관련 영역 관리 방식을 이용한 가상 환경 관리기의 설계

탁진현*, 이승근*, 장진윤*, 이세훈**, 왕창종*

Design of Virtual Environment Manager using Management of Interested Area

Jin-Hyun Tak*, Seung-Geun Lee*, Jin-Yun Chang*, Sei-Hoon Lee**, Chang-Jong Wang*

ABSTRACT

It is actively studying that research about virtual environment supporting multi-users that can support interaction for collaborative work. This virtual environment system is doing often interaction with users participating in virtual environment, but legacy virtual environment systems don't provide a management considering of interaction with participants.

In this study, we define the concept of activity area considering distance of participants. Also, we design virtual environment manager using this concept. So, participants in the same activity area should share their events, and through this mechanism, the system can provide the consistence between interacting participants. Also, participants register to events they interested. As the system can transmit to events specific participants by registered events, transmitting overhead of events is decreased.

Key Words **: 가상 공간 관리, 이벤트 처리, VRML**

.

1. 서 론

컴퓨팅 능력의 발달과 네트워크의 개선으로 인해서 다중 사용자를 지원하는 가상 환경에 대한 연구가 활발하게 진행되고 있다. 특히, 3차원 가상 공간은 공동 작업을 위한 상호 작용을 보다 현실감있게 지원할 수 있다는 측면에서 각광받고 있다[1].

특히, 다중 사용자를 효과적으로 지원하기 위해서 가상 공간의 관리 기술에 대해서 많은 연구가 진행되고 있다. 이러한 연구들은 아직도 전송되어야 하는 많은 데이터에 비해서 상대적으로 느린 네트워크 때문에 어떻게 하면 보다 적은 데이터의 이동으로 가상 공간을 관리할 수 있는가에 초점이 맞추어지고 있다. 기존의 연구에서는 가상 공간 관리 방식을 중앙 집중형 서버-클라이언트 모델, Peer-to-Peer 모델, 부분 공간 모델 등으로 설계하고 있다. 그러나, 이들 모델에서는 참여자간의 상호작용을 고려한 관리 방식을 제공하지는 못한다. 다시 말해서, 다중 사용자가 하나의 가상 공간에 참여하는 경우는 대부분 빈번한 상호작용이 이루어지며, 따라서 가상 공간 관리에서 이를 고려하면 보다 효과적인 가상 공간 관리가 이루어질 수 있을 것이다.

본 연구에서는 이를 위해서 참여자간의 거리를 고려해서 활동 범위라는 개념을 적용하고, 이를 통해서 참여자간의 상호작용을 공간 관리에 이용하려고 한다. 활동 범위는 참여자의 활동이 참여자에게만 국한되는 것이 아니라 다른 사용자에게 밀접한 영향을 미칠 수 있는 범위를 말한다. 따라서, 같은 활동 범위에 있는 참여자들간에는 이벤트가 반드시 공유되어야 하며, 이를 통해서 상

* 인하대학교 전자계산공학과
{tak,sglee,ellio}@selab.inha.ac.kr, cjwangse@inha.ac.kr
** 인하공업전문대학 전자계산기과
seihoon@true.inhatc.ac.kr

호작용이 이루어지는 참여자들간에 일관성을 유지시킬 수 있다. 또한, 참여자가 자신이 원하는 이벤트를 등록시키고, 이를 바탕으로 특정 참여자에게만 이벤트를 전송시킬 수 있게 함으로써 이벤트의 전송 부담을 감소시킬 수 있다.

2. 기존에 제안된 가상 공간 관리 구조

이 장에서는 가상 공간 관리기를 구성하기 위해서 필요한 가상 공간 관리 방식과 이벤트 처리 방식이 기존 연구에서 어떻게 진행되어 왔는지에 대해서 설명한다.

2.1 가상 공간 관리 방식

가상 공간 관리기의 구조는 공간 관리를 어디에서 할 것인지에 따라서 중앙 집중형 서버-클라이언트 모델과 Peer-to-Peer 모델, 부분 공간 (Partitioned World) 모델로 구분할 수 있다[2].

