

분산 환경에서의 가상현실 서비스를 위한 클라이언트/서버 시스템의 설계

A Design of Client/Server System for Virtual Reality Service in Distributed Environments

황대훈*

1. 서 론

오늘날 인터넷 상에서 텍스트를 포함한 그림 데이터와 같은 시각적인 정보의 전송 및 표현이 가능해 지고 HTML을 기반으로 하는 문서 표현 방식과 하이퍼미어 방식의 정보 연결 구조를 채택함에 따라, 웹(web)의 활성화를 촉발하는 계기가 되었다. 그러나 기존의 웹에서의 정보 표현 방법은 2차원 평면 정보에 의존함으로써, 보다 효율적인 정보의 표현에 한계가 있어 왔다.

이에 웹 상에서 3차원 정보를 표현함으로써 사용자에게 몰입감과 입장감을 주어 좀더 현실감 있는 표현을 하고자 하는 요구가 제기되었다. 이를 위해 웹 상에서 현실 세계의 3차원 정보를 있는 그대로 표현할 수 있는 방법에 대한 연구가 진행되어 왔고, 그 결과 VRML 1.0을 거쳐 오늘날 VRML 2.0이 발표되기에 이르렀다.

그러나 이러한 VRML 2.0의 지원 형태는 VRML으로 구축된 가상공간을 브라우저 상에서 3차원으로 표현하고 사용자는 단순히 이를 확인하는 수준에 머무르고 있다. 이에 3차원 가상공간 상에서 다수의 참여자(participant)가 네트워크를 통하여 동시에 의사소통 및 상호작용을 할 수 있도록

개선되어야 할 필요성이 대두되었다.

이와 같이 네트워크를 통한 다중 참여자들 간의 상호작용에 기반을 두어 3차원 가상현실의 체험이 공유되는 가상의 공간을 가상공간이라고 하며, 가상공간에서 참여자는 가상세계(cyberworld)를 이루는 구성원이 된다[1]. 이때 참여자들이 서로의 존재를 인지하려면 상대방의 모습을 볼 수 있어야 하는데, 이렇게 상대방을 볼 수 있도록 참여자의 모습을 3차원으로 표현한 것을 아바타(avatar)라 한다[3]. 아바타는 가상세계에서의 참여자의 가상적인 모습으로서, 참여자는 이 아바타를 통하여 자신의 행동 및 모습을 타 참여자에게 나타낼 수 있다.

이러한 3차원 가상공간상에서 참여자들이 네트워크를 경유하여 가상공간을 공유하고 실시간에 상대방의 위치와 행위를 인지하고 의사 전달을 하며 공동작업도 할 수 있는 가상현실 기술을 분산 가상현실(distributed VR) 또는 네트워크를 통한 가상현실(networked VR)이라고 한다[8,9]. 분산 가상현실 시스템은 가상공간에 참여하는 참여자수의 증가 및 이에 수반되는 대규모 가상공간에 대한 공간 관리와 아바타 관리 및 상호 작용 등의 처리를 가상공간 상의 참여자에게 실시간으로 지원한다.

이에 본 논문에서는 VRML 언어로 표현된 3차원 가상공간을 분할하여 관리하기 위한 가상세계

*경원대학교 컴퓨터공학과 교수
본 연구는 2000년도 경원대학교 학술연구비의 지원을 받아 이루어 졌음

서버(VR server)와 참여자들간의 상호작용 이벤트를 중계하고 공간 데이터를 클라이언트에 전송하기 위한 중계 서버(routing server), 그리고 이들 서버와 통신하는 가상세계 클라이언트(VR client)를 제안한다.

2. 관련 연구

이 장에서는 가상현실의 유형, VRML의 배경 및 구조, EAI의 개념 및 기능에 대한 소개, 그리고 분산 환경을 위한 통신망의 연결 구조, 끝으로 기존의 분산 가상현실 시스템의 유형 및 사례 등에 대해 소개한다.

2.1 가상현실

넓은 의미의 가상현실(virtual reality)이란, 사용자에게 몰입감을 부여하고 풍부한 상호작용이 가능하며 다중의 감각적 피드백을 제공하는 가상적 환경을 컴퓨터를 이용해 생성하는 제반 기술을 포함하는 개념이다[2]. 즉, 사용자의 몸짓, 이동 등을 감지하고 그에 따라 적절한 환경 변화 및 객체 반응을 시각, 청각 등 여러 가지 감각을 통한 피드백으로 전달해 줌으로써 사용자로 하여금 가상의 환경을 체험하도록 지원하는 제반 기술이라 하겠다.

가상현실은 하드웨어를 기반으로 하는 몰입형과 제3자형이 있으며, 인터넷을 기반으로 하는 데스크탑형이 있다.

첫째, 몰입형 가상현실(full immersion VR)은 하드웨어를 기반으로 하는 가상현실로, 이러한 유형에는 HMD(Head-Mounted Display), 3D Audio, Vision Station 등이 있다. HMD는 장착되어 있는 두 개의 모니터를 통해 마치 현실세계를 보는 것처럼 볼 수 있게 해주며, 또한 Vision Station은 반구 스크린을 통해 3차원 공간을 렌더링(rendering)

하는 방식으로 자연스러운 시각적 몰입 효과와 다양한 가상 체험효과를 준다[10].

둘째, 제3자형 가상현실은 신체 전체의 움직임을 추적하여 신체 동작이나 몸의 위치를 나타내는 데이터 슈트(data suit)와 장착되어 있는 센서들이 데이터를 컴퓨터로 전달해 줌으로써 가상공간에 존재하는 물체를 만지기도 하며 형태를 변형시켜주기도 하는 데이터 글러브(data glove), 그리고 크로마키 무대 안의 피사체를 카메라로 잡고 그 움직임을 실시간으로 분석하여 가상환경에 합성하는 Mandala VR 등이 있다[10].

끝으로, 데스크탑형 가상현실(desktop VR)은 3차원 데이터를 인터넷을 통해 가시화하고 타인과 대화하며 자신만의 공간을 만들고, 관심있는 공간으로 이동하면서 가상현실을 체험하는 것을 말한다[4,5]. 이러한 데스크탑형 가상현실은 VRML의 출현으로 PC 상에서 가상현실을 체험할 수 있는 보다 일반적인 기술로 자리잡고 있다. 오늘날 VRML 2.0은 VRML 브라우저 상에서 마우스 입력 등의 이벤트를 정의하고, VRML 2.0 파일에 포함된 자바(Java), TCL 등의 인터프리터 언어로 이를 처리함으로써 가상환경 상의 객체가 스스로 혹은 참여자에 반응하여 움직일 수 있도록 지원하는 수준에 이르고 있다.

