般物體不同的地方在於化身的變化性較大，如位置和狀態等資訊。另外，我們也在 IMNET 的通訊協定中新增了詢問訊息的機制，讓元件在互動時可以藉由詢問的方式，取得對方客戶端的化身資訊。在本論文的最後也提供兩個範例來說明動畫元件如何藉由世界語意資訊的分享，取得虛擬世界的資訊，並經由這些資訊產生化身進一步的行為。

在接下來的章節中，我們會先介紹語意虛擬環境的相關研究；第三節我們會介紹我們所設計的 Ontology；第四節我們會介紹 IMNET 在訊息傳遞上的改良；第五節我們會提供一些情境範例，分別由化身和環境間的互動以及化身間的互動來說明如何使用語意來達到動畫元件的需求；最後，我們將作一些結論和說明未來可能的研究方向。

2. 相關研究

在語意虛擬環境的概念產生前，有許多虛擬環境的相關研究著眼於如何結合虛擬環境系統和 AI/ALife 這兩種技術，而智慧型虛擬環境系統 (Intelligent Virtual Reality Systems, IVRS) 即是針對結合此兩種技術後的統稱。R. Aylett 等人在觀察這些擁有 AI 技術的虛擬環境後，發現這些系統大多包含一些特性；例如，系統會加入一個可解決組態 (Configuration)、排程 (Scheduling) 或互動 (Interactive) 等問題的元件；系統會建造一個知識的層級 (Knowledge Level) 來表達場景的概念，同時也支援更高階處理場景的方式，或是提供自然語言處理系統的介面；系統能描述一些因果反應 (Causal Behavior)，以替代物理特性的模擬；系統能夠辨識使用者所產生的互動動作，然後決定系統該採取的適應行為[2]。

基於程式間能自動透過彼此的資訊來作溝通之理由，語意虛擬環境的研究提出以豐富的語意來描述虛擬環境。傳統虛擬環境系統在內容的表現和互動上大多是以人類為設計的焦點，而不是提供給電腦程式來做分析處理[9]。一般的虛擬環境利用場景樹 (Scene tree) 來存 3D 空間中的幾何資訊，然後經由 3D 瀏覽器顯示到使用者面前。人類可以輕易理解場景樹經過算圖後所呈現的圖像結果，但是對於電腦程式來說只能知道場景樹的幾何及繪圖屬性，而不會知道最後所表示的物體是什麼。而語意虛擬環境使用一個抽象的觀點來看待存在的虛擬環境，並且利用語意網 (Semantic Web) 的技術來描述世界模型和網路通訊模型的語意。使用語意網的目的在於讓機器也能夠了解在虛擬環境中，場景所代表的含意。面對可能多變且複雜的虛擬環境，語意虛擬環境提供一個在虛擬環境上開發新元件的方式，使得這些新元件可以在具有語意的虛擬環境下重複使用，突破不同虛擬環境間的侷限。

K. A. Otto 所開發的 SEVEN 是一個可利用軟體元件來開發語意虛擬環境代理人的系統，發揮軟體的可重複使用的特性，使其能快速的將開發好的軟體套用在不同的虛擬環境上。另外，也有許多相關的研究著重於建構虛擬環境中的模型 (Shape) 時，希望能藉由語意的描述，讓 3D 物件具備知識的條件。T. Abacı 等人設計了一個與動畫需求有關的語意表達方式，使虛擬環境系統中的智慧型物件 (Smart Object) 不只可以提供幾何上的資訊，同時也能提供相關的動畫資訊[1]。M. Gutiérrez et al. 提出了一個以語意為基礎的方法，來描述組成虛擬人物的多型態資料，以符合一般性的人體資訊需求，並且為在虛擬人物上建置 Ontology 提供了一個實作範例[5]。在這篇論文中，M. Gutiérrez 也提到建構 Ontology 是一個持續性的過程。目前在虛擬人物上建構的 Ontology 描述，包含了模型的幾何資訊、骨架的結構、虛擬人物個人的屬性、動畫產生方式等資訊，以使動畫中的物件有一個彼此互動的依循。可見使用語意網的技術來描述虛擬環境不只可以達到動畫元件和虛擬環境互動的目的，同時也是將來與其他虛擬環境作整合或延伸的一個方式。

在人體動畫的語意上，藉由一些化身關節參數的指定，可產生各種不同的人體動畫，並具有重複使用的特色。A. Garcia-Rojas 等人以情緒來做為人體動畫查詢和分類的基礎，並在人體動畫上加入了情緒的資訊和人體表達參數的屬性[4]。他們首先指出如何在情緒上做分類，然後訂出幾個情緒表達程度的屬性，如流暢性、強度、空間延展性、速度和重複性等等。在情緒的表達上，A. Garcia-Rojas 等人所使用的情緒模型是 Whissel's wheel activation-evaluation。因此每個人體動畫都可對應到某個情緒，並連結到相關人體關節的人體動畫參數 (Body Animation Parameters, BAP)，例如當使用者需要可表達快樂 (joy) 的動畫時，便可透過 Ontology 查詢的方式來找出符合的人體動畫。A. Garcia-Rojas