중앙 집중형 서버-클라이언트 모델

중앙 집중형 서버-클라이언트 모델은 공유 데이터를 중앙의 서버에 집중하여 관리하는 모델이다[3]. 참여자가 가상 공간내에서 행하는 작업들은 이벤트의 형태로 모두 중앙 서버에 전달되고, 서버는 전달된 이벤트를 가상 공간내에 반영하고, 변경된 가상 공간을 각각의 사용자에게 전달하는 방식이다. 이 방식은 구성이 단순하고, 개발이 쉬우며 공간에 대한 일관성 유지가 용이하다는 장점을 가지고 있으나, 서버에서 모든 참여자에서 발생한 이벤트가 처리되어야 하는 문제점으로 인해서 확장성이 부족하며, 메시지 병목 현상, 서버의 신뢰도 문제등 여러 문제를 가지고 있다. 이러한 모델을 채택한 예로는 BrickNet, AVIARY, dVS등이 있다[4][5][6].

Peer-to-Peer 모델

이 모델은 가상 공간에 대한 정보를 각각의 참여자가 자신의 지역 시스템에 완전히 복제한 상태에서 함께 참여하고 있는 참여자들 간의 통신을 이용하는 모델이다[2]. 따라서, 가상 공간 내에서 발생하는 모든 이벤트는 모든 참여자의 시스템에 전달되고, 참여자의 지역 시스템에서 가상 공간에 반영한다. 이러한 방식은 중앙 집중형 서버-클라이언트 모델에 비해서 메시지의 병목 현상과 서버의 계산 부담을 효과적으로 줄일 수 있으며, 이를 통해서 전체적인 지연 시간을 줄일 수 있다. 그러나, 일관성 유지가 어렵고, 클라이언트의 계산 부담 등의 문제를 가지고 있다. 특히, 지역 시스템에서 가상 공간에 대한 관리가 이루어지기 때문에 가상 공간이 커질수록 불필요한 메시지가 많이 전달되어 오는 단점을 가지게 된다. 또한, 관리하는 서버가 없기 때문에 이동 객체를 구현하기 힘들다. 이 모델을 사용한 시스템으로는 NPSNET과 DIVE 시스템등이 있다[7][8].

부분 공간 모델

부분 공간 모델은 전체 가상 공간을 일정한 영역으로 나누고 각각의 영역을 참여자에 보여줌으로써 대규모의 가상 공간과 함께 지역적으로 분할된 많은 수의 참여자를 동시에 지원하기 위해서 제안되었다[1]. 이와 같은 시스템들은 전체 가상 공간내에 room이라는 개념을 도입해서 이 room 안에서만 상호작용이 발생하도록 하고 있다. 앞에서 설명한 두가지 모델보다는 확장성이 매우 우수하나, room 기반의 기법은 전체적인 가상 공간의 일관성을 보여주지 못하는 문제점이 있다. 이를 해결하는 방법이 전체 가상 공간을 셀로 구성하고, 한 참여자가 발생한 이벤트를 같은 셀에 위치한 아바타에게 전송하는 방식을 채택한다. 그러나 이 방식은 다음 Fig. 1과 같이 참여자가 위치하는 경우 문제점을 가지고 있다.

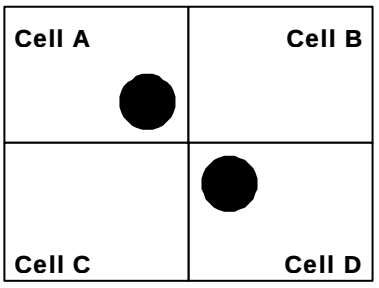

Fig. 1. 기존의 셀 방식 가상 공간 관리

Fig. 1에서와 같이 셀 A와 셀 D에 참여자가 있는 경우, 두 참여자는 실제 거리는 가까우나 다른 셀에 위치하기 때문에 서로 이벤트가 전송이 이루어지지 않기 때문에 가상 공간을 관리하는데 문제가 있다.

또한, 다중 사용자를 고려한 가상 공간은 사용자들이 개별적으로 활동하기 보다는 사용자들간의 상호작용이 보다 중요한 요소가 된다. 참여자가 셀을 이동하는 경우에는 이동된 셀의 정보는

참여자가 가지고 있지 않기 때문에 서버로부터 정보를 전달받아야 한다. 사용자들간의 거리가 가깝다는 것은 서로 상호작용이 매우 빈번하게 발생할 수 있으며, 따라서, 이들간의 위치 이동 역시 매우 빈번하게 발생할 수 있다고 생각할 수 있다. 따라서, 각각의 개별적인 참여자의 위치로 가상 공간을 관리하기 보다는 참여자들간의 상호작용 관계를 고려해서 가상 공간을 관리하는 것이 보다 적합하다.