2.2 VRML

2.2.1 VRML의 배경

VRML은 Virtual Reality Modeling Language의 약어로 3차원 물체(object)들의 상호작용과 3차원 세계를 웹 상에서 경험할 수 있도록 기술하는 언어이다[11]. VRML은 어떤 다른 언어보다 HTML과 유사하여, HTML처럼 VRML도 파일에 ASCII 포맷을 사용하며 브라우저에 의해 해석되어지는 서술 언어이다. HTML은 그 자체의 한

계로 인하여 제공되는 정보에 역동적인 요소를 추가하고자 하면 다른 어플리케이션을 사용할 수 있는 별도의 인터페이스를 설치하여야 한다. 그러나 VRML은 그 자체가 역동적인 환경을 제공하기 때문에, HTML과 같이 페이지의 형식을 설정하는 대신 3차원 환경과 물체의 형태를 설정한다[1].

1994년 11월 Silicon Graphics사의 Open Inventor를 토대로 한 VRML(Virtual Reality Modeling Language)이 등장한 이래, 현재 VRML은 웹을 통해 3차원 정보를 표현하기 위한 표준으로 자리 잡았으며, 이 VRML을 통해 현실 세계를 웹 상에서 3차원으로 표현할 수 있다[1].

초기의 VRML 1.0은 웹에 3차원 공간을 구축하거나 3차원 정보를 서로 교환하는 등의 뛰어난 3차원 그래픽을 구현할 수 있으나, VRML 1.0은 객체들을 움직이거나 스스로 움직이는 애니메이션 등이 불가능한 정적인 공간만을 표현하였다 [1,2]. 이에 동적인 데이터의 표현과 참여자 간의 상호작용을 지원하는 VRML 2.0이 발표되었으며, 라우트 노드(Route node)와 스크립트 노드(Script node)를 통해 객체와 행위를 연결하기 때문에 동적인 장면(scene)의 조작이 가능하게 되었다[2].

현재 VRML 2.0은 ISO의 JTC 1(Joint Technical Committee 1)과 VRML 콘소시엄과 파트너쉽을 맺고 있는 IEC에 의해 국제 표준(ISO/IEC 14772)인 VRML 97로 제정되어 있다[19].

2.2.2 VRML의 구조

VRML 2.0의 파일은 헤더, 노드, 필드 및 문장 등으로 구성된다. 이들을 포함하고 있는 파일은 브라우저에 의해서 처리되고 표현되며, 사용자와 상호작용을 하게 된다[2]. 그림 1은 간단한 VRML 파일의 소스 예이다.

VRML 2.0에서는 54개의 노드를 지원하고 있는데, 이러한 VRML 2.0이 지원하는 전체 노드들

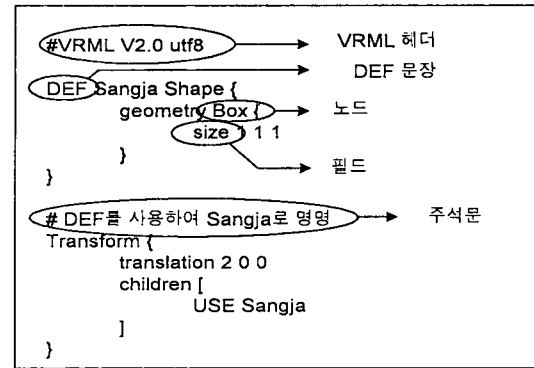


그림 1. VRML 문서 구조의 예

을 크게 분류하면 Geometry 노드, 그룹 노드, 센서 노드 및 Interpolator 노드로 나눌 수 있다.

Geometry 노드는 물체를 표현하려 할 때 사용하는 노드로 항상 Shape 노드의 geometry 필드에 올 수 있는 노드를 통틀어 Geometry 노드라고 한다. 그룹(group) 노드들은 하위에 노드 리스트들을 포함할 수 있도록 children이라는 필드를 가지고 있다. 또한 scene graph를 구성하고 각 geometry에 다양한 특성을 부여하기 위해서도 사용된다. 센서(sensor) 노드는 마우스의 움직임, 위치 이동, 시점 이동 등과 같이 관찰자의 동작에 따라 이벤트를 발생시킬 때 사용한다. Interpolator 노드는 센서 노드와 함께 선형 키프레임 애니메이션을 만드는데 사용된다.

이밖에 스크립트 노드와 사운드, 조명, 배경 등에 관련된 노드들이 있다.

필드는 VRML 파일의 노드 문장 안에 위치하여 그 값을 가지며, 가상공간의 상태를 정의하기 위하여 사용되는데, 노드의 특성을 나타내기 위한 참/거짓 혹은 정수나 실수의 값 등으로 표현되는 부분이 바로 필드이다. 모든 필드는 16개의 필드 타입(field type)중 어느 하나에 속하게 되는데, 필드 타입은 하나의 값을 갖는 종류와 두개 이상의 값을 갖는 종류가 있다.

2.2.3 Scene graph의 구조

Scene graph는 객체와 객체간의 관계 및 특성들을 설명하는 노드들로 구성되어 있다. 여기서는 객체들을 표현하기 위해서 사용되는 청각적이고 시각적인 기능을 가지는 노드들뿐만 아니라, 사건을 발생시키고 전달하는 매커니즘에 참여하는 노드들까지도 포함된다. 이들은 scene graph 안에서 계층적인 구조를 가지고 연결된다.

그림 2는 VRML 파일에 대응하는 scene graph의 구조를 나타낸 것이다.

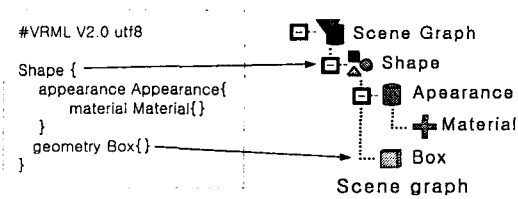


그림 2. Scene graph의 구조

2.3 EAI

EAI(External Authoring Interface)는 SGI사의 Chris Marrin이 1997년에 제안[18]한 것으로, VRML 브라우저와 자바 애플릿(Java applet)과 같은 외부 세계와의 인터페이스로 외부 어플리케이션과의 통신을 지원한다.

VRML로 제공되는 가상환경은 그 환경의 용도와 문맥에 적합한 사용자 인터페이스를 필요로 한다. 이러한 인터페이스를 VRML 환경 내에 삽입하는 것은 부적절하며 사용자들은 스크롤 바나 버튼 같이 보다 익숙한 인터페이스를 선호할 것이다. 이러한 인터페이스를 제공하는 자바 애플릿과 같은 외부 프로그램을 통해 가상세계를 조작하기 위해 EAI가 제안되었다.

