

점진적 데이터 전송을 지원하는 분산 가상현실 서비스 시스템

○
황상훈*, 황대훈*
*경원대학교 전자계산학과

Distributed Virtual Reality Service System Supporting Progressive Data Transmission

○
Sang-Hoon Hwang*, Dae-Hoon Hwang*
*Dept. of Computer Science, Kyungwon Univ.

요 약

웹에서 3차원 인터페이스를 제공하는 가상현실 공간을 구축하기 위해 VRML이 등장하였다. 또한 웹과 가상현실을 접목하고자 하는 과정에서, 3차원 가상환경을 사용자들이 네트워크를 통해 서로 공유하며 실시간으로 서로의 움직임을 인지하고 의사 전달을 하며 공동 작업을 수행하기 위한 분산 가상현실이라는 분야가 등장하였으나, 아직까지 실세계와 유사한 분산 가상현실을 지원하지 못하고 있는 실정이다.

이에 본 논문에서는 VRML 언어로 표현된 3차원 가상공간을 분할하여 관리하기 위한 가상세계 서버와 참여자들간의 상호작용 이벤트를 중계하고 공간 데이터를 클라이언트에 전송하기 위한 중계 서버, 그리고 이들 서버와 통신하는 가상세계 클라이언트를 설계하였다. 특히 본 시스템에서는 가상공간을 분할공간 단위로 세분하여 이 단위로 서버가 클라이언트에게 전송하는 방법을 제시하였으며, 아바타의 행위 정보를 타 참여자에게 전송 분할공간 개념을 이용하여 같은 분할공간에 속한 참여자에게만 전송함으로써 불필요한 통신 부하를 줄이고 사용자 네비게이션 환경을 개선하였다.

1. 서론

오늘날 인터넷은 HTML을 기반으로 직관적인 하이퍼미디어 방식의 채용과 텍스트 형태의 데이터 및 그림 데이터와 같은 시각적인 정보의 표현이 가능해짐에 따라, 인터넷 사용을 촉발하는 계기가 되었다. 그러나, 기존의 인터넷에서의 정보 표현 방법은 2차원적 평면 정보에 의존함으로써, 보다 효율적인 정보의 표현에 한계가 있어 왔다.

이에 인터넷상에서 3차원 정보를 표현함으로써 사용자에게 몰입감과 입장감을 주어 좀더 현실감 있는 표현을 하고자 하는 요구가 대두되었다. 이러한 요구에 따라 1994년 11월 Silicon Graphics사의

Open Inventor를 토대로 한 VRML (Virtual Reality Modeling Language)이 등장하였다. 현재 VRML은 인터넷을 통해 3차원 정보를 표현하기 위한 표준으로 채택되었다[1].

그러나 이러한 VRML의 지원 형태는 단순히 가상공간을 3차원으로 표현하는 수준으로서, 가상공간이 3차원으로 표현됨과 동시에 다수의 참여자(participant)가 그 가상공간에서 의사소통 및 상호작용을 할 수 있도록 개선되어야 할 필요성이 대두되었다. 이렇게 3차원 가상공간상에서 참여자들이 네트워크를 경유하여 가상공간을 공유하고 실시간에 상대방의 위치와 행위를 인지하고 의사 전달을 하며 공동작업도 할 수 있는 분산 가상현실(distributed

VR) 시스템이 개발되었다.[5] 이러한 분산 가상현실 시스템들은 참여자가 서버에 접속 시 가상 공간을 구성하는 VRML 파일 전체를 클라이언트로 전송되며, 파싱 및 렌더링 과정에 있어 클라이언트에 많은 부담을 주게 된다. 또한, 모든 참여자에게 아바타 행위 정보를 전송함으로써 불필요한 데이터 전송에 따른 네트워크 트래픽을 증가시키는 단점을 가진다.

이에 본 논문에서는 분산 가상현실 시스템에서 가상공간을 분할공간(region)이라는 공간 단위로 세부 분할하기 위한 방법을 연구하였으며, 또한 아바타의 행위 정보를 타 참여자에게 전송 시 분할공간 개념을 이용하여 같은 분할공간에 속한 참여자에게만 전송하기 위한 방법을 설계하고 구현하였다.