2.2 이벤트 전송 방식

이벤트 통지는 주로 이벤트 전송자와 이벤트 수신자 사이의 단 방향 통신으로 설명할 수 있다. 이러한 전송하는 이벤트들을 처리하기 위한 방식으로는 classic, mediator, observer 방식 등이 있다[9].

Classic 방식

이벤트가 발생할 때마다 이벤트 전송자는 발생되는 이벤트들을 수신자에게 전달하는 방식이다. 예를 들어, 교사가 학생에게 시험 기한을 알릴 경우 교사는 이벤트 전송자이고 학생은 이벤트 수신자이다. 많은 학생들이 있다면, 교사는 각각의 학생들에게 통지해야 하는 경우가 발생할 것이다. 컴퓨터 프로그램에 적용하면, 호출된 컴포넌트들중 하나의 통지 인터페이스에 변화가 있을 때 마다, 관련된 모든 컴포넌트가 변경되어야 하는 것이다.

Mediator 방식

위 classic 방식의 제한을 극복하기 위하여 제안된 방식이다[9]. 이벤트 송신자와 수신자사이에 상호작용을 조정하기 위하여 관리되는 컴포넌트로 mediator를 두는 것이다. 이벤트 송신자들은 이벤트들을 직접 전달하는 것이 아니라 mediator가 이벤트들을 모아서 이벤트 수신자들에게 전달하는 방식이다. 이 경우, 이벤트 송신자들이나 수신자들은 누가 누구한테서 이벤트들이 전달된지를 관여를 안 하게 된다.

Observer 방식

Observer 방식은 이벤트 수신자들이 자신이 관심 있는 이벤트들을 등록해 놓고, 이벤트들이 발생을 하면 observer에서 이벤트들을 필터링하여 원하는 이벤트들만 받게 하는 방식이다[9]. 이러한 접근 방식은 Gupta et al[10]이 설명한 것처럼, 시스템의 다른 부분에 영향을 주지 않고 새로운 관리 시스템이 추가 될 수 있으므로 mediator 방식보다 더욱 동적인 행위가 가능하다.

이 논문에서는 위 mediator 방식과 observer 방식의 장점만을 절충하여서 새로운 이벤트 서비스 방식인 notifer 방식을 도입하였다.

3. 동적 관련 영역 결정 방식을 이용한 가상 공간 관리기

이 연구에서는 확장성이 우수한 가상 공간 모델을 변형함으로써 부분 공간 모델의 문제점인 전체적인 가상 공간의 일관성 유지를 가능하게 하는 방법을 제안한다. 이를 위해서 설계되는 가상 공간 관리기에서는 전체 가상 공간을 여러 개의 셀로 구성한다. 기존 연구에서는 하나의 셀에 위치한 참여자들에게 메시지를 전송하는 방법의 문제점을 해결하기 위해서 참여자들 간의 인접도와 상호작용 간의 관계를 고려해서 관련 영역이라는 개념을 도입하고, 이 관련 영역을 동적으로 결정하는 방식을 이용한다. 동적 관련 영역 결정 방식은 참여자들의 인접도를 계산해서 한 참여자의 행위가 영향을 미칠 공간의 영역을 동적으로 결정하고, 이를 바탕으로 가상 공간을 관리하는 방식이다. 또한, 참여자가 원하는 이벤트의 종류를 등록시킬 수 있게 함으로써 불필요한 이벤트의 전달을 방지할 수 있게 하였다.

3.1 시스템 설계

이 연구에서는 사용자의 상대적인 위치를 고려해서 이벤트를 전송할 참여자의 범위를 결정하고 참여자가 원하는 이벤트를 등록할 수 있는 동적 셀 결정 방식을 이용한 가상 공간 관리기를 설계한다. 설계된 가상 공간 관리기는 Fig. 2 와 같다.