EAI를 통해 외부 프로그램은 VRML 브라우저가 가지고 있는 scene graph의 계층 구조를 변화

시키거나 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. EAI는 현재 VRML 2.0 명세의 일부는 아니지만 "External Interface for VRML 2.0" 문서에 채택 가능하다고 되어 있다[15].

현재 EAI를 지원하는 VRML 2.0 VRML 브라우저로는 Sony사의 Community Place, Silicon Graphics사의 Cosmo Player, Blaxxun Interactive사의 Contact 등이 있다. 현재 자바로 구현된 EAI가 있으며 이것은 그림 3과 같이 자바 애플릿이 VRML 브라우저와 EAI를 통해 인터페이스 할 수 있도록 해준다[17].

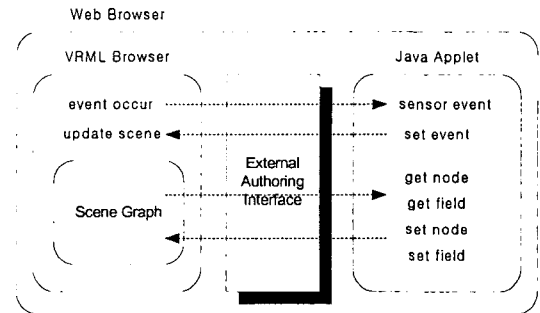


그림 3. EAI를 통한 VRML 브라우저와 자바 애플릿 간의 통신

2.4 분산 시스템의 연결 구조

분산 시스템을 위한 클라이언트와 서버의 연결 구조는 서버의 개수, 클라이언트와 서버의 연결 형태에 따라 중앙 집중형, 완전 분산형, 복합형, 혼합형, 다중 서버-다중 클라이언트 유연형 등으로 나눌 수 있다[8].

2.4.1 중앙 집중형과 완전 분산형

중앙 집중형 구조는 중앙의 서버가 모든 통신

의 중계 역할을 하는 구조로 일반적으로 클라이언트/서버 구조로 알려져 있다. 이 구조에서는 서버가 시스템 전체의 정보를 관리하고 모든 통신을 중계하므로 제어하기 쉽고 일관성을 유지하기가 용이하다. 그러나 서버에 처리 부담과 통신 부하가 가중되므로 서버가 병목점으로 작용하는 단점이 존재한다. 이러한 병목현상에 의한 지연은 가상환경에서 사용자가 공동작업을 할 때 요구되는 실시간성을 보장할 수 없게 한다.

완전 분산형의 구조는 모든 클라이언트가 서로 연결을 해서 데이터를 주고받고 제어를 하는 구조이다. 이 구조에서는 서버의 병목점을 제거할 수 있으나 다수의 클라이언트 사이에 발생할 수 있는 동시성 제어(concurrency control)나 클라이언트 간에 공유되는 정보의 제어가 어렵고 시스템 일관성을 유지하기 힘들다.

2.4.2 복합형

복합형 구조는 그림 4와 같이 중앙 집중형 구조와 완전 분산형 구조가 절충된 구조로 제어나 일관성을 유지하기 위한 데이터와 무손실을 요구하는 데이터는 점선으로 표현된 중앙 집중형 구조로 전송하고 각각의 클라이언트간에 손실 가능한 데이터의 전송은 실선으로 표현된 완전 분산형 구조처럼 전송하는 구조이다.

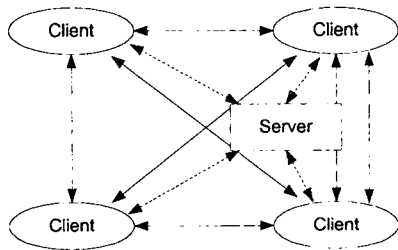


그림 4. 복합형

2.4.3 혼합형

혼합형 구조는 그림 5와 같이 새로운 네트워크

모델로서 기존의 중앙 집중형 구조와 완전 분산형 구조의 단점을 일부 보완한 것으로 서버로부터 공유된 가상세계와의 접속을 위해 기존의 클라이언트-서버 방식으로 접속하고, 서버간에는 peer-to-peer 형태의 완전 분산 구조를 채택하고 있다. 본 시스템은 이러한 혼합형 구조를 취하고 있다.

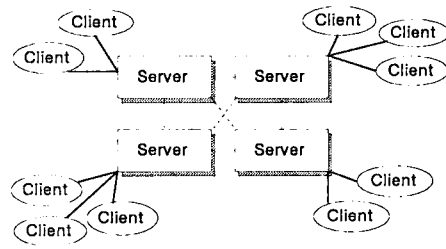


그림 5. 혼합형 구조

2.4.4 다중 서버-다중 클라이언트 유연형

다중 서버-다중 클라이언트 유연형은 그림 6과 같이 구성 요소로서 각 참여자를 나타내는 클라이언트, 클라이언트간의 중재자 역할을 하는 서버, 몇 개의 인접한 서버들이 한 그룹으로 묶여진 단위인 풀(pool) 등이 존재한다.

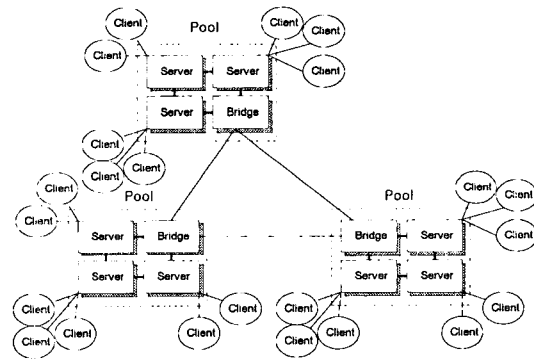


그림 6. 다중 서버-다중 클라이언트 유연형

그리고 각각의 풀들은 브리지(bridge)라고 불리는 서버들로 묶어지며, 가상세계, 참여자, 협동

작업 등의 관리를 담당한다. 이는 서버에서 발생하는 병목현상을 제거할 수 있으나 한 클라이언트에서 다른 서버가 관리하는 클라이언트로 메시지를 전달할 때 여러 서버 계층을 통과하므로 정보 전송 지연 문제가 발생할 수 있다[6,7,8].

본 연구에서는 이상과 같이 각 통신망 연결구조의 장단점을 고려하여, 중앙 집중형 구조와 완전 분산형 구조의 단점을 보완하여 많은 참여자를 포함할 수 있는 혼합형 구조를 본 연구의 분산 환경에서의 가상현실 서비스를 위한 클라이언트/서버 시스템의 모델로 채택하였다.

