3차원 가상공간에서의 Multi-Avatar 중계시스템의 설계 및 구현

○
허도영*, 전영훈*, 여인국**, 황대훈*
*경원대학교 전자계산학과
**산업기술정책연구소

A Design and Implementation of Multi-Avatar Routing System on 3D Virtual Space

○
Do-Yeong Heo*, Young-Hoon Jun*, InGook-Yeo**, Dae-Hoon Hwang*
*Dept. of Computer Science, Kyungwon Univ.
**KAITECH

요 약

오늘날 네트워크 기술의 발전과 인터넷 보급이 대중화됨에 따라, 가상현실 기술로 구축된 인터넷상의 가상공간에 3차원 가상현실 기술을 접목하므로서, 가상현실이 제공하는 몰입감과 임장감을 체험하고 원거리의 네티즌과 동일한 가상환경을 공유하면서 상대방의 움직임과 동적 위치를 확인하고 의사 전달을 수행하고자 하는 요구가 발생하고 있다. 현재의 인터넷상에서의 웹은 HTML문서에 기반을 둔 정적인 2차원의 정보만을 제공하고 있다. 그러나 이러한 2차원의 인터넷에서도 3차원 정보를 표현할 수 있는 수단을 제공하는 VRML이 등장하였다.

이에 본 논문에서는 이 언어로 표현된 3차원 가상공간에서의 사용자간 이벤트를 중계하는 이벤트 중계 서버와 중계 서버들을 관리하는 위치 서버, 그리고 이들 서버와 통신하는 클라이언트와 인터넷상의 다른 세계를 관리할 수 있는 모듈을 설계하고 구현하였다. 또한 이를 통하여 가상공간에 참여하는 사용자를 나타내는 분신인 아바타를 통하여 각자의 가상공간을 체험하고 다른 참여자와 대화할 수 있는 시스템을 구현하였다.

I. 서론

오늘날 인터넷의 응용 분야가 다변화되고 있는 것과 더불어, 다음과 같은 사회적 변화와 기술적 시도들이 인터넷 환경에 큰 변화를 요구하고 있다.

우선, 정보의 흐름이나 상황의 전개를 사용자가 참여하여 제어할 수 있는 메커니즘인 쌍방향성과 네트워크 상의 3차원 가상공간(cyberspace)에서 또 다른 참여자의 움직임을 인지하면서 상호작용을 통하여 의사를 전달하는 원격존재(telepresence)를 요구하고 있다. 아울러 현실세계와는 또 다른 세계인 가상세계를 구축하고자 하는 인공성(artificiality)과 사용자가 실제 세계에 빠져있는 듯한 느낌을 제공하는 몰입감(immersion)에 대한 요구가 발생하고 있다.

이러한 요구를 만족시키기 위해서는 원거리의 네티즌과 3차원의 동일한 가상환경을 공유하면서 상대방의 움직임과 동적 위치를 확인하고 상호 의사 전달을 수행할 수 있는 시스템이 필요하다. 이에 본 논문에서는 인터넷상에서 가상현실을 구현하기 위해 다중 사용자간 장면 공유 및 가상공간에서 사용자의 대리인인 아바타(avatar) 관리를 하는 중계 시스템을 설계 및 구현하였다.

II. 관련 연구

2.1 가상현실과 VRML

가상현실(virtual reality)은 컴퓨터가 만들어낸 가

상환경(virtual environment)에서 사용자가 마치 실제 세계에서 상호작용 하는 것처럼 할 수 있게 해주는 인간과 컴퓨터와의 진보된 인터페이스이다.[1]