等人的設計主要是結合情緒模型和虛擬人物架構，如此一來便可以情緒為關鍵字來搜尋具備此情緒表達方式的虛擬人物模組。而本論文則是以化身互動相關的語意描述為目標來設計世界和化身的 Ontology，在接下來的章節會說明我們的設計。

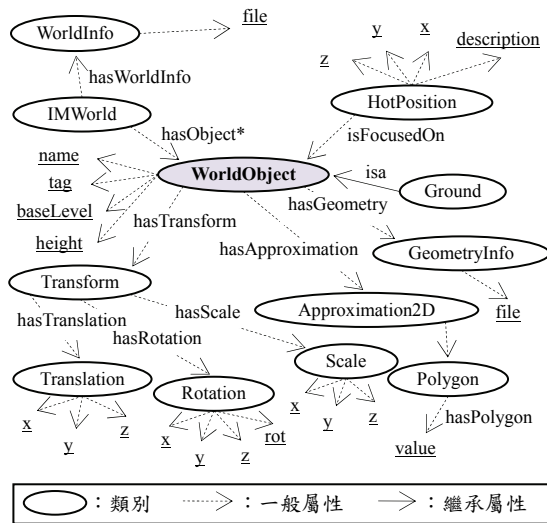


圖 2 虛擬世界 Ontology 設計藍圖

3. 虛擬環境中 Ontology 設計

語意虛擬環境的實現將有助於虛擬世界中的元件彼此溝通，讓化身能夠取得虛擬世界中的資訊，甚至取得其他化身的資訊。如此一來便能在虛擬環境系統上開發可自動化的元件處理程序。接下來我們會先說明在 IMNET 系統中設計 Ontology 的動機，以及我們所設計的 Ontology 類別。在第二部份會說明虛擬化身的 Ontology 設計。最後，我們將說明如何利用所設計的 Ontology 來載入虛擬世界中的模型。

3.1 IMNET 多人虛擬環境之 Ontology 設計

語意虛擬環境可讓動畫元件了解虛擬世界中的物件含意，並透過化身間的語意描述，取得和對方互動的方式。在語意設計的目的上，我們希望能讓動畫元件可以取得虛擬世界的資訊，並透過這些資訊來決定接下來的行為和動作。因此作法是希望能夠讓動畫元件透過世界或化身的 Ontology 描述來產生與環境互動或不同化身間的溝通行為。而在設計的需求上，我們希望能提供一個較為廣泛的 Ontology 設計，而不只限於一個特定的應用，也就是可供延展的空間。

在目前的 IMNET 多人虛擬環境系統下，環境資訊的語意設計上大概有幾點考量。第一個是能夠提供 3D 物件的幾何資訊；第二個是提供 3D 物件的轉置(Transform)資訊；第三是能提供 3D 物件的 2D 近似值。另外，為了讓 Ontology 設計上保留彈性，我們也考慮加入在不同應用中都可使用的標籤註

記(Tag Annotation)的設計，如在某特定應用中，註記 'sidewalk' 和 'crosswalk' 標籤的虛擬物件，使此特定應用之元件理解此註記之虛擬物件具有人行道或斑馬線等特性。

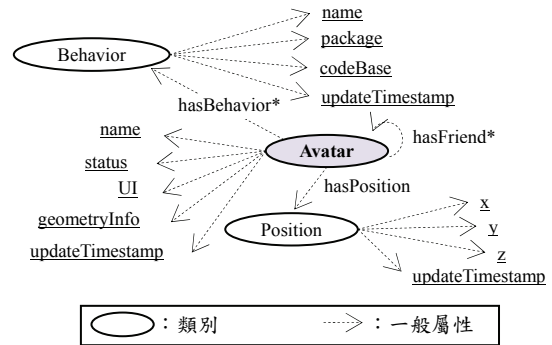


圖 3 化身 Ontology 設計

如圖 2，IMWorld 是整份世界文件中的根節點，描述了世界的一些資訊，以及世界中所有物件的資料。WorldInfo 中主要包含這個世界的背景和光線等資訊。IMWorld 則是經由 hasWorldInfo 來連結到特定的 WorldInfo。

WorldObject 可用來描述世界中的某個物件，但事實上不太容易完全考慮到物件所有可能的屬性，因此在制訂的時候，我們主要的考量是以本論文範例應用中使用到的屬性。例如大部分的應用會希望能知道物件如何稱呼，讓程式能夠依名稱來分類物件或對物件進行相對應的互動，所以制訂了名稱屬性 (name)，如桌子、建築物或是樹木等。另外，在某些應用中可能需要取得物件的高度 (height) 和最低點的高度 (base level) 等屬性，如欄杆的柱子是立在離水平面 1 公尺的地方，然後欄杆的高度是 1.5 公尺等；所以，當虛擬人物要規劃一個由欄杆下穿過的動作時，程式就可以透過這些資訊來判斷虛擬人物是要從下面鑽過還是從上面跨過去。

