

VRML과 JAVA를 이용한 Web 기반 Virtual 방송국 시스템 설계 및 구현

김정현*, 임영태*, 정희경*
 배재대학교 컴퓨터공학과*

Design and implementation of web based virtual broadcasting system Using VRML and JAVA

Jung Hyun Kim, Young Tea Lim, Hoe-kyung Jung
 Dept. of Computer Engineering, Paichai University

최근 WWW(World Wide Web)이 확산되면서, 정보들을 좀더 자유롭게, 시간과 공간의 제약을 받지 않으면서 이용할 수 있게 되었다. 하지만, 초기 Web 기반 기술은 2 차원적이고, 수동적인 정보를 제공하는 정도였다. 이에 비해 사용자들은 좀더 실제의 세계와 유사한 3 차원 환경에서, 능동적으로 참여하기를 원했고 이런 사용자들의 욕구에 발 맞춰 많은 기술들이 쏟아져 나오기 시작했다.

본 논문에서는 최신의 VRML 과 JAVA 기술들을 활용하여 Web 상에서 3 차원 실시간 방송시스템을 구현하였다. 클라이언트가 직접 방송에 참여하거나 관람만 할 수 있는 모드를 제공하고 있으며, 실제 방송 참여자들은 자신의 아바타(avatar)를 선택하여 행동 콘트롤러와 자동 행동 발생시스템을 통해 좀더 동적인 동작과 함께 방송에 참여할 수 있도록 하였으며, 일반 관람인의 경우 웹상에서 실시간으로 방송이 진행되는 모습을 볼 수 있다.

1. 서론

인터넷은 WWW 을 기술하는 언어인 HTML(Hyper Text Markup Language)에 이어 좀더 일관성 있는 포매팅을 제공하고 확장 가능한 마크업 언어인 XML(eXtensible Markup Language)로 발전되어져 왔으며, 3 차원 환경 기술 언어인 VRML 을 통해서 입체적이고, 능동적인 환경을 표현할 수 있었다. 지금까지 다중 참여자들을 위한 가상현실 시스템에 대한 연구들이 이루어져 왔으며, Active world 와 Blacxxun, 그리고 Sony 사의 Community Place 등이 그 대표적인 예라 할 수 있다. 하지만, 이들 역시 기존 브라우저외의 다른 어플리케이션들을 필요로 한다. 그러므로 일반 사용자들이 사용하기에 복잡한 절차를 필요로 한다는 단점이 있다.

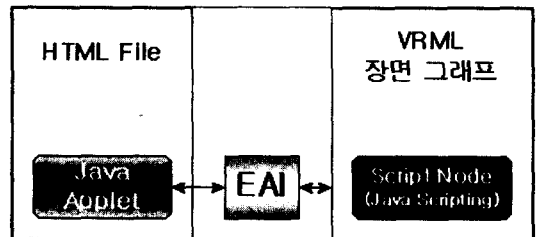
이에 본 논문에서는 가상세계의 표현을 위한 VRML 언어와 아바타의 동작과 제어를 위해 이식성이 뛰어난 JAVA 를 이용한 EAI(External Authoring Interface)기술을 사용하여, 별도의 어플리케이션 없이도 일반 사용자들이 부담감 없이 방송에 참여하거나 관람할 수 있도록 WEB 상에서 별도의 어플리케이션이 없이 참여할 수 있는 가상 방송국 시스템을 구현하였다.

2. 배경

VRML1.0 은 실리콘 그래픽스사의 OpenInventor 를 기반하여 1994 년에 정의되었다. VRML1.0 은 단순하게 물체와 장면들을 3 차원으로 표현해주고, 좀더 다양한 시점들만을 제공하여, 물체를 이동할 수 있는 기능

외에는 별다른 특징이 없었다. 이러한 3 차원 장면과 오브젝트들에 좀 더 다양한 움직임을 부여하고자 실리콘 그래픽스에서 제한한 Moving World 가 1996 년에 VRML2.0 으로 발표되었다.