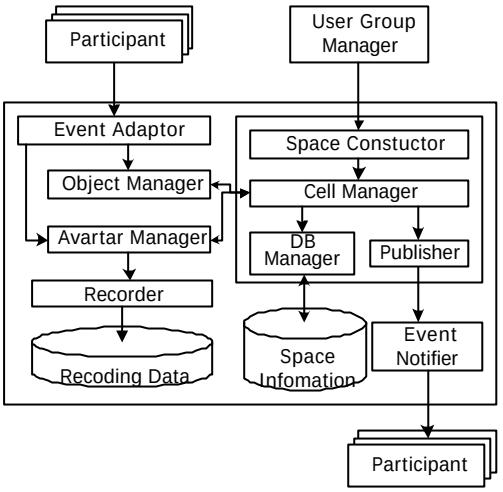

Fig. 2. 동적 셀 결정 방식을 이용한 가상 공간 관리기

가상 공간 관리기는 참여자로부터 이벤트를 전송받아서 객체 관리기와 아바타 관리기기에 전달하는 이벤트 어댑터와 아바타의 동작을 저장하고 관리하기 위한 레코더, 참여하는 참여자를 고려해서 가상 공간을 구성하는 공간 구성자와 전체 가상 공간내의 아바타의 위치를 고려해서 동적으로 가상 공간을 관리하는 셀 관리기, 아바타가 관심있는 이벤트를 등록시킬 수 있는 등록기와 이벤트 전송자로 구성된다.

3.2 셀 관리기

셀 관리기는 공간 구성자가 생성한 가상 공간을 셀 단위로 분할해서 관리한다. 셀의 크기는 응용에 따라 결정된다. 셀의 크기에 영향을 미치는 요인으로는 아바타의 이동 형태나 아바타의 크기, 상호작용의 정도등이 될 수 있다. 셀 관리기는 아바타간의 거리를 동적으로 계산해서 전체 가상 공간을 셀 기반으로 한 관련 영역으로 분할한다. 관련 영역내에 포함된 참여자들간에는 이벤트를 공유하게 한다. 관련 영역을 결정하는 방법은 다음과 같다. 하나의 참여자가 위치한 셀과 그 주변의 셀들을 초기의 관련 영역으로 정의하고, 정의된 관련 영역내에 다른 참여자가 위치한 경우에는 관련 영역내에 위치한 참여자로 확대한다. 알고리즘은 다음과 같다.

알고리즘 1. 관련 영역 결정 알고리즘

```
Input : Virtual Space which is divided by cells
Output : Concern area infomation
Begin
  while( fibot avatar <- avatar is not includes
any Concern area)
    begin while
      boolean expandable <- true;
     while(expandable)
     begin while
       Concern Area <- Concern Area +
              cell where lie the fibot avatar +
              circumstancial cell of fibot
avatar;
        if there are not other avater in Concern
Area then
         expandable <- false;
     end while
     Store Concern area infomation
   end while
 End
```

알고리즘 1을 이용하면 그림과 같이 아바타가 위치한 경우에는 세 개의 관련 영역을 추출할 수 있다. 아바타가 셀과 셀 사이를 이동하는 경우 셀 관리기는 알고리즘 1을 이용해서 동적으로 활동 영역을 결정하고 이를 통해서 얻어진 관련 영역 정보를 관리한다.

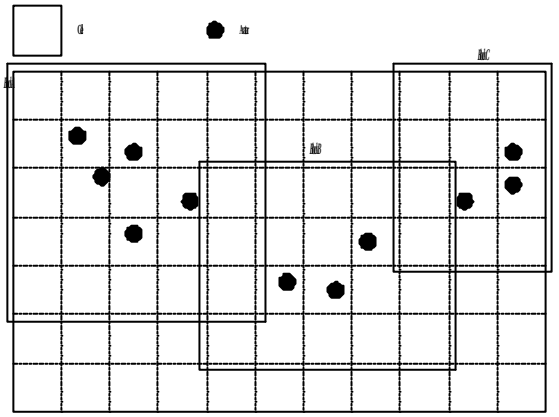

Fig. 3. 관련 영역으로 분할된 가상 공간

참여자에서 발생한 이벤트는 같은 관련 영역안에 있는 사용자들에게 전달된다.

3.3 아바타 관리기

아바타 관리기는 아바타의 위치를 추적, 관리하는 기능을 수행한다. 이러한 기능은 현재 인터넷 분야에서 사용자의 폭발적인 증가를 보이는 쇼핑몰과 같은 응용에서 상품의 제원, 가격, 사진 등의 전달뿐만 아니라 대형의 공간에서는 사용자의 쇼핑을 돕기 위한 기능으로서 사용자의 위치 추적이 큰 역할을 할 수 있다. 위치추적 기능을 구현하기 위하여 ProximitySensor와 Viewpoint를 이용한다. ProximitySensor의 크기는 구현된 가상공간을 포함할 만큼 충분히 크게 해야 한다.