Ground 主要是繼承於 WorldObject，最大的不同在於表示此世界的物件具有地板的特性。在一般的應用中可以透過 Ground 來設定一個虛擬世界的範圍，如設定一個扁平的物體為 Ground 後，當動畫元件有需要知道世界的範圍時，就會去尋找世界中的 Ground 物件，並以 Ground 物件的範圍當成世界的範圍。

GeometryInfo 是用來描述 WorldObject 的幾何資訊，如 VRML 檔案的連結。一般 3D 的虛擬環境都是以讀取幾何資訊為主，其他的語意資訊為輔。不過，這裡所提供的方式是以讀取語意資訊為主，並藉由外部連結的方式，連結到虛擬物體所屬的幾何檔案。因此，透過 GeometryInfo 可以取得外部物體幾何檔案的位置，並讀取該位置的幾何模型到虛擬環境中。

HotPosition 主要是來自於 Point of Interest (POI)

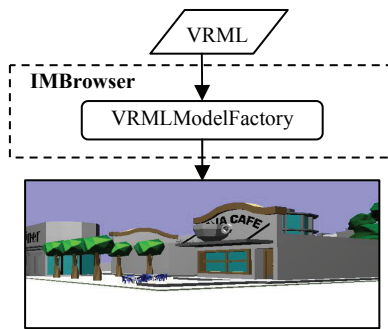


圖 4 直接處理 VRML 檔案以產生虛擬世界

的概念[6]，可用來描述一些虛擬世界裡的有趣景點。在 HotPosition 中包含了景點的座標資訊和景點描述。另外，動畫元件也可能需要知道這個景點所聚焦(Focus)的物體是甚麼，以幫助虛擬人物切換到正確的視角。所以，每個特定的 HotPosition 都會經由 isFocusOn 的屬性連結到某個 WorldObject 物件上。

除了紀錄世界中物體的名稱等基本屬性外，在語意的描述中也應該包含物體轉置的資訊。因此，每個 WorldObject 都可以藉由 hasTransform 連結到一個 Transform 物件，再由 Transform 分別由 hasTranslate、hasRotation 和 hasScale 等屬性連結到 Translate、Rotation 和 Scale 物件。

Approximation2D 主要是用來描述 3D 虛擬環境中物體的 2D 近似值，並以 Polygon 的方式來描述 2D 近似值所涵蓋的範圍。使用 Approximation2D 來描述物體的 2D 近似值最大的好處，就是可以減少程式親自去算 2D 近似值時，所耗費的時間，例如在路徑規劃的例子中，規劃器元件需要取得世界中所有物體的 2D 近似值，以產生世界的 2D 地圖；如果規劃器元件可以取得 Approximation2D 所提供的 2D 近似值描述，即可省去計算每個物體 2D 近似值所花的時間。

3.2 化身之 Ontology 設計

在多人虛擬環境中，化身的操作者可以是真實使用者(Real User)，也就是一般常說的虛擬環境使用者；另外，化身的操作者也可以是模擬真人行為的電腦程式，也就是所謂的虛擬使用者(Virtual User)。虛擬使用者能模擬真實使用者，來處理一些簡單的工作，如在虛擬世界的互動應用中，虛擬使用者所控制的化身可用來做解說或導覽的工作。

因此，如圖 3 所示，為了化身間互動的應用，我們設計一些化身基本的 Ontology 類別和屬性，如 hasBehavior 屬性等。在行為 (Behavior) 的類別中，包含了 name 和動畫元件的 codeBase 等資訊，可供化身在互動時客製化自己對特定行為的動畫。除了行為外，化身也包含了一些基本資訊，如 name、geometryInfo、status 和 UI 等。另外，透過