2.5 가상현실 서비스 시스템의 유형

본 절에서는 가상현실 서비스 시스템에 대하여 간단히 개념적 구성을 알아본다. 가상현실 시스템은 단일 사용자 시스템과 다수 사용자 시스템으로 구분할 수 있다. 또한 참여자와 가상환경의 상호작용 정도에 따라 단순한 가상현실 시스템과, 자율적 움직임이나 조작 등이 가능한 동적 가상현실 시스템으로 크게 구분할 수 있다.

다음은 이들의 개념적 구조에 대하여 간단히 설명한 것이다.

2.5.1 단일 사용자 가상현실 시스템

일반적인 단일 사용자 가상현실 시스템은 그림 7과 같이 구성된다.

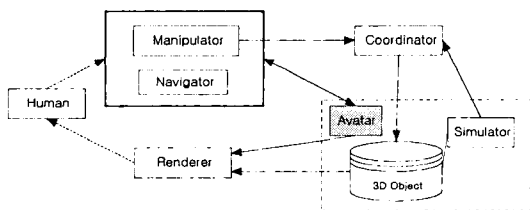


그림 7. 단일 사용자 가상현실 시스템

그림 7에서 점선으로 둘러싸인 가상세계 데이

터베이스는 응용과 목적에 따라 다양한 구성과정의가 가능하며, 일반적으로 시뮬레이션과 렌더링이라는 2가지 측면에서 볼 수 있다. Navigator는 입력장치의 값을 읽어서 사용자의 시점을 계산하는데, 사용자 시점의 이동은 가상환경과의 충돌 검사를 통하여 범위를 제약받기도 한다. Renderer는 사용자의 시점을 기준으로 3차원 모델들을 가시화 하여 사용자에게 제공한다[9].

2.5.2 단순 가상현실 시스템

단순 가상현실 시스템은 다수의 참여자가 동시에 같은 공유 공간에 접속하여 가상환경을 둘러보고 서로 상대방을 인지하는 것을 지원한다. 이는 단일 시스템에서도 가능하나 다수 참여자를 대상으로 하는 경우 일반적으로 LAN과 WAN에 분산된 네트워크 시스템으로 가정한다.

공유 환경은 일반적으로 각 참여자 노드에 복제된다. 이는 실시간 렌더링을 위하여 필수적이다. 여기서서는 각 노드는 자신의 아바타만을 변화시킬 수 있고 다른 아바타나 공유 환경에 대한 갱신은 없는 시스템이다. 따라서 동일 데이터에 대한 동시 조작은 발생하지 않는다.

그림 8과 같이 아바타를 제외한 3차원 모델은 복제 후 변경되지 않으므로 렌더링을 효율적으로 수행하기 위한 형태로 구성되는 것이 일반적이다[9].

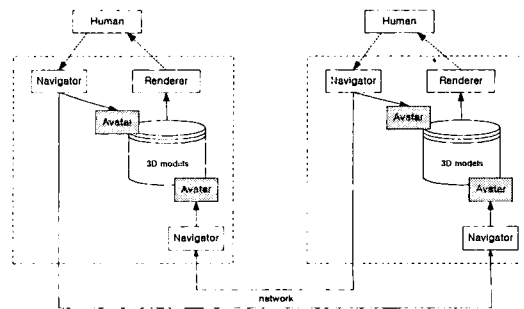


그림 8. 단순 가상현실 시스템

2.5.3 동적 가상현실 시스템

동적 가상현실 시스템은 단순 가상현실 시스템 기능에 더하여 참여자의 공유 가상환경에 대한 자유로운 갱신이 가능한 것이다. 그 개념적 구조는 다음 그림 9와 같다. Manipulator와 Coordinator를 제외한 부분은 앞의 단순 가상현실 시스템과 대동소이하다.

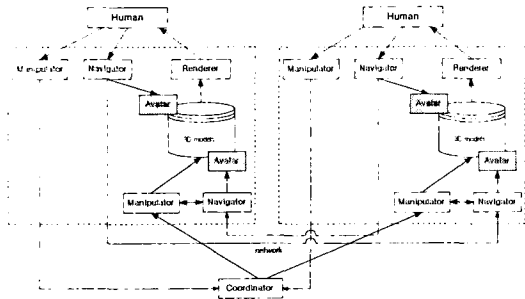


그림 9. 동적 가상현실 시스템

Manipulator는 참여자의 조작을 추출하고, 조작은 Coordinator에 의하여 중재된 후에 각 노드의 가상환경에 적용된다. 다수 참여자의 동시조작이 분산 복제된 가상환경에 가해지므로 각 참여자 노드에 복제된 가상환경의 일관성을 유지하는 coordination 방법이 필요하다.

가상세계의 분산 정도에 따라 하나의 가상세계 관리 서버를 중심으로 중앙 집중식으로 하는 방법으로부터 개개의 가상 객체가 각기 다른 프로세서를 가지고 이들이 서로 필요한 정보를 주고받는 완전히 분산된 경우까지 다양하다. 통신 방법에 있어서는 통신 서버를 통한 집중형, 두 프로세서 간의 unicast 통신에 의한 점대점(point-to-point) 통신과 한 프로세서에서 보낸 메시지를 동시에 하나 이상의 프로세서에서 수신하는 broadcast 혹은 multicast 통신 방식이 대표적이다[6].

본 연구에서는 이러한 분산 가상현실 서비스

시스템의 유형 중, 동적 가상현실 시스템 구조를 본 연구의 분산 환경에서의 가상현실 서비스를 위한 클라이언트/서버 시스템의 서비스 모델로 채택하였다.

2.6 시스템 사례

분산 가상현실은 3차원 가상공간을 참여자들이 네트워크를 경유하여 공유하고 실시간에 상대방의 위치와 행위를 인지하고 의사 전달을 하며 공동작업도 할 수 있는 가상현실의 새로운 연구분야이다.

분산 가상현실을 구현하기 위해서는 분산 가상현실이 네트워크 기반이라는 특징 때문에 기존의 가상현실과는 다르게 다음과 같은 문제점들을 해결해야 한다[12,15].

첫째, 확장성(scalability)에 관한 문제는 가상세계에 참여하는 참여자가 많아질수록 네트워크 트래픽과 서버의 오버헤드(overhead) 증가에 따른 것이다.

둘째, 영속성(persistency)에 관한 문제는 참여자가 가상세계에서 가상객체의 속성을 바꾸었다면 다음 번에 다시 가상세계에 참여 할 때에도 이 가상객체의 속성이 유지되어야 하는 것이다.

셋째, 일관성(consistency)에 관한 문제는 참여자가 가상객체에 행한 행위나 참여자의 위치가 인지 가능한 영역에 속해 있는 다른 참여자에 의해 인식되어야 하는 것이다.

넷째, 지역제(zoning)에 관한 문제는 가상세계를 작은 가상세계로 나눌 때 전체 가상세계들을 어떻게 관리하느냐와 어떻게 관계를 맺느냐에 관한 것이다.