2. 관련 연구

2.1 가상현실

가상현실(virtual reality)은 컴퓨터가 만들어낸 가상 환경(virtual environment)에서 사용자가 마치 실제 세계에서 상호작용 하는 것처럼 할 수 있게 해주는 인간과 컴퓨터와의 진보된 인터페이스이다. [2]

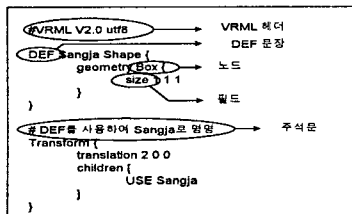
VRML은 이러한 가상현실 세계를 특히 인터넷상에서 웹 브라우저를 통해 가상현실 세계를 표현하기 위해 고안된 개방형이며 기계 중립적인 언어로써, 기존의 HTML 문서들처럼 URL(Uniform Resource Locator)을 통해 접근할 수 있으며 MIME 타입으로는 model/vrml을 사용한다.[1]

2.2 VRML과 EAI

VRML은 Virtual Reality Modeling Language의 약어로 본 절에서는 VRML의 구조적 특징과 기능과 EAI에 대해 기술한다.

2.2.1 VRML의 구조

VRML 파일은 헤더, 노드, 필드 및 문장 등으로 구성된다. 이들을 포함하고 있는 파일은 브라우저에 의해서 처리되고 표현되며, 사용자와 상호작용을 하게 된다.[2][11] <그림2.1>은 간단한 VRML 파일의 소스 예이다.



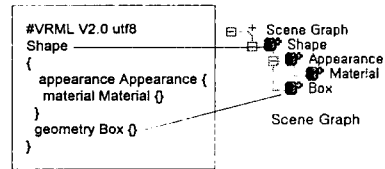
<그림2.1> VRML 파일의 소스 예

(1) 노드와 필드

VRML2.0에서는 54개의 노드를 지원하고 있는데, 구체적으로 화면에 물체를 나타내는 역할을 하는 노드도 있지만 직접 눈에 보이는 결과를 만들지 않는 노드도 있다. 노드는 기능에 따라 그룹노드(group node), 센서노드(sensor node), 구조노드(geometry node), 외양노드(appearance node), Bindable node로 묶을 수 있으며 필드는 크게 MF(Multiple Field)와 SF(Single Field)로 나눌 수 있으며 이로는 MFString, MFFloat, SFString, SFFloat 등이다.

(2) Scene graph의 구조

Scene graph는 장면의 구조를 계층적으로 표현하여 사용자에게 보여주며 이는 사용자가 가상세계를 저작할 때 각 노드가 어떤 관계로 구성되어 있는지 알 수 있을 뿐만 아니라 어떻게 이를 구성해야 할 지 보여준다.



<그림2.2> Scene graph의 구조

<그림2.2>는 실제 .wrl 파일이 scene graph의 계층 구조로 표현된 예를 나타낸 것으로서, 'Scene graph'로 표시된 부분인 최상단의 루트 노드로부터 출발하여, 위에서 아래로, 좌에서 우로 해당 노드를 찾아가면서 그 기능을 실행해 간다.

2.2.2 EAI

EAI(External Authoring Interface)는 SGI사의 Chris Marrin이 1997년에 제안한 것으로, VRML 브라우저와 자바 애플릿과 같은 외부 세계와의 인터페이스로 외부 어플리케이션과의 통신을 지원한다. EAI를 통해 외부 프로그램은 VRML 브라우저가 가지고 있는 scene graph의 계층 구조를 변화시키거나 특정 노드에 접근하여 필드의 값을 바꾸어 그 결과를 VRML 브라우저에 반영하게 한다. 또한 VRML로 표현된 특정 객체를 문자열로 입력받아 직접 scene graph에 삽입할 수도 있다.[2]

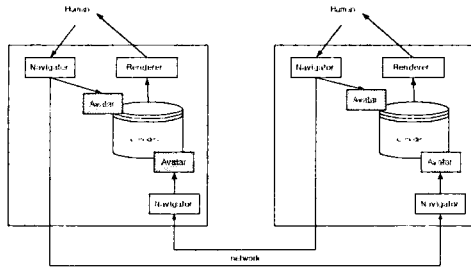
2.3 분산 가상현실 시스템의 유형

본 절에서는 본 논문의 대상이 되는 가상현실 시스템에 대하여 간단히 개념적 구성을 알아본다.

2.3.1 단순 가상현실 시스템

단순 가상현실 시스템은 다수의 참여자가 동시에 같은 공유 공간에 접속하여 가상환경을 둘러보고

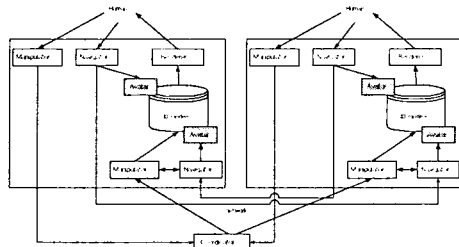
서로 상대방을 인지하는 것을 지원한다.[3]
 <그림2.3>은 단순 가상현실 시스템을 나타낸 것이다.