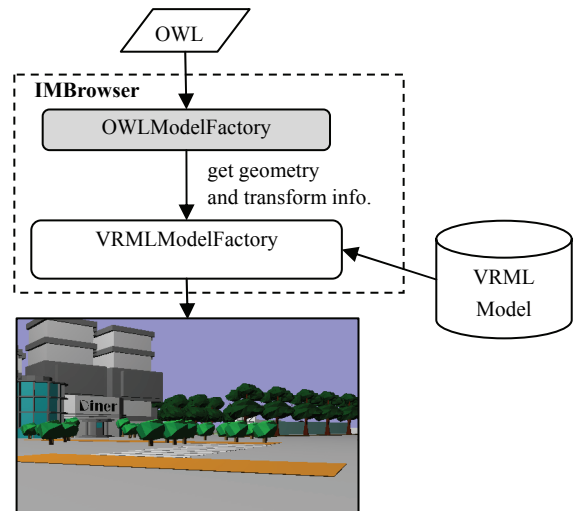


圖 5 透過 OWL 文件來產生虛擬世界

化身的 hasFriend 和 hasPosition 屬性，我們可以分別取得化身的的朋友資訊和化身目前的位置資訊。

3.3 利用 Ontology 檔案讀取世界模型

在前面兩小節裡，我們制訂了一個可描述虛擬世界的 Ontology，系統可透過這份 Ontology 取得虛擬物體的幾何資訊。我們使用 W3C 所制訂的標準 Web Ontology Language (OWL) 做為虛擬世界 Ontology 的檔案格式，並且透過自動產生的 Java 類別和 Protégé API，將一份 OWL 文件以物件的方式來存取它。

在一開始，系統會先將 OWL 文件載入成 OWLModel，並且取得每個 WorldObject 物件實體的 GeometryInfo。然後會以每個 GeometryInfo 的 file 屬性為路徑，將路徑所在的 VRML 檔讀取到虛擬世界中。同時，系統會去讀取 WorldObject 的 Transform 屬性，包含位移、旋轉和縮放屬性，然後套用此 Transform 到世界的模型上。因此，本論文在這邊延伸原先 IMBrowser[7] 的功能，讓 IMBrowser 可同時處理一般的 VRML 檔和 OWL 文件。圖 4 為處理單一 VRML 世界檔案的流程。原先的世界是以單一模型的形式存在，因此我們可直接用 VRMLModelFactory 將世界的模型檔轉換成 3D 場景。圖 5 則為透過 OWL 文件來產生虛擬世界的流程。虛擬世界中每個物件的資訊都被保存在這份 OWL 文件中，因此透過 OWLModelFactory 可以取得虛擬物件的幾何資訊，如外部 VRML 檔案的位置描述。然後再由 VRMLModelFactory 透過虛擬物件的幾何資訊，載入外部的 VRML 檔案而產生虛擬世界的場景。

4. IMNET 訊息傳遞機制的改良

IMNET 的通訊協定中包含了訊息傳送的内容

和訊息封裝方式兩個部分，而訊息傳送的内容為 XML 格式的標籤；如圖 6 為 IMNET 訊息的範例，user1 想要送出一段 <Chat> 的訊息給 user2，因此 <IMNET> 標籤中記錄了訊息的傳送方和接收方，然後交由伺服器將這段訊息傳送給 user2。

```
<IMNET from="user1" to="user2">
  <Chat>...</Chat>
</IMNET>
```

圖 6 IMNET 訊息範例

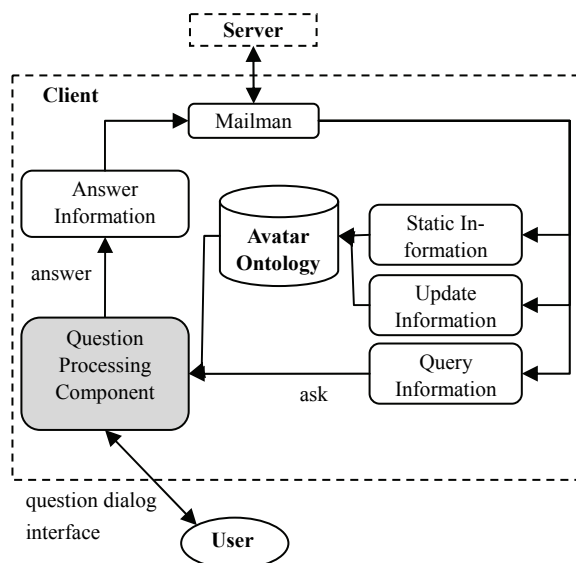


圖 7 客戶端處理三種類型資訊架構

在原先的 IMNET 中，化身間的互動功能還沒有被加入到系統中，因此 IMNET 在通訊協定的設計上，並未包含我們所需要的訊息傳遞機制，如訊息的詢問。在本節中我們將改良 IMNET 的訊息傳遞機制，以合乎我們化身間互動的需求。如圖 7，在化身間互動時的訊息傳遞上，我們區分為三種不同類型的資訊交流。第一種為靜態資訊(Static Information)，如帳號、暱稱等登入後不太會變更的資訊；第二種為更新資訊(Update Information)，如化身所在位置，為需要時常更新的資訊；第三種為詢問資訊(Query Information)，如朋友名單等，可能在特定需求下才需要知道。另外，此三種資訊類型代表著不同的更新頻率。靜態資訊會在使用者登入時，將此資訊自動傳送給線上的使用者，因此當程式需要取得靜態資訊時，就可以直接查找本地端的資料，當無特定需求時，不會特地去詢問對方。更新資訊則是在特定事件發生的時候，自動傳送給線上使用者，因此也能夠直接查找本地端的資料，而不需特地詢問。而詢問資訊，則是在特定應用需求時，到遠端去取得需要的資訊。相較於前者，詢問資訊以詢問者的角度來看，屬於主動式的資訊交流，而不是由被詢問者自動傳送給詢問者，而是在有需要時才主動去詢問對方。