다섯째, 지연 시간(latency)에 관한 문제는 참여자에게 높은 수준의 현실감을 제공하기 위해서 사건의 발생과 인지가 연속적으로 일어나는 자연

현상과 어떻게 유사하게 표현하느냐 하는 것이다.

기존의 분산 가상현실 시스템은 VRML를 사용하지 않은 가상현실 시스템과 인터넷 상에서 VRML을 기반으로 하는 분산 가상현실 시스템으로 분류할 수 있는데, VRML를 사용하지 않은 가상현실 시스템의 예로는 DIS, VEOS, DIVE, BrickNet 등이 있다.

DIS(Distributed Interactive Simulation)는 분산 가상현실의 개념을 실제 시스템에 응용하여 구축한 최초의 예로, 미 국방성에서 군사작전의 시뮬레이션을 위해 제작한 것이었다. 워싱턴 대학의 VEOS (Virtual Environment Operating Shell)는 몰입감을 주는 가상 환경의 응용 프로그램을 매우 보편적이고 일반적인 방법으로 지원한다 [16]. 스웨덴에서 개발된 DIVE(Distributed Interactive Virtual Environment)는 일관성을 유지하기 위해 데이터베이스의 복제와 점대점 통신을 사용하는 다중 참여자 가상환경 지원 시스템이다 [7]. BrickNet은 클라이언트/서버 모델을 사용하였으며, 서버의 병목현상을 제거하기 위해서 서버 시스템을 분산하여 각 서버 시스템들이 가상공간에 접속하는 참여자들을 나누어 관리하는 방식을 채택하고 있다[16].

인터넷 상에서 VRML을 사용한 분산 가상현실 시스템으로는 Blaxxun Interactive사의 Community Platform, Circle of Fire Studios사의 Active World, Sony사의 Community Place, Intel사의 IDMOO(Intel Distributed MUD Object Oriented) 등이 있다.

Blaxxun Interactive사의 Community Platform은 TCP/IP 프로토콜을 사용하며 URL을 통해 연결된 여러 개의 영역들로 구성된 가상세계를 제공한다. 이는 가상세계 서버인 Community Server와 클라이언트인 Contact로 이루어져 있으며, 통신 및 채팅을 비롯한 가상세계에 대한 여러 가지

부가 기능을 수행한다[13].

그림 10은 Blaxxun Interactive사의 Community Platform이다.

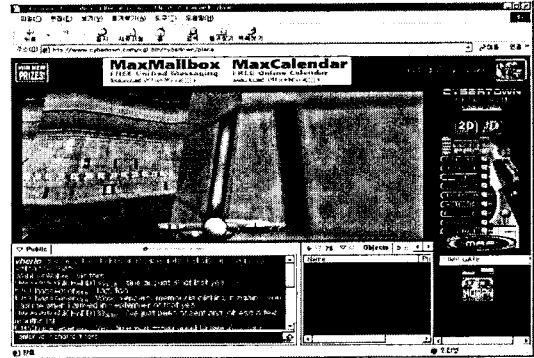


그림 10. Blaxxun Interactive사의 Community Platform

Sony사의 Community Place는 가상세계 서버인 Community Place Bureau와 클라이언트인 Community Place Browser로 이루어져 있다. Community Place Browser는 VRML 브라우저를 포함하고 있으며, 또한 웹 브라우저로 사용할 수 있다. 통신은 TCP 포트를 통해 이루어지며 사용자는 포트 번호를 변경할 수 있다[14].

Intel사의 IDMOO는 VRML로 구축된 세계에서 아바타를 통한 텍스트 및 음성 채팅을 지원하며, Intel이 Java로 쓰여진 객체 행위의 상호작용을 시험하기 위한 도구로서 개발되었다.

3. 시스템 설계

분산 가상현실 시스템은 물리적으로 떨어져 있는 다수의 참여자들이 가상환경의 전부 혹은 일부를 공유하고 이를 통하여 의사교환 및 상호작용할 수 있는 시스템이다. 따라서 시스템은 참여자들의 동시조작 등에 대하여 일관성을 유지하여야 하며, 참여자의 조작에 의한 변화를 다른 참여자

에 알려 실시간 상호작용이 가능하게 하여야 한다.

본 장에서는 웹 상에서 다수의 참여자들이 서로 상호작용을 할 수 있고, 3차원 가상세계의 데이터 전송 처리를 제공하는 분산 가상현실 시스템의 설계 방안을 제안한다. 우선 가상현실 데이터의 전송 처리를 위한 전체 시스템의 구성과, 아바타의 행위와 위치 정보를 전송 및 처리하는 과정, 가상현실 데이터 전송 처리 과정 및 프로토콜에 대하여 설명한다.

3.1 전체 시스템 구성

3차원 가상공간에서 다수의 참여자들이 가상공간을 공유하는 분산 가상현실 시스템의 구성요소는 크게 세가지로 나눌 수 있다.

첫째, 가상공간 및 참여자를 관리 및 운영하는 가상세계 서버와 둘째, 가상공간을 렌더링하여 가시적으로 표현하고 참여자와의 상호작용을 처리하는 가상세계 클라이언트, 그리고 마지막으로 클라이언트와 서버, 클라이언트와 클라이언트 간에 정보를 교환하고 중계하는 역할을 하는 중계 서버로 구성된다.

그림 11은 본 논문에서 제안하는 분산 가상현실 시스템의 전체 시스템 구성도를 나타낸 것이다.

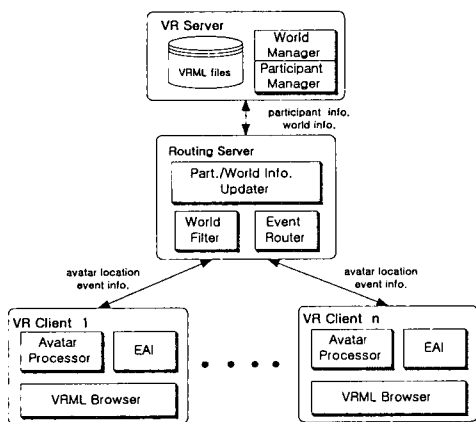


그림 11. 전체 시스템 구성도

3.2 가상세계 서버

가상세계 서버(VR server)는 가상세계의 VRML 파일을 가상세계(VR world) 단위로 구분하여 보관하며 참여자가 소속하기를 원하는 가상세계 식별자(world ID)를 전송 받아 필요한 VRML 파일을 전송한다.

그림 12는 가상세계 서버의 처리 과정을 나타낸 것으로서, 가상세계 서버에 접속한 참여자는 참여자 관리자(Participant Manager)에 의해 참여자 테이블(Participant Table)에 등록된다. 이때 참여자 관리자는 사용자 정보(User Info.)를 참조하여 참여자를 인증하며, 이 과정에서 참여자가 소속하기를 원하는 가상세계를 선택하도록 한다.