VRML(Virtual Reality Modeling Language)은 네트워크에서 3차원 가상 공간을 표현하기 위해 고안된 개방형이며 기계 중립적인 표준언어이다. 특히 인터넷상에서 웹 브라우저를 통해 여타 HTML 문서들처럼 URL(Uniform Resource Locator)을 통해 접근할 수 있으며 MIME타입으로는 model/vrml을 사용한다.[2] 1994년 11월 SGI 오픈인벤터를 토대로 하여 정적인 3차원 장면을 기술할 수 있는 VRML 1.0 이 발표되었으며 그 후 SGI에서 제안한 Moving World가 1996년 8월 VRML 2.0으로 정식 발표되었는데, 이는 다양한 기능의 센서를 통한 상호작용 및 객체 행위를 기술하기 위한 스크립트를 지원하고 있다. 그러나, 여전히 다중 참여자가 공통의 가상환경에 함께 참여하여 서로 통신하고 상호 작용할 수 있도록 언어차원에서 지원해주지는 못하고 있다. 다중 사용자에 대한 지원은 VRML 3.0 명세에 반영될 것이다.[3]

그림 1의 VRML 브라우저 인터페이스는 Chris Marrin이 제안한 개념적인 VRML 2.0 브라우저를 보인 것이다.[4] SAI(Script Authoring Interface) 와 EAI(External Authoring Interface)는 VRML 장면에 대한 장면내의 Script 노드나 HTML 페이지 상의 Java Applet 등과의 인터페이스로 사용된다. 나머지 Browser Programmer's Interface 등은 VRML 브라우저의 기능을 확장하려는 프로그래머를 위한 것이다.

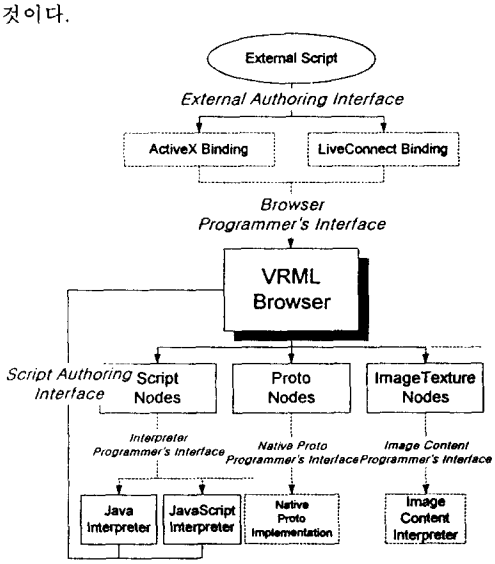


그림 1. VRML 브라우저 인터페이스

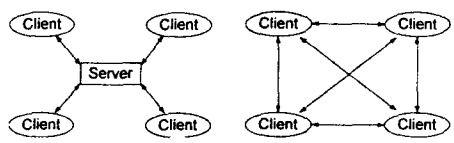
SAI는 Script 노드에 포함된 스크립트와 브라우저 기능과의 인터페이스이다. 그에 반해 EAI는 VRML 브라우저와 Netscape의 LiveConnect 또는 Microsoft의 ActiveX등의 외부 세계와의 인터페이스[5]로 3차원의 동적 컨텐트를 포함하는 응용 프로그램을 생성할 수 있게 한다.

2.2 통신망의 연결 구조

클라이언트와 서버의 연결 구조는 서버의 개수, 클라이언트와 서버의 연결 형태에 따라 중앙 집중형, 완전 분산형, 복합형과 혼합형 등으로 분류된다.[6]

(1) 중앙 집중형과 완전 분산형

중앙 집중형 구조는 중앙의 서버가 모든 통신의 중계 역할을 하는 구조로 서버가 시스템 전체의 정보를 관리하고 모든 통신을 중계하므로 제어하기 쉽고 일관성을 유지하기가 용이하다.



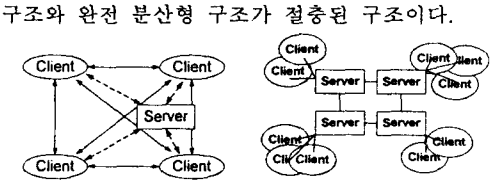
(a) 중앙 집중형 (b) 완전 분산형

그림 2. 중앙 집중형과 완전 분산형

완전 분산형의 구조는 모든 클라이언트가 서로 연결을 해서 데이터를 주고받고 제어를 하는 구조로 서버의 병목점을 제거할 수 있지만 동시성 제어나 클라이언트간에 공유되는 정보의 제어가 어렵고 시스템 일관성을 유지하기 힘들다.