4.1 新增詢問訊息標籤

在上述三種類型的資訊中，目前的 IMNET 僅支援部分的靜態和更新資訊，而不支援詢問資訊。

```
<Info from="user1" to="user2" timestamp="1215476323">
  <queryInfo ask="make friend"/>
</Info>
```

圖 8 詢問訊息範例

```
<Info from="user2" to="user1" timestamp="1215476330">
  <queryInfo askId="1215476323" answer="yes"/>
</Info>
```

圖 9 回應訊息範例

```
<Info from="user2" to="all" timestamp="1215476345">
  <staticInfo name="Jack" avatarName="Businessman"/>
</Info>
```

圖 10 靜態資訊範例

```
<Info from="user2" to="all" timestamp="1215476355">
  <updateInfo position="20 34"/>
</Info>
```

圖 11 更新訊息範例

另外，IMNET 的訊息和 XAML 不同的地方在於 IMNET 沒有支援處理新標籤的功能。因此，為了讓動畫元件彼此間能夠詢問對方資訊，我們在 IMNET 上新增詢問訊息的標籤。我們設計了一個新的 <Info> 標籤來處理化身間互動時的詢問訊息，以及原有的靜態和更新訊息。

圖 8 為此詢問訊息的範例，user1 利用 <queryInfo> 來詢問 user2 是否願意做朋友。user1 在送出訊息時，會以 timestamp 屬性來記錄送出訊息的時間點，如此一來，我們可以設置 timeout 作為等待 user2 回應的最長時間。如圖 9，user2 收到這段訊息後，可以利用此 timestamp 作為 askId，並送出一段回應給 user1。另外，我們以 <staticInfo> 來代表靜態訊息；以 <updateInfo> 來代表更新訊息。如圖 10 和圖 11 關於靜態的資訊和更新的資訊都是被動的接收訊息，如某一使用者行走或更改情緒狀態時，就必須用上述標籤來描述並傳給其他使用者。

4.2 虛擬使用者管理機制

在 IMNET 原先的架構下，使用者和客戶端是結合在一起的，使得在新增虛擬使用者的同時，也要維護一個客戶端介面。因此在這節中我們會說明如何在 IMNET 中加入虛擬使用者，並如何在一個客戶端介面下同時擁有多位使用者。圖 12 為原先在 IMNET 中的連線架構，同一個客戶端擁有唯一的使用者，而伺服器則保留一份最新的線上使用者名單。

在原先的架構下，每加入一位虛擬使用者就必

須新增一個客戶端，每新增一個客戶端就需要新增一個 Socket 和伺服器連線。由於客戶端和使用者是 1 對 1 的關係，假設目前有 n 位使用者在虛擬世界中更新資訊(如某使用者在虛擬世界中移動，則其位

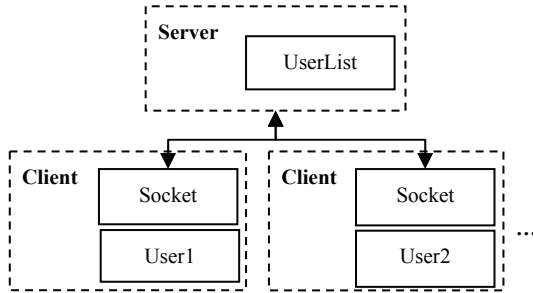


圖 12 原先 IMNET 客戶端架構

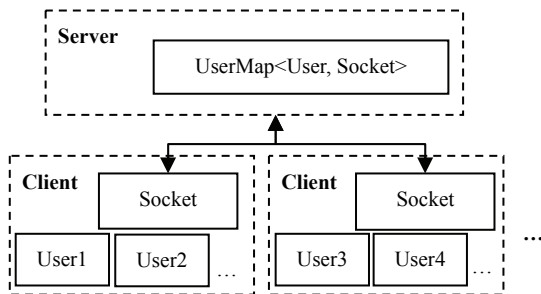


圖 13 具有多重使用者的客戶端架構

置資訊會傳送給其它 $n-1$ 位使用者)，則彼此間的訊息傳送將產生 $O(n^2)$ 的複雜度。因此在同一台電腦下模擬數量龐大的虛擬使用者時，這樣的架構會造成伺服器很快變成整個系統的瓶頸。因此，我們設計了一種方式讓同一個客戶端能同時擁有多位使用者，同一個客戶端下的使用者都透過相同的 Socket 來傳送資料，所有的資料一個客戶端只需傳送一份，再由收到的程式分送給各個真實或虛擬使用者。圖 13 為多重使用者的架構。除了修改客戶端和使用者間的關係之外，我們也修改了伺服器保存線上使用者資訊的資料結構，從原先以 List 的方式來紀錄線上使用者，修改為使用 Map 的結構在使用者和 Socket 間做對應，以記錄使用者使用的通訊介面之物件。