<그림2.3> 단순 가상현실 시스템

2.3.1 동적 가상현실 시스템

동적 가상현실 시스템은 단순 가상현실 시스템 기능에 더하여 참여자의 공유 가상환경에 대한 자유로운 갱신이 가능한 것이다. 그 개념적 구조는 다음 <그림2.4>와 같다. Manipulator와 Coordinator를 제외한 부분은 앞의 단순 가상현실 시스템과 거의 같다.



<그림2.4> 동적 가상현실 시스템

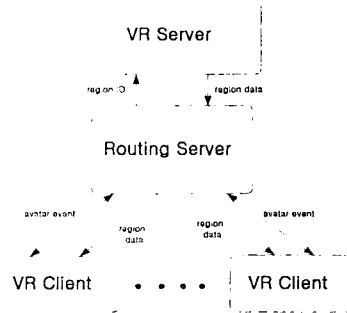
Manipulator는 참여자의 조작을 추출하고, 조작은 coordinator에 의하여 중재된 후에 각 노드의 가상 환경에 적용된다. 다수 참여자의 동시조작이 분산 복제된 가상환경에 가해지므로 각 참여자 노드에 복제된 가상환경의 일관성을 유지하는 coordination 방법이 필요하다.[3][4]

본 논문에서는 이러한 분산 가상현실 시스템 유형중 동적 가상현실 시스템 구조를 사용하여 시스템을 확장하였다.

3. 시스템 설계

본 시스템은 크게 가상세계 서버(VR server), 중계 서버(routing server), 가상세계 클라이언트(VR client)로 이루어진다. <그림3.1>은 본 논문에서 설

계한 분산 가상현실 시스템의 전체 구조도를 나타낸 것이다.



<그림3.1> 전체 시스템 구성도

가상세계 서버는 가상세계의 VRML 데이터를 분할공간(region) 단위로 분할하여 보관하며 참여자의 위치 정보에 맞는 공간 식별자를 전송 받아 필요한 분할공간 데이터를 전송한다.

중계 서버는 가상세계 클라이언트에서 전송된 참여자의 위치 및 행위 등과 같은 이벤트 정보를 연관된 타 클라이언트에게 중계하며 클라이언트가 필요로 하는 분할공간 데이터를 전송한다.

가상세계 클라이언트는 참여자의 위치와 행위 정보를 중계 서버에 전송하고, 중계서버로부터 전송된 타 참여자의 변경된 정보를 수신하여 변화된 정보를 갱신하여 렌더링한다. 또한 가상 공간을 렌더링하여 브라우저에 반영한다.

3.1 아바타 행위 유형

가상공간에서 참여자는 가상세계를 이루는 구성원이 된다. 참여자들이 서로의 존재를 인지하려면 상대방의 모습을 볼 수 있어야 하는데, 참여자의 가상적인 모습인 아바타를 통해 참여자의 위치나 행동을 타 참여자에게 나타낼 수 있다.

가상공간상에서 아바타의 행위에는 향해, 행동, 대화, 공간이동 등의 행위가 있다. 네비게이션(navigation)은 아바타의 좌표 이동에 관한 이벤트로 인한 행위이다. 행동(activity)은 가상 공간상의 타 객체에 영향을 미치지 않는 범위 내에서 일어나는 인사, 춤추기, 손 흔들기, 점프 등과 같은 아바타 자체의 이벤트를 의미한다. 대화(conversation)는 참여자가 아바타를 통하여 다른 참여자와 대화하는 행위를 나타내며, 공간 이동(space movement)은 다른 가상세계로의 이동을 의미한다. <표3.1>은 이 행위를 분류한 표이다.