그렇지 않을 경우 이 센서의 위치를 벗어난 곳에서는 사용자의 위치추적이 실패하게 된다. VRML에서의 길이 단위는 암시적으로 미터이므로 여기에서는 가로, 세로, 높이 각 1㎞가 되는 센서를 설치하였다. 센서의 중심은 가상공간의 중심과 맞도록 해둔다.

ProximitySensor 외에 두 개의 Viewpoint를 설치한다. 첫 번째 Viewpoint는 가상공간을 표현하는 자바 프로그램에서 마우스 이벤트가 발생할 경우 VRML로 구현한 가상공간에서 해당하는 위치로 이동하기 위한 것이며 두 번째 Viewpoint는 위치 이동을 위한 보조적인 역할을 한다.

VRML 코드 내에 위와 같이 센서와 뷰포인트를 설치한 다음, 가상공간에서 사용자의 내비게이션을 돕기 위하여 가상공간의 2차원 평면을 표현하는 이미지를 준비한다. 이것은 구현된 가상현실 공간의 Viewpoint 위치를 가상공간 중심의 위에서 아래 방향으로 조절하여 얻을 수 있다.

3.4 이벤트 등록기

이벤트 등록기의 기본적인 형태는 publish/subscribe 메카니즘을 따른다.

publish/subscribe 메카니즘은 클라이언트가 서버쪽에 이벤트를 요구하는 방식이 아니라 서버에서 이벤트가 발생하면 그 이벤트들이 클라이언트 쪽으로 전송되는 형태이다. 또한 클라이언트들은 자신들이 관심이 있고, 받기를 원하는 이벤트들과 메시지들을 채널에 등록을 하여서 그 이벤트들과 메시지만을 받을 수 있도록 한다. 다시 말하면, 이벤트들은 계속 발생을 하게 되고, 발생되는 이벤트들은 이벤트 수신자에게 계속 전송이 된다. 채널은 발생되는 이벤트들을 모아서 이미 등록되어 있는 이벤트 리스트를 참조하여 클라이언트들에게 송신하게 된다.

참여자들은 처음 참여할 때 자신이 관심 있는 이벤트들을 이벤트 등록기에 등록을 하게 된다. 이벤트가 발생하는 경우, 그 이벤트들은 이벤트 어댑터로 전송이 되고, 이벤트 어댑터는 아바타들을 관리하는 아바타 관리자와 객체들을 관리하는 객체관리기에 전송을 한다. 이벤트들을 전송할 때는 등록되어 있는 참여자들과 이벤트들을 참조하여 전송하므로 전체 네트워크 트래픽을 줄이도록 하였다.

또한 통지 서비스에서 요구되는 디자인 이슈 중에 이벤트 전달에 대한 QoS(Quality of Service)가 있는데, 이 논문에서는 QoS 중 우선순위와 신뢰성에 대한 처리 방식으로 이벤트를 두 가지로 구분하여 처리하도록 하였다. 즉, 아래와 같이 서로 다른 이벤트 타입에는 각각의 QoS가 필요한 경우이다.

·참여자들이 이해하여야 하는 정보인 content event는 신뢰할 수 있게 전달될 필요가 있다. 그러나 강제적인 전송 시간의 제약은 필요하지 않을 것이다.

·반대로, 사용자 인터페이스와 관련된 presentation event는 신뢰도는 낮아도 되지만, 더 강력한 전송 우선순위를 갖을 것이다.

이벤트 전송기는 이러한 이벤트 전송시 네트워크의 대역폭을 고려하여 이벤트들을 표현하게 하였다. 즉, presentation 이벤트의 경우 네트워크의 대역폭 용량이 작을 경우 그 사실을 표현하는데 텍스트로 표현하는 것도 가능할 것이다.

4. 구현

이 장에서는 참여자들간의 상호작용이 빈번하게 발생하는 교육 시스템을 구현함으로써 3장에서 설계한 가상 공간 관리기를 입증한다. 실험환경은 100base-TX Ethernet의 TCP/IP 프로토콜 상에서 운영 체제는 MS WindowsNT Server 4.0을 사용하였고, WWW 서버로는 Internet Information Server 4.0을 사용했다. 각각의 클라이언트는 펜티엄 Ⅱ 450MHz PC에 운영 체제로는 Windows 98을 사용하였다. 가상 공간을 보기 위해서 가상 공간 브라우저인 cosmoplayer 2.11을 각 pc에 addon 하였으며, 제공되는 인터페이스는 Jbuilder 3.0을 이용하여 구현하고, VRML 97을 이용하여 가상 공간을 구현하였다. 가상 공간과 자바 언어와의 인터페이스는 EAI를 이용하였다 [11][12].