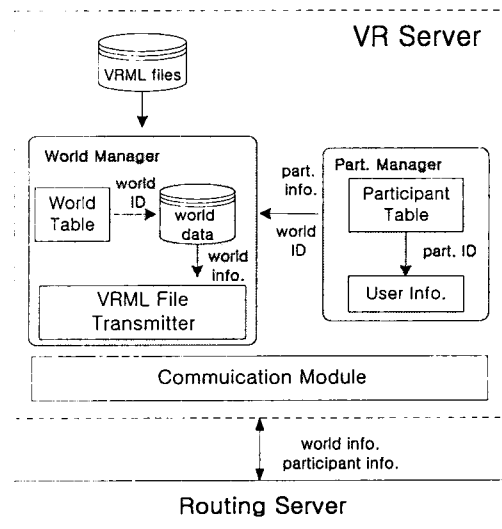


그림 12. 가상세계 서버

참여자에 의하여 가상세계가 선택되면, 참여자 관리자는 참여자 정보(participant info.)와 선택된 가상세계 식별자를 가상세계 관리자(World Manager)에게 전달한다. 이를 접수한 가상세계 관리자는 가상세계 테이블(World Table)을 참조하여 가상세계 구성을 위하여 필요한 VRML 파일을

포함하는 가상세계 정보(world info.)를 통신 모듈(Communication Module)에게 전달하고 통신 모듈은 이 정보를 가상세계 클라이언트에게 전송한다.

이와 동시에, 가상세계 관리자는 새로운 참여자가 가상세계에 접속하였음을 중계 서버에게 알려야 하는데, 이를 위하여 참여자 정보와 가상세계 정보를 통신 모듈에게 전달한다.

3.3 중계 서버

중계 서버(Routing server)는 가상세계 클라이언트에서 전송된 참여자의 위치 및 행위 등과 같은 이벤트 정보를 같은 가상세계에 속하는 타 클라이언트에게 중계하며 클라이언트가 소속되어 있는 가상세계를 관리한다.

그림 13은 중계 서버를 나타낸 것이다.

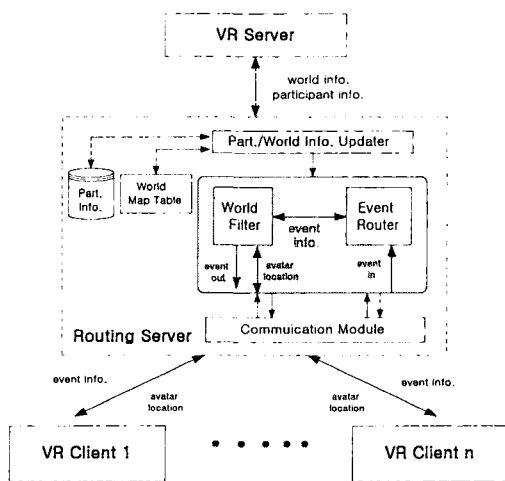


그림 13. 중계 서버

가상세계 클라이언트를 통해 전송된 참여자의 이벤트 정보 중 위치 정보가 중계 서버에 전달되면, 이 정보는 가상세계 필터(World Filter)에게 전달된다. 가상세계 필터는 이 정보가 어떤 가상

세계에 속해 있는 참여자로부터 전달된 것인지를 구분하여 참여자/가상세계 정보 갱신모듈(Part./World Info. Updater)에게 전달함으로써, 사용자 위치 정보는 가상세계 맵 테이블(World Map Table) 상에 갱신된다. 갱신된 위치 정보는 다시 동일한 가상세계 상에 존재하는 다른 모든 참여자에게 공유되어야 하는데, 이 과정도 앞에서와 같은 방법으로 진행된다.

통신 모듈을 통하여 가상세계 클라이언트에서 전송된 참여자의 행위 등과 같은 이벤트 정보는 이벤트 라우터(event router)를 경유하여 가상세계 필터에게 전달되며, 가상세계 필터는 이 정보가 어떤 가상세계에 속해 있는 참여자로부터 전달된 것인지를 구분하여 참여자/가상세계 정보 갱신모듈에게 전달함으로써 아바타 위치 정보와 같은 방법으로 처리된다. 물론 이 이벤트 정보는 앞에서의 방법의 역순으로 처리되어 동일한 가상세계 상에 존재하는 다른 모든 참여자에게 전달된다.

3.4 가상세계 클라이언트

가상세계 클라이언트(VR client)는 참여자의 위치와 행위 정보를 중계 서버에 전송하고, 중계 서버로부터 전송된 타 참여자의 변경된 정보를 수신하여 변화된 정보를 갱신하고 렌더링 하여 브라우저에 반영한다.

그림 14는 가상세계 클라이언트를 나타낸 것으로서, 참여자가 초기 접속 시에 가상세계 서버로부터 VRML 파일을 전송 받으면 VRML 브라우저(VRML Browser)의 Scene Graph 관리자(Scene Graph Manager)는 이 문서 파일을 파싱(parsing)하여 노드 정보를 추출하고 이를 기반으로 scene graph를 구성한다. 이 scene graph 정보는 렌더링 모듈(Rendering Module)에 전달되고 다시 뷰어(Viewer)를 통하여 가시적으로 표현된다.

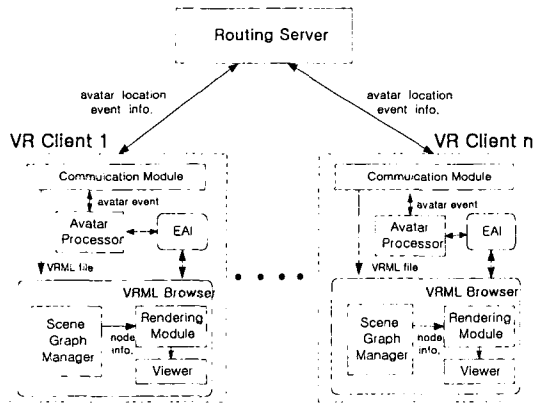


그림 14. 가상세계 클라이언트

그후 참여자가 외부 장치를 통해 이벤트 발생하면 이 정보는 EAI와 아바타 처리기(Avatar Processor)를 통해 중계 서버에 전송한다. 중계서버는 이를 가상세계 내의 타 참여자들에게 전송하게 되며, 이를 전송받은 타 참여자는 이 이벤트 정보를 EAI를 통해 VRML 브라우저에 반영하여 가시적으로 표현한다.