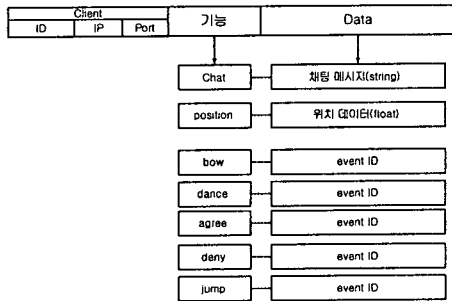
<표3.1> 행위의 종류

종류	내용
항해	walk, rotate, pan, fly
공간이동	가상공간 상의 특정 위치또는 다른 가상공간으로 이동
대화	타 참여자의 아바타 또는 자율객체와의 대화
행동	인사, 춤추기, 손 흔들기, 동의, 거절, 점프

3.2 아바타 정보 전송

본 절에서는 설계한 분산 가상현실 시스템에서의 아바타의 행위 정보를 타 클라이언트에게 전송하고 공유하는 방법에 대해 서술한다.

분산 가상현실 시스템에서는 가상의 공간에 다수의 참여자가 동시에 참여하게 된다. 이때 각 참여자는 정적인 공간상에 계속 머무는 것이 아니라, 메시지 전송 및 위치 이동이다. 행위 공유 등과 같은 참여자간의 상호작용에 의한 동적인 행동을 발생시키기 때문에 각 사용자간에는 표현의 공유가 가능하여야 한다. 따라서 이러한 아바타들의 공유와 일관성 유지를 위하여 일정한 형식을 구성하는 아바타 전송 프로토콜이 정의되어야 한다.



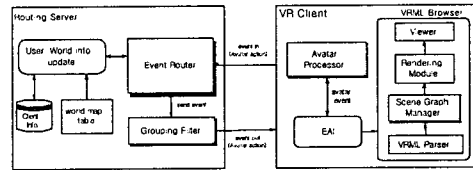
<그림3.2> 이벤트 전송 패킷의 구조

<그림3.2>는 아바타 행위 및 위치 정보 전송을 위한 패킷을 나타낸 것이다. Client ID 필드는 이벤트를 발생시킨 참여자의 클라이언트 식별자를 나타낸 것이며, 기능 필드는 아바타의 행위에 대한 기능을 나타내고 데이터 필드는 각 기능에 대한 값을 나타낸 것이다.

참여자가 아바타를 통하여 이벤트를 발생시키면, 가상세계 클라이언트는 이 아바타 이벤트를 Avatar Processor에서 클라이언트의 정보 및 이벤트의 기능과 데이터를 통신 패킷으로 묶어 중계 서버로 전송한다. 전송된 이벤트는 Event router를 통해 Grouping Filter로 보내지게 되며, Grouping filter를 통하여 아바타의 이벤트 정보는 클라이언트 목록 모듈을 참조하여 동일한 가상공간을 공유하는 타 클라

이언트에게 이벤트 정보를 전송한다. 이렇게 전송된 이벤트 정보는 Avatar Processor에서 이벤트 정보에 해당하는 이벤트 노드를 얻어와 EAI를 통해 Scene Graph Manager에서 scene graph를 갱신하며, 갱신된 정보는 렌더링 모듈을 통해 가시적으로 표현된다.

<그림3.3>는 중계 서버와 참여자들간의 이벤트 중계 과정을 나타낸 것이다.



<그림3.3> 아바타 이벤트 중계 과정

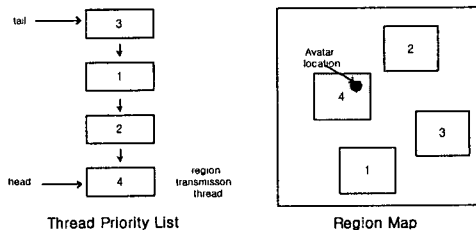
3.3 점진적 데이터 전송 시스템

본 절에서는 기존 분산 가상현실 시스템의 단점을 보완하기 위해 가상공간을 분할공간(region) 단위로 세분화하여 전송하는 방법을 설명한다.

3.3.1 점진적 데이터 전송

가상세계의 VRML 파일을 효율적으로 전송하기 위하여 3차원 가상공간을 분할공간 단위로 세부 분할하여 전송함으로써 전송 속도와 렌더링 성능을 향상할 수 있다.

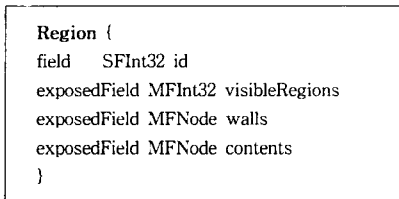
<그림3.4>는 가상공간 데이터를 점진적으로 전송하기 위한 방법을 나타낸 것으로, region map은 가상세계를 구성하는 가상공간을 분할공간 이라는 작은 공간 단위로 분할하여 관리한다. 클라이언트에서 전송된 아바타의 위치 정보는 region map을 참조하여 분할공간과 아바타 사이의 위치를 계산한다. 분할공간과 아바타 사이의 위치 계산은 가까운 분할공간의 전송 스레드를 스레드 우선순위 리스트(thread priority list)의 앞쪽에 위치시킨다. 이로써 아바타의 위치에서 꼭 필요한 분할공간을 먼저 받을 수 있게 하여 사용자의 기다리는 시간을 최소화 한다.