(2) 복합형과 혼합형

그림 3의 (a)와 같은 복합형 구조는 중앙 집중형 구조와 완전 분산형 구조가 절충된 구조이다.



(a) 복합형 (b) 혼합형 구조

그림 3. 복합형과 혼합형

혼합형 구조는 기존의 중앙 집중형 구조와 완전 분산형 구조의 단점을 일부 보완한 것으로 서버로부터 공유된 가상 세계와의 접속을 위해 기존의 클라이언트-서버 방식으로 접속하고, 서버간에는 peer-

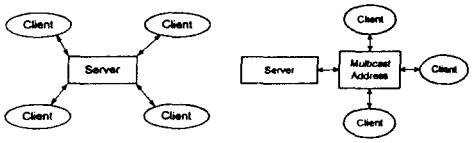
to-peer 형태의 완전 분산 구조를 채택하고 있다.

본 논문에서는 이러한 혼합형 연결 구조를 사용하고 있다.

2.3 데이터 중계 모델

클라이언트의 데이터를 다른 클라이언트들로 중계하기 위한 중계 모델은 그림 4와 같이 유니캐스팅과 멀티캐스팅 방식이 있다.

클라이언트 쪽에서 발생시킨 가상세계의 변화를 서버가 다른 클라이언트들에게 중계할 때 연결된 클라이언트의 수만큼 데이터를 보내는 방식이 유니캐스팅이다. 이는 클라이언트의 수가 많을수록 부하가 커지고 전체 시스템의 속도도 떨어지는 단점이 있다.



(a) 유니캐스팅 (b) 멀티캐스팅

그림 4. 유니캐스팅과 멀티캐스팅

멀티캐스팅을 이용한 경우는 IP(Internet Protocol) 주소에 한 번만 전송하면 각 참여자가 이를 받을 수 있으므로 전송 패킷을 줄일 수 있는 장점이 있으나 인터넷 전체에서 통용되는 기술은 아니며 부가로 멀티캐스트 라우터 같은 장비가 필요하다. 현재 인터넷에서는 멀티캐스트 시험망인 MBONE(Multicast Backbone)[7] 이 존재한다.

본 논문에서는 구현이 용이하고 클라이언트에 특별한 설정이 필요 없는 유니캐스팅 방식을 사용하였다.

III. Multi-Avatar 중계 시스템의 설계

3.1 아바타와 가상객체

가상공간에 존재하는 객체는 그림 5 처럼 가상공간에서의 사용자들의 시각적인 형상[8]인 아바타와 사람이 아닌 가상객체로 나눌 수 있으며, 가상객체는 자율성의 수준에 따라 자율객체와 동적객체, 정적객체로 구분된다.

자율객체(autonomous object)는 자신의 기능이 컴퓨터 프로그램에 의해 기술된 객체로 가상 공간을 스스로 항해하면서 특정한 업무를 수행하거나 참여자들의 편의를 제공한다. 동적객체(dynamic object)는 자율성은 없으나 특정 이벤트가 발생했을 때 이에 반응하여 단순한 동작을 할 수 있는 객체이다. 정적객체(static object)는 행위를 하지는 않고 가상공간을 이루는 구성요소로서 기능하는 객체이다.

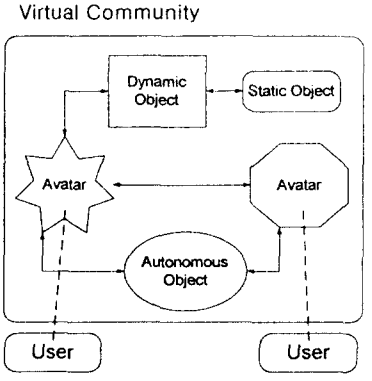


그림 5. 아바타와 가상객체

아바타와 가상객체간의 행위에는 대화, 항해, 공간이동, 객체선택, 객체이동 등의 행위가 있다.