3.5 아바타 유형 및 행위

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. 아바타란 가상공간상에 가시화된 실제 참여자들의 대리인 또는 분신을 3차원으로 시각화한 것이다.

가상공간에 존재하는 객체는 아바타와 가상객체로 나눌 수 있는데, 아바타는 서비스의 주체가 누구냐에 따라 서버 아바타(server avatar)와 클라이언트 아바타(client avatar)로 구분할 수 있다. 예를 들면, 상점에서 물건의 위치를 묻는 아바타는 클라이언트 아바타이며 물건의 위치를 안내해주는 아바타는 서버 아바타로 분류할 수 있다.

또한, 자유도에 따라 자율 아바타(autonomous avatar)와 단순 아바타(animated avatar)로 구분할 수 있다. 자율 아바타는 자신의 기능이 컴퓨터 프로그램에 의해 기술된 객체로 가상공간을 스스로 항해하면서 특정한 업무를 수행하거나 참여자들의 편의를 제공한다. 길을 안내해주는 안내인 객체 등이 그 예가 될 수 있다. 단순 아바타는 아바타의 형태를 하고 있으나 동적 객체와 같이 단순 행위만을 반복하는 아바타를 말한다. 도어맨이나 가상공간상의 구경꾼 등이 그 예가 될 수 있다.

가상객체(virtual object)는 동적 객체와 정적 객체로 구분된다. 동적 객체(dynamic object)는 자율성은 없으나 참여자 또는 다른 가상객체의 이벤트에 의해 즉각적이면서 단순한 행위를 갖는 가상객체로 불을 켜기 위한 전등 스위치나 버튼을 누르면 열리는 문 등이 그 예이다. 정적 객체(static object)는 가상공간을 이루는 구성 요소로서 책상, 벽 같은 행위를 갖지 않는 3차원 객체이다.

그림 15는 아바타 간의 상호작용을 나타낸 것으로서, 아바타와 아바타 간의 다양한 형태의 상호작용과, 제한된 정보를 가지고 있는 자율 아바타와 아바타 간의 상호작용, 그리고 끝으로 아바타와 단순 아바타 및 동적 객체의 단순한 반복 행위만을 보이는 상호작용으로 구분할 수 있다.

가상공간 상에서 아바타의 행위(activity)에는 내비게이션, 공간 이동, 대화, 행동 등의 행위가

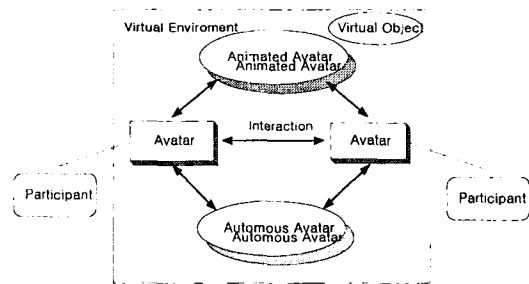


그림 15. 아바타 간의 상호작용

있다.

내비게이션(navigation)은 아바타의 좌표 이동에 관한 이벤트로 중력이 작용한 상태에서 바닥으로 작용하는 사물을 딛고 앞으로 이동하는 walk, 가상공간상의 객체의 중심을 기준으로 회전하는 rotate, 공간상의 수평이동을 위한 pan, 그리고 공간 상에서 중력 없이 지표면 위를 떠서 이동하는 fly 등이 있다.

공간 이동(space movement)은 다른 가상세계로의 이동을 의미하며, 대화(conversation)는 참여자가 아바타를 통하여 다른 참여자와 대화하는 행위를 나타낸다. 행동(activity)은 가상공간 상의 타 객체에 영향을 미치지 않는 범위 내에서 일어나는 인사, 춤추기, 손 흔들기, 점프 등과 같은 아바타 자체의 이벤트를 의미한다.

3.6 아바타 정보 전송 및 공유

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

3.6.1 이벤트 라우팅

아바타의 이벤트 정보가 이벤트를 발생시킨 노드로부터 다른 노드로 전파(propagation)되는 과정을 이벤트 라우팅(event routing)이라고 한다. 센서 노드(sensor node)들은 사용자 상호작용을 감지하는 시작점이 되며, 감지된 상호작용의 종류에 따라 적절한 이벤트를 타 VRML 노드들로 전파할 수 있는데, 이 과정을 나타내면 그림 16과 같다.

그림 16에서 Trigger는 애니메이션의 시작과 끝을 제어하기 위해 센서노드 등의 이벤트를 발생하는 노드를 의미한다. 이렇게 발생한 이벤트는 ROUTE 문에 의해 다른 노드에 중계되어 노드의 속성을 바꾸게 되고, 이로 인하여 이 노드의 속성

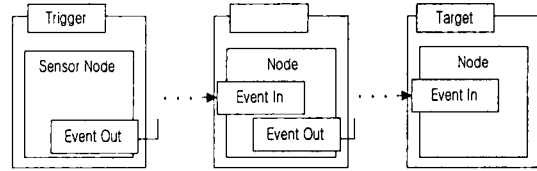


그림 16. 이벤트의 전파과정

이 바뀌면서 새로운 이벤트를 발생한다면 다른 노드로 계속하여 중계될 수 있다.

그림 17은 ROUTE 문을 이용한 경로 지정의 예를 나타낸 것으로서, 가상공간에 박스를 하나 만들고 이를 좌우로 왕복 이동시키는 VRML 문서의 예를 나타낸 것이다.

여기서 사용되는 노드 중에서 TimeSensor 노드와 PositionInterpolator 노드는 각각 애니메이션을 담당하는 노드이다. TimeSensor 노드는 미리 설정된 시간 간격으로 주기적인 이벤트를 발생

```
#VRML V2.0 utf8
Group
{
  children
  {
    DEF box1 Transform
    {
      children Shape
      {
        appearance Appearance
        {
          material Material{}
        }
        geometry Box{}
      }
    }
    DEF time TimeSensor
    {
      loop TRUE
    }
    DEF loc PositionInterpolator
    {
      key[0.0 0.5 1.0]
      keyValue[-2.0 0.0 0.0,
      0.0 0.0 0.0, 2.0 0.0 0.0]
    }
  }
  ROUTE time.fraction_changed TO loc.set_fraction
  ROUTE loc.value_changed TO box1.set_translation
}
```

그림 17. ROUTE 문을 이용한 경로지정의 예

시킬 수 있는 센서 노드이고, PositionInterpolator 노드는 시간의 변화에 따른 3차원 객체의 위치 이동을 설정하는 노드이다.

3.6.2 이벤트 정보 전송

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

그림 18은 아바타 행위 및 위치 정보 전송을 위한 패킷(packet) 구조를 나타낸 것이다. Client ID, IP, port 필드는 이벤트를 발생시킨 참여자 클라이언트의 정보를 나타내며, 기능 필드는 발생한 이벤트에 대한 종류를 나타내고 데이터 필드는 각 기능에 대한 값을 나타낸 것이다.