<그림3.4> 점진적 데이터 전송 방법

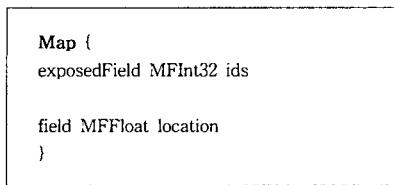
3.3.2 노드의 확장

분산 가상현실 시스템에서 참여자가 꼭 필요로 하는 가상공간 데이터를 전송하고 클라이언트의 렌더링 성능을 향상하기 위하여 VRML의 54개 노드에 분할공간의 개념을 지원하는 새로운 노드인 Region node와 이를 지원하는 Map node를 정의하였다.



<그림3.5> Region 노드의 정의

<그림3.5>는 Region 노드를 정의한 명세를 보이고 있다. Region 노드는 가상 세계를 이루는 VRML 파일을 임의의 분할공간 단위로 분할한 것으로 각 분할공간을 식별하기 위한 식별자와 그 분할공간과 함께 렌더링될 분할공간 식별자(region ID), 분할공간의 영역 경계를 표시하는 wall, 분할공간에 포함될 객체를 나타내는 content로 구성되어 있다.



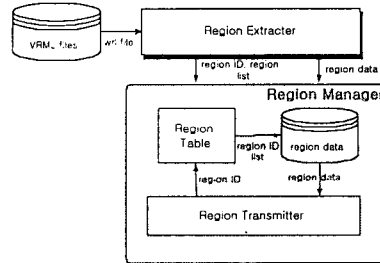
<그림3.6> Map 노드의 정의

<그림3.6>은 Map 노드의 명세를 나타낸다. Map 노드는 앞에서 정의한 Region 노드의 정보를 담고 있는 노드로써, 가상세계를 구성하는 분할공간들의 식별자를 나타내는 분할공간 식별자와 각 분할공간의 위치를 나타내는 location 필드로 구성되어 있다.

본 논문에서는 이러한 Region 노드와 Map 노드의 확장을 통해 분산 가상현실 시스템에서의 점진적 데이터 전송을 지원하는 시스템에 대해 설계하였다.

3.3.3 분할공간 관리

<그림3.7>은 가상세계를 이루는 VRML 파일을 region 단위로 분할하는 과정은 <그림3.7>과 같다.

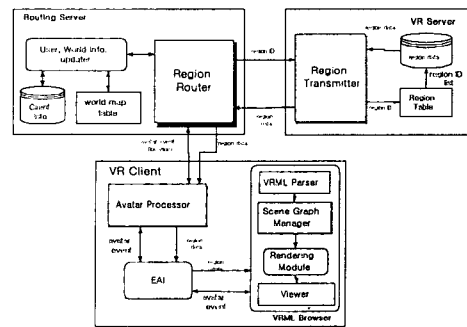


<그림3.7> Region Manager의 처리 과정

먼저 가상세계를 이루는 VRML 파일은 Region Extractor를 거쳐 Region 노드 단위로 가상공간의 장면을 분리하며, 또한 분할공간과 연관된 분할공간의 리스트를 추출한다. 이렇게 추출된 분할공간들의 데이터는 분할공간 데이터에 저장되고, 분할공간과 연관된 분할공간 리스트의 목록은 분할공간 테이블에서 저장하여 관리하게 된다. Region Transmitter는 중계서버에서 분할공간 식별자 요청 시 테이블을 참조하여 요청된 분할공간과 그것을 포함하는 분할공간 리스트에 맞는 분할공간 데이터를 추출한다.

3.3.4 정보 전송 프로토콜

본 논문에서 제안한 분산 가상현실 시스템에서 점진적 데이터 전송처리과정은 <그림 3.8>과 같다.



<그림 3.8> 점진적 데이터 전송 처리 과정

참여자가 가상 공간에 접속하면, 클라이언트는 아바타의 이벤트 정보를 중계 서버에 전송한다. 중계서버의 Region Router 모듈은 분할공간의 목록, 각 분할공간의 위치 정보를 가지는 world map table과 클라이언트에서 전송된 아바타의 위치 정보를 참조하여 아바타의 위치에 적합한 분할공간 식별자를 추출한다. 이렇게 추출된 분할공간 식별자는 가상세계서버의 Region Manager에 있는 Region Transmitter에 보내진다.