대화는 참여자가 아바타를 통하여 다른 참여자와 대화하는 행위이며 항해는 사용자가 자신의 아바타를 통해 가상세계를 옮겨 다니면서 가상객체들을 보는 것을 의미하며 공간이동은 다른 세계로의 이동을 의미한다. 객체 선택은 3차원 가상객체를 마우스 등의 입력장치로 클릭 하는 등의 동작으로 이를 통해 아바타와 가상객체가 서로 상호작용 할 수 있다. 또한 객체이동은 특정 객체를 선택하여 다른 위치에 위치하는 것이다.

가상세계의 객체들 사이의 통신은 아바타 대 아바타간의 통신과 아바타 대 가상객체간의 통신의 두 가지로 나눌 수 있다. 첫 번째는 참여자간의 의사교환의 수단을 제공하는 것이며 마지막은 참여자가 가상공간내의 가상객체에 서비스를 요청하고 그 결과를 받는 것이다.

가상 세계는 여러 개의 작은 가상세계로 나누어 있는데 이들 작은 가상 세계를 브라우징할 수 있는 디렉토리 브라우징을 구현하기 위하여 다음의 정보를 관리한다.

표 3.1. 디렉토리 브라우징을 위한 정보

디렉토리 브라우징을 위한 정보들
가상공간내의 전체 또는 구역별 참여자의 수
각 참여자의 현재 위치(어느 가상 세계에 존재하는가 ?)
각 참여자의 로그인 이후 경과시간 (time stamp)
참여자의 ID

3.2 사건 정보 전송

(1) VRML 이벤트

VRML에서의 이벤트는 무언가 발생했다는 것을 나타낸다. 예를 들면 사용자가 마우스를 클릭 한다든지 시간이 경과되는 것들이다. 이벤트에는 'eventIn' 이라는 쓰기 전용 이벤트와 'eventOut' 이라는 읽기 전용 이벤트가 있다. VRML 문법에는 다음과 같은 네 가지 종류의 필드 접근 방법이 있다.

field	일반적인 접근으로부터 숨겨진 private 필드
eventIn	한 노드에게 값을 보냄-쓰기 전용 필드
eventOut	한 노드로부터 값을 보냄-읽기 전용 필드
exposedField	읽기와 쓰기에 대해 모두 접근이 가능.

(2) 이벤트 중계

노드는 여러 개의 입력과 출력 필드로 구성될 수 있지만, 모든 것이 연결될 필요는 없다. 대개 단지 몇 개만이 연결된다. VRML은 다음과 같이 ROUTE 키워드를 사용하여 노드의 명백한 연결을 기술한다.

```
ROUTE fromNode.fieldName1 TO toNode.fieldName2
```

이러한 Route 메커니즘은 스크립트와 결합될 때 매우 강력해질 수 있다. 기본적으로 VRML ROUTE나 Java 스크립트는 하나의 eventIn 이나 프로세스에게 eventOut을 보낼 수 있다.

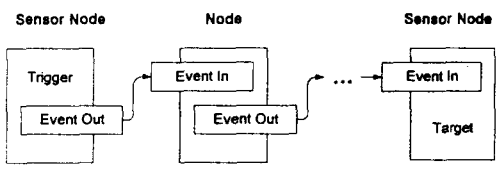


그림 6. VRML의 이벤트 라우팅

(3) Script 노드

Script 노드는 VRML 내에 일반적인 동작을 통합하는 방법을 제공한다. 동작은 브라우저가 지원하는 언어와 API를 사용하는 언어로 프로그램 된다. 때때로 센서는 ROUTE 에 의해 Script 노드와 묶일 수 있다. Script 노드의 정의는 다음과 같다.

```
Script{
  exposedField MFString url []
  field        SFBool directOutput      FALSE
  field        SFBool mustEvaluate      FALSE
  # any number of:
  eventIn      필드형          이벤트명
  field        필드이름        초기값
  eventOut     이벤트형        이벤트명
}
```