참여자가 아바타를 통하여 이벤트를 발생시키면, 가상세계 클라이언트는 이 이벤트를 아바타

처리기에서 클라이언트의 정보 및 이벤트의 기능과 데이터를 통신 패킷으로 묶어 중계 서버로 전송한다. 이렇게 전송된 이벤트 정보는 이벤트 라우터를 통해 가상세계 필터로 보내지게 되며, 이 정보는 다시 가상세계 필터를 통하여 동일한 가상 공간을 공유하는 타 클라이언트에게 전송된다.

중계 서버로부터 전송된 타 참여자의 이벤트 정보는 아바타 처리기에서 이벤트 정보에 해당하는 이벤트 노드를 얻어와 EAI를 통해 Scene Graph 관리기에서 scene graph를 갱신한다. 이렇게 갱신된 정보는 렌더링 모듈을 통해 가시적으로 표현된다.

또한 각 클라이언트에서 추가되거나 갱신된 아바타의 이벤트 정보는 중계 서버에서 이벤트 라우터를 거쳐 클라이언트 정보 목록에 저장된다. 클라이언트 정보 목록은 이벤트를 발생시킨 클라이언트의 정보와 현재의 아바타 상태 및 위치를 포함하고 있다.

4. 결론

웹 상에 3차원 그래픽을 표현하기 위한 표준안으로 VRML이 등장한 이후, 이는 가상현실을 위한 개발 수단으로 광범위하게 사용되고 있으며, 오늘날에는 보다 동적인 표현과 참여자간의 상호작용을 지원하기 위한 새로운 VRML 스펙이 등장하기에 이르렀다. 그러나 현재까지의 VRML은 3차원 가상환경을 사용자들이 네트워크를 통해 서로 공유하며 사용자들 간의 상호작용을 지원하는 수준까지는 이르지 못하고 있는 실정이다.

이에 따라 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 등장하였

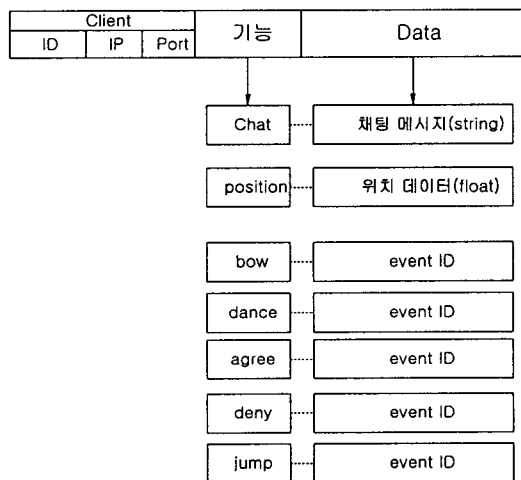


그림 18. 이벤트 전송 패킷의 구조

으며, 현재 활발한 연구가 이루어지고 있는 실정이다.

이에 본 논문에서는 3차원 가상공간을 관리하기 위한 가상세계 서버와 이벤트를 중계와 공간 데이터를 전송하기 위한 중계 서버, 그리고 이들 서버와 통신하는 가상세계 클라이언트의 구조 및 이들 간의 상호작용을 제안하였다.

본 시스템은 가상 쇼핑물, 가상 은행, 원격 교육, 사이버 오피스, 3차원 온라인 게임 등 다양한 분야에 적용될 수 있을 것으로 전망된다.

참 고 문 헌

- [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] R. Lea, Y. Honda, and K. Matsuda, "Community Place : Architecture and Performance", ACM VRML '97, pp. 41-49, 1997.
- [4] 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
- [5] Rory O'Neill, Eden Muir, *Web Developer. Com Guide to Creating 3d Worlds*, Wiley & Sons, 1996.
- [6] Wolfgang Broll "Populating the Internet : Supporting Multiple Users and Shared Applications with VRML", Jurgen Fechter, 1997.
- [7] Olof Hagsand, "Interactive Multiuser VEs in the DIVE System", IEEE MultiMedia, Spring, pp. 30-39, 1996.
- [8] Mike Wray, Rycharde Hawkes, "Distributed virtual environments and VRML : an event-based architecture", Computer Networks and ISDN Systems, pp. 43-51, 1998
- [9] D. R. Pratt, S. M. Pratt, P. T. Barham, R. E. Barker, M. S. Waldrop, J. F. Ehler, and C. A. Chrislip, *Humans in Large-scale, Networked Virtual Environments*, PRESENCE, MIT Press, Vol. 6, No. 5, pp. 547-564, 1997
- [10] Cremer, J. Kearney, "Driving Simulation : Challenges for VR Technology", IEEE Computer Graphics and Applications, pp. 16-20, 1996.
- [11] 허도영 외, "3차원 가상공간에서의 Multi-Avatar 중계 시스템의 설계 및 구현," 추계멀티미디어학회 추계 학술논문집 제1권 제2호, 1998.11.
- [12] 이진호 외, "WWW에서 다중사용자를 지원하는 가상환경 공유 기법," 정보과학회논문지(A) 제 25권 제4호, 1998. 4.
- [13] Community Plaform, Multi-User VR Browser, Blaxxun Interactive. <http://www.blaxxun.com/>
- [14] Living Worlds, Making VRML 97 Applications Interpersonal and Interoperable. Draft 2, 1997, <http://www.wrml.org/WorkingGroups/living-worlds/>
- [15] "Distributed Interactive Simulation" DIS-Java-VRML Working Group, <http://www.web3d.org/WorkingGroups/vrtp/dis-java-vrml/>
- [16] Jean Bunn, MBONE(Multicast Backbone), Geneva University, <http://www.unique.ch/seinf/mbone.html>.
- [17] History of the VRML Specification, <http://www.vrml.org/about/historyofvrml.htm>
- [18] Chris Marrin, "External Interface for VRML 2.0", <http://www.marrin.com/vrml/externalAPI.htm>
- [19] "VRML97 Specification, ISO/IEC 14772-1:1997", <http://www.vrml.org/Specifications/VRML97/1>



황 대 훈

- 1977년 동국대학교 수학과 졸업(학사)
 - 1983년 중앙대학교 대학원 전자계산학과(공학석사)
 - 1991년 중앙대학교 대학원 전자계산학과(공학박사)
 - 1983년~1985년 한국산업경제기술연구원(KIET) 연구원
 - 1995년~1999년 경원대학교 전자계산소장
 - 1987년~현재 경원대학교 교수
 - 관심분야: 멀티미디어 시스템, 인터넷 응용기술, 가상현실 등
-
-