5. 利用語意資訊之 3D 路徑規劃器元件

在本節中所介紹的路徑規劃器元件的目的是在 3D 環境中，透過語意來取得世界中物體的位置和所佔的區域等資訊，並為化身做移動到指定目標地點的路徑規劃。路徑規劃器元件可藉由幾何語意資訊來產生 2D 地圖和位能場，然後用最佳搜尋演算法(Best-First Search)，根據位能場的高低找到一條可行的路徑；最後再將這條路徑以 XAML 的方式描述後交由化身來執行。

我們所設計的路徑規劃器元件主要包含四個部份(如圖 14)。第一個部分是根據場景中障礙物

的位置取得 2D 地圖資訊；第二個部分是藉由 IMBrowser 所提供之世界的語意資訊來建立 2D 地圖的位能場；第三個部分是從 2D 地圖和位能場的資訊，利用位能的高低來搜尋到一條規劃好的路

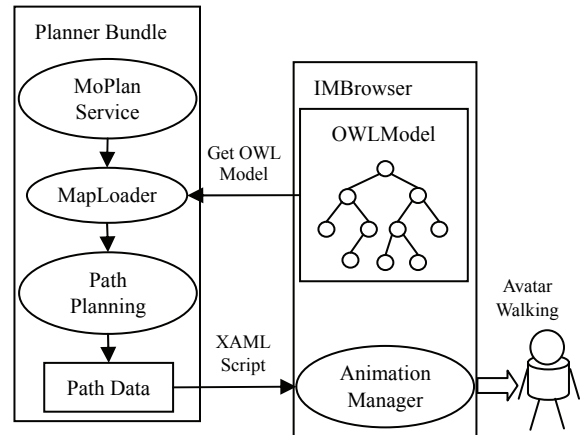


圖 14 路徑規劃器元件產生動畫的流程圖

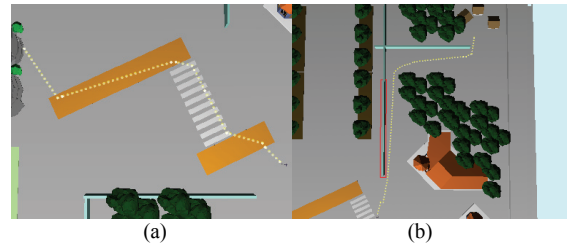


圖 15 由路徑規劃器產生之路徑的範例。(a) 將斑馬線及人行道納入考量後產生的路徑、(b) 避開環境中障礙物的路徑

徑；第四個部分是將規劃好的路徑轉換成可供 IMNET 處理的 XAML 腳本，以產生虛擬人物由起點到終點的動畫。

5.1 化身過馬路情境

我們希望透過虛擬世界中的語意讓路徑規劃器能夠遵守虛擬世界的規則。以過馬路為例，在虛擬世界中有人行道和斑馬線的模型，而在虛擬世界中的規則是化身會沿著人行道和斑馬線行走。因此，規劃的路徑必須以人行道和斑馬線為優先。一個可行的方式就是在人行道和斑馬線上加註 sidewalk 和 crosswalk 標籤，讓路徑規劃器在處理到包含這些標籤的物件時，能夠降低這些物件的位能場，讓最後搜尋到的路徑能夠趨近於這些標註 sidewalk 和 crosswalk 標籤的物件(如圖 15(a)範例所示)。

5.2 化身判斷障礙物情境

我們的路徑規畫器元件所產生避開障礙物的路徑範例如圖 15(b)所示。在語意虛擬環境中，語意只是提供虛擬世界資訊的一種方式，並非絕對必要的屬性。因此我們設計了一個部份缺乏語意資訊的

情境來說明此一特性。如果我們將虛擬世界中某個物體 2D 近似值的語意描述省略掉，讓路徑規劃器找不到該物體的 2D 近似值，則路徑規劃器應該也要能自行計算出 3D 物體的 2D 近似值，只是要花費

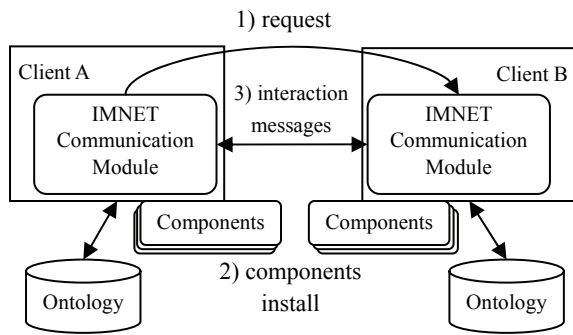


圖 16 化身互動之步驟

```
<SocialService package='imlab.osgi.bundle.interfaces'
codebase='http://imlab.cs.nccu.edu.tw/social.jar'/>
```