Url 필드는 요청된 동작 스크립트로의 URL을 명시하는 임의개수의 스트링을 포함할 수 있다. 컴파일된 Java에 대해, 이것은 .class 파일의 URL이 된다. directOutput 필드는 스크립트가 이벤트를 다른 노드로 보내기 위해 직접 접근할지를 제어한다. Java 메소드는 필드 값을 지정할 때 다른 노드의 노드참조를 요구한다. 예를 들어, 만약 한 스크립트가 이 필드를 TRUE로 지정하고, 트랜스폼 노드의 인스턴스로 전달된다면 이 스크립트는 직접 그 노드에게 이벤트를 보낼 수 있다. 그룹에 디폴트 박스를 생성하기 위한 스크립트 코드는 다음과 같다.

```
SFNode group_node = (SFNode)getField("group_node");
group_node.postEventIn(
        "addChildren",
        (Field)CreateVRMLfromString("Box")
);
```

3.3 사건 정보 중계 프로토콜

기존 클라이언트 서버 시스템은 서버에 연결된 클라이언트의 수가 많을수록 부하가 증가되며 서버의 용량에 따라 클라이언트의 수가 제한된다. 이러한 문제는 서버를 여러 개 두어 분산시켜 해결하는데, 이때 고려해야 할 문제는 처음 클라이언트를 서버에 연결할 때 어떤 서버에 연결하느냐하는 것과 서로 다른 서버에 연결된 클라이언트들이 상호간에 어떤 방식으로 통신하는가 하는 것이다.

(1) 중계 서버로의 연결

첫 번째 문제는 그림 7과 같이 위치 서버(location server)를 두어 해결할 수 있다. 사용자는 원하는 가상세계가 어디에 있는지 모르므로, 위치 서버가 다른 서버들의 데이터를 가지고 이를 대신 찾아주는 것이다. 즉, 단순히 잘 알려진 위치 서버에 연결하여 원하는 중계 서버(routing server)로 연결만 하면 된다.

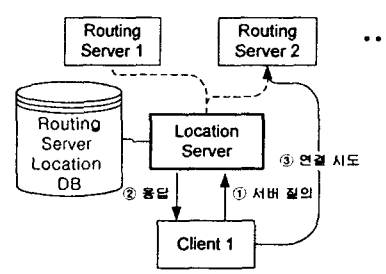


그림 7. 위치 서버의 기능

(2) 초기화면 전송

클라이언트가 중계 서버에 연결된 후 서버가 클라이언트에게 가상객체들의 형상 데이터를 전송하는 방법은 두 가지가 있다.

하나는 그림 8과 같이 웹서버를 통해 wrl 파일 형태를 받는 방법이고, 또 다른 하나는 커스텀 하게 구현된 통신 모듈을 통해 현재 장면을 유지하고 있는 서버로부터 장면 데이터를 전송 받아 클라이언트 측의 VRML 브라우저에 설정하는 방법이다.

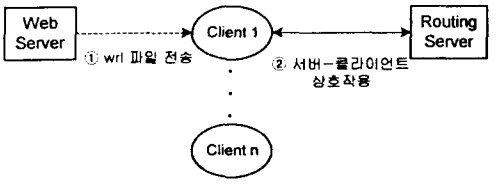


그림 8. 웹서버를 이용한 초기 장면 전송

그림 8과 같은 첫 번째 방법은 단순하고 쉽게 구현이 가능한 방법이나 장면이 사용자의 행위에 의해 바뀌게 되었을 때 이 바뀐 장면을 유지·전송할 방법이 없다.

그림 9와 같은 두 번째 방법은 웹 서버 측이 아닌 중계 서버측에서 데이터를 보관해야 되고 클라이언트에 의해 변경된 가상객체의 속성 정보를 업데이트 해서 보관해야하는 부담을 가지는 단점이 있다.