Region Transmitter에서는 전송된 분할공간 식별

자는 분할공간 테이블에서 분할공간의 리스트 식별자를 추출하며, 이를 이용하여 분할공간 데이터를 추출하게 된다. 이렇게 추출된 분할공간 데이터는 중계 서버를 거쳐 가상세계 클라이언트에 전송된다.

본 논문에서는 가상세계를 이루는 VRML 파일 전체를 클라이언트에 전송하는 방법이 아닌 분할공간 단위로 분할하여 아바타의 위치에 적절한 가상공간 데이터를 클라이언트에 전송하게끔 하였다.

3.4 기존 시스템에의 적용

본 논문에서 구현한 분산 가상현실 시스템의 클라이언트로 Microsoft사의 Internet Explorer 5.0 한글 버전 또는 Netscape사의 Communicator 4.7 한글 버전에서 VRML 브라우저로 VRML 컨소시엄내의 그룹인 Java3D&VRML의 VRML97Player를 사용하였다.

클라이언트 프로그램의 화면구성은 상단에 VRML 브라우저가 있고 하단에 자바 애플릿이 위치하게 된다. <그림 3.9>는 클라이언트의 수행 모습이다. 서버에 로그인한 클라이언트는 VRML 브라우저를 통해 네비게이션을 하게 된다.



<그림 3.9> 클라이언트의 실행 장면

4. 결론

본 논문에서는 분산 가상현실 시스템에서 가상공간을 분할공간(region) 이라는 공간 단위로 세부 분할하기 위한 방법을 연구하였으며, 또한 아바타의 행위 정보를 다 참여자에게 전송 시 앞에서 제시한 분할공간이란 개념을 이용하여 같은 분할공간에 속한 참여자에게만 전송하기 위한 방법을 연구하였다.

본 시스템은 가상세계 서버에서 분할된 공간 데이터를 분할공간 단위로 클라이언트에 전송함으로써 전송 속도의 향상과 클라이언트의 렌더링의 속도 향상을 보여 보다 실세계와 유사한 분산 가상현실을 서비스할 수 있는 장점이 있다. 또한 관심영역에 속한 아바타의 행위 정보만을 전송함으로써 인터넷상에서 실시간으로 아바타의 위치에 적절한 가상공간

데이터의 전송과 이벤트 전송을 지원하는 시스템이다.

향후 사용자가 동시에 사용하는 대용량 시스템에서의 동기화 문제는 고려하지 않았다. 향후 이러한 동기화 문제를 처리하기 위한 연구와 더불어 Region node와 Map node와 같은 노드 확장을 통해 생성한 3차원 가상공간을 쉽게 구축할 수 있는 개발 환경 또는 도구에 관한 연구를 하고자 한다.

참고문헌

- [1] Jed Hartman, Josie Wernecke, *The VRML 2.0 Handbook*, Addison-Wesley, 1996.
- [2] Andrea L.Ames, David R. Nadeau, John L. Moreland, *VRML 2.0 Sourcebook*, Wiley, 1997.
- [3] J. W. Barrus, R. C. Waters, and D. B. Anderson, *Locales: Supporting Large Multi-user Virtual Environments*, IEEE Computer Graphics and Applications, pp. 50-57, Nov. 1996
- [4] Wolfgang Broll "Populating the Internet : Supporting Multiple Users and Shared Applications with VRML", Jurgen Fechter, 1997.
- [5] Mike Wray, Rycharde Hawkes, "Distributed virtual environments and VRML : an event-based architecture", Computer Networks and ISDN Systems, pp. 43-51, 1998
- [6] 허도영 외, "3차원 가상공간에서의 Multi-Avatar 중계 시스템의 설계 및 구현," 추계멀티미디어 학회 추계 학술논문집 제 1권 제 2호, 1998.11.
- [7] Community Platform, Multi-User VR Browser, Blaxxun Interactive.
<http://www.blaxxun.com/>
- [8] "Distributed Interactive Simulation" DIS-Java-VRML Working Group
<http://www.web3d.org/WorkingGroups/vrtp/dis-java-vrml/>
- [9] VRML-Java3D
<http://www.vrml.org/WorkingGroups/vrml-java3d/>
- [10] History of the VRML Specification
<http://www.vrml.org/about/historyofvrml.htm>