圖 17 透過 XML 片段啟動化身間的互動機制

額外的運算，且 2D 近似多邊形計算出的結果也無法由模型的设计者控制。

6. 化身間互動元件

我們主要的目標是希望能讓不同客戶端的動畫元件進行互動，而在溝通上是透過客製化的標籤，來取得雙方化身所擁有的 Ontology 資訊，並利用取得的語意來完成動畫元件間的互動。因此，我們希望能設計出兩階段的範例：第一個階段是證明真實使用者間互動的可行性；第二個階段是證明真實使用者和虛擬使用者間互動的可行性。

6.1 互動元件處理流程

為了讓虛擬世界可以透過 Ontology 來取得化身資訊，並使虛擬世界能夠處理使用者所設計的新元件，我們設計了一套與互動元件溝通的機制，來完成互動元件和虛擬世界的接合。此溝通機制包含了化身間互動請求、安裝互動元件和化身間訊息溝通三個部分。如圖 16 所示，第一步，互動請求者會將互動要求以 XML 的格式傳送給互動回應者；第二步，互動雙方會將此互動所需的元件安裝到系統中；第三步，在互動過程中，互動請求者端的程式會依據使用者或虛擬使用者的需求，將詢問傳送給互動回應者；互動回應者也會提供適當的回答給互動請求者。

化身間互動必須先由某方來啟動整個互動機制。如前所述，使用腳本來操作化身是 IMNET 的特色之一。因此，在化身間互動的啟動上，將透過動態載入動畫元件的機制，讓互動雙方開啟訊息的

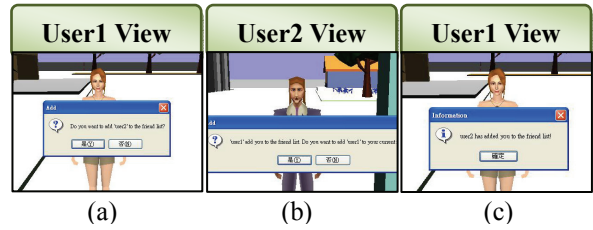


圖 18 加為好友之互動範圍

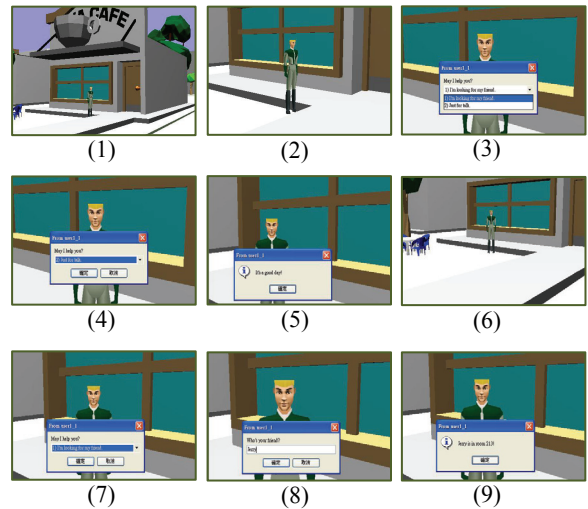


圖 19 User2 與宴會看門者之互動

溝通。如圖 17 所示，互動的一方傳送社交服務的請求給另一方以開啟社交服務所提供之化身間互動程序。

6.2 化身參與宴會範例

以參與宴會為例，在進入會場前使用者首先必須取得對方的信任，並成為對方的好友。在生日聚會的當天，主辦人希望只邀請自己的朋友來參與，因此在會場入口處必須檢查來訪者是否為主辦人的朋友。在化身的 Ontology 中，HasFriend 屬性可以連結不同的化身實體，連結的雙方具有朋友的關係。

如圖 18(a)，User1 希望能將 User2 加為好友，因此在 User2 的視景中(圖 18(b))，就會出現“‘user1‘ add you to the friend list. Do you want to add ‘user1‘ to your current friend list?”的詢問訊息。User2 如果選擇‘是’的話，就會將 User1 加為好友，並回傳一個確認訊息給 User1(圖 18(c))。在此互動中，好友的資訊會被加註在 Ontology 中。之後，看門的虛擬使用者可透過此好友資訊，決定是否要提供資訊給來訪的化身。

接下來是真實使用者和虛擬使用者間的互動。真實使用者向扮演看門者的虛擬使用者詢問好友的位置；看門者會取得詢問者的好友資訊，然後決定是否要告知其所要求的資訊。圖 19 為本範例執行時的實際截圖，User1 安排看門者為來訪的客