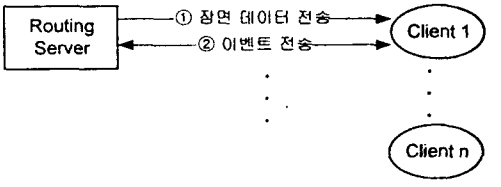


그림 9. 중계 서버를 이용한 초기 장면 전송

(3) 이벤트 중계

VRML 이벤트의 전송을 위해서는 장면에서의 특정 사건을 탐지하는 Sensor Node를 두고 ROUTE를 사용하여 이 이벤트를 서버로 전송하는 Java Applet으로 전달한다. Java Applet에서는 중계 서버로 파라미터와 함께 이벤트 번호를 전송한다. 중계 서버는 이를 다른 모든 클라이언트들에게 전송한다. 이를 전송 받은 다른 클라이언트는 전송 받은 이벤트 값을 어떤 노드의 필드로 설정하는지를 나타내는 이벤트 테이블을 확인하여 해당 필드의 값을 바꿈으로서 이벤트를 처리한다.

3.4 다중 참여자간의 상호작용

(1) 객체의 생성과 삭제

VRML 브라우저가 유지하고 있는 장면에 객체를 동적으로 추가하거나 삭제하기 위해서는 그룹 노드를 사용한다. 노드를 이 그룹 노드에 추가하기 위해서는 그룹 노드에 *eventIn* 이벤트인 addChildren을 보낸다.

```
MFNode box = (MFNode)CreateVRMLfromString("Box");
group_node.postEventIn("addChildren",(Field)box);
```

이 그룹 노드에 있는 서브 노드를 삭제하려면 다음과 같이 eventIn 이벤트인 removeChildren를 보낸다.

```
group_node.postEventIn
(
        "removeChildren",
        (Field)box
);
```

(2) 아바타 형태정보 전송

가상공간의 참여자가 아바타를 만들고 이를 다른 사용자와 공유하기 위해서는 클라이언트 쪽에서 아바타 설계를 위한 인터페이스를 제공해 주어야한다. 또한 동적으로 클라이언트에서 생성된 아바타는 다른 클라이언트에게도 전송되어야 한다. 이때 클라이언트는 자신이 연결된 서버로 아바타 형태 정보인 VRML 문자열 데이터를 전송하고 서버는 이를 다른 클라이언트로 중계한다.

아바타의 행위가 발생하면 다른 참여자가 이를 볼 수 있어야 한다. 이를 위해 아바타의 움직임을 ProxymitySensor 노드 등의 센서 노드로 감지하여 중계 서버로 전송할 수 있어야 하며 또한 참여자가 접속을 끊었을 때는 중계 서버가 연결 해제 통신 이벤트를 감지해 모든 클라이언트로 알려 주어야 한다.

(3) 참여자간 의사교환

인터넷상에서 가능한 대화 방법은 텍스트 기반의 IRC와 같은 채팅과 음성 기반의 인터넷 폰, 화상전송 등 여러 가지 유형이 있지만 적은 트래픽에서 실시간 서비스가 가능한 것은 텍스트를 기반으로 한 대화 방법이다. 따라서 가상 세계에서도 키보드를 통한 텍스트 기반의 채팅이 유효할 것이다. 채팅 클라이언트는 Java Applet으로 구성할 수 있고 채팅 서버는 중계 서버와는 별도로 제작할 수 있다.

IV. 시스템 설계 및 구현

본 장에서는 본 논문의 중계 시스템을 구성하는 구성 모듈들의 종류와 그 기능을 설명한다.

본 논문에서는 그림 10과 같이 웹 브라우저 안에 디렉토리 브라우징 모듈과 채팅 모듈, 그리고 VRML 브라우저로 구성된 사용자 인터페이스를 구성하였다.

디렉토리 브라우징 모듈은 사용자에게 현재 참여 가능한 가상 세계들의 목록을 보여주며 위치 서버와 통신한다. 채팅 모듈은 같은 세계에 있는 다른 참여자의 지원하는 대화창을 제공한다. VRML 브라우저는 3차원 가상세계를 공유할 수 있도록 디스플레이를 제공하는 역할을 한다.