본 실험은 가상 공간에는 강의자 1명과 학습자 30명으로 구성된 학급을 보여준다. 교사는 수업 중에 강의실을 계속 이동하면서 학생들을 지도한다. 학생들은 3~4명씩 한 조를 구성해서 조별 활동을 통해서 학습을 진행한다. 교사는 여러 조들을 이동하면서 활동 지도와 조들간의 상호작용을 돕는다.

Fig. 4. 과제물을 설계하는 과정

실험을 통하여 논문에서 설계한 가상 공간 관리기는 가상 공간이 매우 큰 경우에 보다 높은 이벤트의 전송 효율을 얻을 수 있었다.

5. 결론

이 연구에서는 참여자간의 상호작용을 고려한 활동 범위 개념을 도입한 가상 공간 관리자를 설계하였다. 설계한 가상 공간 관리자는 기존의 가상 공간 관리자에서 제공하지 못하는 참여자간의 상호작용 요소를 공간 관리에 도입함으로써, 참여자들간의 일관성을 보장하였다. 또한, 참여자가 원하는 이벤트를 등록하고, 이를 통해서 이벤트를 전송하는 방법을 도입함으로써, 전송되는 이벤트의 양을 감소시키고, 이를 통해서 참여자 시스템들의 부담을 감소시켰다. 또한, 가상 공간내에서의 참여자의 이동 경로를 관찰하고 이를 관리함으로써, 전자 상거래 공간이나 극장 혹은 공연장의 좌석에 따른 무대의 3차원 공간에서의 확인, 가상현실 캠퍼스나 가상현실 모델 하우스에서의 위치 확인, 가상 공간내에서 참여자의 행동을 분석할 수 있는 방법을 제공함으로써 교육 시스템에서 학습자 관리등 의 분야에 응용될 수 있다. 이 논문에서 제안한 가상 공간 관리 시스템은 대규모 가상 환경상에서 참여자들간의 상호작용이 빈번하게 발생하는 경우 보다 효과적이며, 현재 고정적인 셀의 크기가 아닌 응용의 특성이나 참여자의 동작 형태 분석을 통한 동적 셀 크기 결정에 대해서 연구하고 있다.

참고문헌

[1] 성운재, 심재한, 원광연, "다중 참여자 네트워 가상현실 시스템을 위한 복수 멀티캐스트 통신구조," *한국시뮬레이션학회 논문지*, 7권, 1호, 1998

[2] Gossweiler, R, Robert J. Laferriere, Michael L. Keller, and Randy Pausch, "An Introductory Tutorial for Developing Multiuser Virtual Environments," PRESENCE, Vol. 3, No. 4, MIT press, Fall 1994, pp. 255-264

[3] Gisi, M. A., Cristiano Sacchi, "Co-CAD: A Collaborative Mechanical CAD System," PRESENCE, Vol. 3, No. 4, MIT press, Fall 1994, pp. 341-350

[4] Gurminder Singh, Luis Serra, Willie Png, Audrey Wong, and Hern Ng, "Bricknet: Sharing Object Behaviors on the Net," *IEEE VRAIS*, pp. 19-25, 1995

[5] Snowdon D. N., Adrian J. W, "AVIARY: Design Issues for Future Large-Scale Virtual Environments," PRESENCE, Vol. 3, No. 4, MIT press, 1994.

[6] Steve Ghee, "dVS A distributed VR systems infrastructure," SIGGRAPH ' 95 Course Note, 1995

[7] Macedonia, M.R., M.. J. Zyda, D.R. Pratt, T.Barham, and S.Zeswitz, "NPSNET: A Network Software Architecture for Large-scale Virtual Environments," PRESENCE, Vol. 3, No. 4, MIT press, Fall 1994.

[8] Carlsson C, Olof Hagsand, "DIVE – A Multi-user Virtual Reality System," *Proceedings of IEEE VRAIS ' 93*, Seattle, September 1993, pp.394-400

[9] Gamma et al., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1997.

[10] S. Gupta, J. Hartkopf, S. Ramaswamy, "Event Notifier: A Pattern for Event Notification", in Java Report Magazine, July 1998.

[11] The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997

[12] External Authoring Interface Specification http://www.vrml.org/WorkingGroups/vrml-eai/