人提供資訊(圖 19 (1~2))。當 User2 接近到問答的範圍時,看門者便轉身面向 User2,並詢問“May I help you?”。一開始 User2 可能是不小心路過,所以選擇的回答是“Just look around.”,因此看門者就回了一句“Have a good day!”(圖 19 (3~5))。此時看門者的狀態會被設為 FINISHED,並等待使用者離開問答的範圍並回復到 IDLE 的狀態(圖 19 (6))。然後 User2 第二次接近看門者,並選擇“I'm looking for my friend.”。看門者詢問“Who's your friend?”,User2 回答“Jerry”。看門者這時就會在化身的 Ontology 中詢問 Jerry 是否有一位朋友叫做 User2。如果是的話,看門者就將位置訊息告知 User2;如果不是的話,看門者就會回答“Jerry does not seem to know you.”;如果沒有 Jerry 這位使用者,看門者就會回答“Jerry is not in.”(圖 19(7~9))。

7. 結論與未來研究

為了達到虛擬世界中化身行為的即時共享,我們希望能以 Web2.0 的觀念實現語意虛擬環境。在先前的研究中,我們已經完成動態載入動畫元件的機制,讓設計者所設計的動畫元件可以被動態加入到 OSGi Framework 中,而使用者可透過 XAML 系統來處理包含特定行為的 XML 標籤,透過標籤中的屬性值來下載處理這段 XML 標籤的程式碼,並安裝到系統中。在本論文中,我們說明了如何在虛擬環境系統上加入了 Ontology 的描述,讓動畫元件能透過此描述來取得世界或化身的資訊。與先前語意虛擬環境研究不同的地方在於,我們主要著重於多人虛擬環境系統下,如何利用語意來實現化身的客製化行為。另外,我們延展了 IMNET 的通訊協定,增加了詢問的訊息標籤,同時也提供 IMNET 管理虛擬使用者的機制。我們在最後的部份也提供了兩個元件範例來說明如何透過語意的描述,使化身能與環境或是其它化身作互動。

在未來研究的部分,我們希望在動態載入動畫元件的部份,能加入安全性的考量。另外,在 XAML 的處理上我們希望能提供先處理後傳送的選擇性。在面臨運算量較大的元件時,可以先在本地端的電腦中處理,然後再將運算的結果傳送給其他使用者,如此一來其他使用者就不需要再次運算。在環境和化身語意的延伸上,我們也希望能持續擴充環境或化身語意的豐富性,以面對不同種類的應用。

致謝

此研究在國科會 NSC 96-2221-E-004-008 計畫的支助下完成,特此致謝。

參考文獻

- [1] T. Abacı, J. C'iger and D. Thalmann “Action semantics in Smart Objects,” *Proceedings of Workshop towards Semantic Virtual Environments*, 2005.
- [2] R. Aylett and M. Cavazza, “Intelligent Virtual Environments - A State-of-the-art Report,” *Proceedings of Eurographics*, 2001.
- [3] Y.L. Chu, T.Y. Li and C.C. Chen, “User Pluggable Animation Components in Multi-user Virtual Environment,” *Proceedings of the International Conference on Intelligent Virtual Environments and Virtual Agents*, China, June 2008.
- [4] A. Garcia-Rojas, F. Vexo, D. Thalmann, A. Raouzaoui, K. Karpouzis and S. Kollias, “Emotional Body Expression Parameters In Virtual Human Ontology,” *Proceedings of 1st International Workshop on Shapes and Semantics*, pp. 63-70, Matsushima, Japan, 2006.
- [5] M. Guti'erez, D. Thalmann, F. Vexo et al., “An Ontology of Virtual Humans: incorporating semantics into human shapes,” *Proceedings of Workshop towards Semantic Virtual Environments*, 2005.
- [6] F. Kleinermann, O.D. Troyer, C. Creelle and B. Pellens, “Adding Semantic Annotations, Navigation paths and Tour Guides to Existing Virtual Environments,” *Proceedings of 13th International Conference on Virtual Systems and Multimedia (VSMM'07)*, Brisbane, Australia, 2007.
- [7] T.Y. Li, M.Y. Liao and J.F. Liao, “An Extensible Scripting Language for Interactive Animation in a Speech-Enabled Virtual Environment,” *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2004)*, Taipei, Taiwan, June 2004.
- [8] T.Y. Li, M.Y. Liao and P.C. Tao, “IMNET: An Experimental Testbed for Extensible Multi-user Virtual Environment Systems,” *Proceedings of International Conference on Computational Science and its Applications, LNCS 3480*, O. Gervasi et al. (Eds.), Springer-Verlag Berlin Heidelberg, pp. 957-966, May 2005.
- [9] K.A. Otto, “The Semantics of Multi-user Virtual Environments,” *Proceedings of Workshop towards Semantic Virtual Environments*, 2005
- [10] Second Life, <http://secondlife.com>.