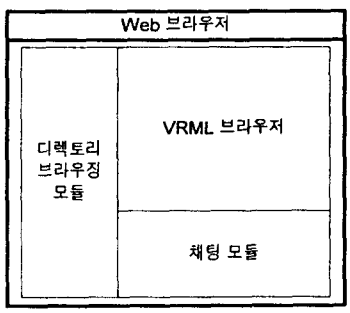


그림 10. 클라이언트의 사용자 인터페이스

다음은 이러한 구성 모듈들을 설명한 표이다.

표 3.1. 구성 모듈

프로그램		설명
서버	위치 서버	중계 서버들의 정보를 수집하여 디렉토리 브라우징을 구현
	중계 서버	참여자들의 사건 정보와 채팅 메시지를 중계
클라이언트	클라이언트 모듈	서버와의 통신을 담당하며 참여자에 의해 발생한 이벤트를 서버에 전달하고 다른 참여자의 이벤트를 전송 받아 화면에 보인다.
	채팅 모듈	중계 서버로 참여자들의 문자열 메시지를 전송하거나 타 참여자의 메시지를 수신
	디렉토리 브라우징 모듈	다른 세계에 대한 정보와 다른 참여자들의 정보를 브라우징

4.1 중계 서버

그림 11은 전체 시스템에서 중계 서버의 처리과정을 나타낸 것이다.

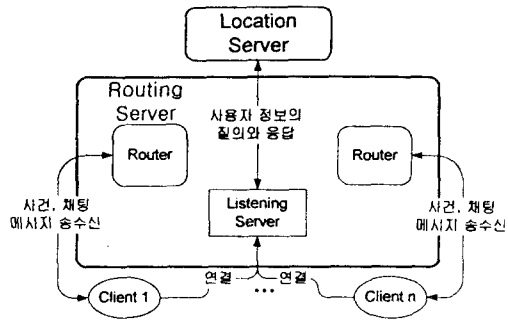


그림 11. 시스템 구성도

중계 서버는 다음과 같은 기능을 수행한다.

- 클라이언트의 행위 이벤트, 아바타 형상 데이터, 텍스트 채팅 문자열등을 다른 클라이언트로 중계.
- 다른 서버의 중계 데이터를 클라이언트로 송신.
- 위치 서버의 질의(현재 사용자수나 최대 사용자 수)에 응답.
- 현재 공유되는 장면을 계속 유지하여 새 클라이언트가 참여할 때 이를 전송.

4.2 위치 서버

위치 서버는 그림 12와 같은 처리과정을 가지며 다음 기능을 수행한다.

- 사용자의 중계 서버 목록 요청 질의를 받으면 중계 서버 목록을 전송
- 중계 서버의 상황 데이터를 수집

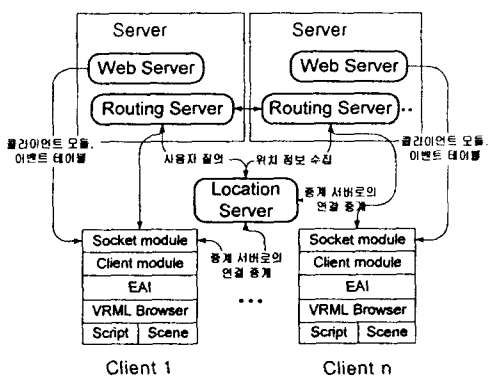


그림 12. 위치 서버의 처리과정

4.3 클라이언트와 VRML 브라우저간의 상호작용

다음은 클라이언트가 수행하는 처리내용이다.

- 클라이언트는 중계 서버로부터 수신되는 행위 이벤트를 해당 VRML 객체에게 이벤트로서 전송
- 클라이언트는 자신에게 일어나는 모든 이벤트 또는 일부 이벤트를 중계 서버에 통보
- 참여자의 채팅 메시지를 다른 참여자에게 보내기 위해 중계 서버로 전송하거나 다른 참여자들의 메시지 수신
- 사용자에게 다른 세계들의 목록을 보이고 선택할 수 있도록 지원

클라이언트는 웹서버와 접속할 수 있는 웹 브라우저와 VRML 브라우저를 필요로 하며, 본 연구에서는 웹브라우저로서 대중적으로 많이 사용되는 Microsoft사의 Internet Explorer 4.0과 Netscape사의 Communicator 4.03 한글판을 사용하였고 EAI를 지원하는 VRML 2.0 브라우저인 Intervista사의 WorldView를 사용하여 실험하였다.

다음 그림 13은 본 연구에서 구현된 사용자 인터페이스를 나타낸 것이다.

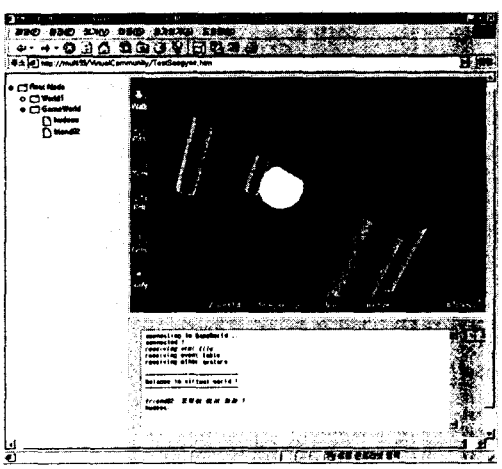


그림 13. 클라이언트의 사용자 인터페이스

V. 결론

본 논문에서는 VRML로 표현된 3차원 가상공간에서의 참여자간 이벤트를 중계하는 중계 서버와 중계 서버와 사용자의 위치를 관리하는 위치 서버, 그리고 이들 서버와 통신하는 클라이언트와 인터넷상의 다른 세계를 관리할 수 있는 모듈을 설계하고 구현하였다. 이를 통하여 가상공간에 참여한 사용자는 자신의 분신인 아바타를 통하여 각자의 가상공간을 체험하고 다른 사용자와 대화할 수 있으며 가상객체들과 상호작용 할 수 있다.

향후 본 연구와 관련하여 지금까지의 결과를 바탕으로 하여 3차원 가상 쇼핑몰, BBS, MUD, 가상 학교, 가상 은행, 모델 하우스 등의 응용 분야에 본 중계 시스템을 활용하기 위한 연구를 수행할 계획이다. 또한, 아울러 3차원 가상세계를 쉽게 구축하기 위한 3차원 객체 모델러, 객체 템플릿 라이브러리, 이벤트 관리기 등과 같은 가상 공간에서의 다중 사용자 지원을 위한 서비스 개발 환경 또는 도구에 관한 연구도 병행할 계획이다.

참고문헌

[1] George Wolberg, Digital Image Warping, IEEE Computer Society Press, 1990.

[2] 전영훈, 여인국, 황대훈, "VRML 파서를 이용한 Scene Graph 생성기의 설계 및 구현", 한국정보처리학회 추계 학술발표논문집 제5권 제2호, 1998.

[3] Jed Hartman, Josie Wernecke, "The VRML 2.0 Handbook", Silicon Graphics, Inc. 1996.

[4] Chris Marrin, "Anatomy of a VRML Browser" (http://www.marrin.com/vrml/Interface.html), August 26, 1996

[5] Chris Marrin, "Proposal for a VRML 2.0 Informative Annex External Authoring Interface Reference", Silicon Graphics, Inc., January 21, 1997 (http://cosmosoftware.com/developer/moving-worlds/spec/ExternalInterface.html)

[6] 김병극, 이 헌주, "네트워크 가상환경에서 확장가능한 멀티미디어 공동작업 시스템", 전자공학회지(제 25권 제 2호), 1998. 2.

[7] Jean Bunn, MBONE(Multicast Backbone), Geneva University (http://www.unique.ch/seinf/mbone.html)

[8] Bernie Roehl, "Late Night VRML 2.0 with Java", ZDPress