VRML2.0 에서는 스크립트 노드들과 이벤트를 전달하는 라우터, 새로운 노드를 생성할 수 있는 프로토타입 등과 같은 좀 더 다이나믹한 정의 방법들을 제공하며, EAI 와의 사용으로 멈추어 있던 3 차원 오브젝트들에 움직임을 부여하고 제어할 수가 있게 되었다. EAI 은 [그림 1]에서와 같이 VRML 세계와 HTML 문서에 있는 JAVA 애플릿과의 인터페이스를 제공하는 것이다. 이것은 개발자가 대화적이고 동적인 3D 장면들을 갱신하는 특화된 응용프로그램을 개발하기 위한 방법을 제공한다. 따라서 이러한 외부 응용프로그램들이 VRML 장면과 대화하게 되므로 JAVA 애플릿을 통하여 다양한 사용자 인터페이스 구성이 가능하고, VRML 의 기능을 크게 확장시킬 수 있다.



[그림 1] VRML, EAI, Java Applet 간의 관계

3. 시스템 설계

3.1 클라이언트 서버 (Client - Server)

하나의 컴퓨팅 성능이 뛰어난 서버와 이를 이용하는 성능이 서버보다는 약한 여러 개의 클라이언트 시스템이 연결된 형태이다. 서버는 각 클라이언트에 의하여 변경된 정보를 처리하여 동일한 가상환경을 유지하는 역할을 담당한다. 각각의 클라이언트들은 서버를 통하여 다른 사용자들에게 전달되는 방식이며, 이는 클라이언트들의 정보와 데이터를 주고받음으로써 일어나는 일들에 대해서 서버가 관리한다.

3.2 시스템 설계

3.2.1 서버

- 캐릭터 데이터 (Character DATA)

각각의 캐릭터에 대한 정보를 담고 있다. 캐릭터 정보는 VRML 형태의 각 부분별 데이터로 저장되어 있다. 이 데이터를 VRML Data Agent 에서 조립하여 하나의 완성된 캐릭터로 만든후 예약 캐릭터 콘트롤 시스템, 프로듀스 엔진 및 자동 행동 발생 시스템에 넘겨 주게 된다.

- VRML 2D/3D 차원 채팅 시스템

3 차원 적인 채팅 시스템 및 일반적인 2 차원 적인 채팅 시스템을 모두 갖추고 있다. 3 차원적인 채팅 시스템으로 아바타의 상단에 글자가 위치하며 글자 자체가 3 차원 텍스트이기 때문에 보는 각도에 따라 모두 다르게 보인다. 이로써 직접 다른 아바타와 좀더 친밀히 혹은 그 반대의 경우로도 대화를 나눌 수 있다.

- 프로듀스 엔진

모든 캐릭터 데이터와 클라이언트를 관리해주는 모듈이다. 캐릭터 데이터를 조합하고 행동 양식을 설정한 후 각각의 클라이언트에게 재분배하는 동작을 한다.

- 예약 행동 발생 시스템

일정한 동작 양식을 미리 짜여진 스크립트에 의해 일련의 동작을 취하는 방식이다.

아직까지는 캐릭터 부분별 세분화된 제어가 어렵기 때문에 이런 스크립트된 동작을 이용함으로써 다양한 행동 양식을 표현 할 수 있다.

이러한 행동 양식으로 고객 돌리기, 팔들고 내리기, 다리 들고 내리기가 구현되어 있으며 앉기, 인사하기, 춤추기 등의 다양한 동작도 예약 행동 발생 시스템을 통해 구현 할 수 있다.

- 자동행동 발생 시스템

자신이 취하는 행동이 아닌 상대방의 특정 동작에 대해 자동으로 아바타가 반응 하는 동작이다. 예약 행동 발생 시스템의 응용 시스템으로써 상대방이 링크를 할 경우 눈이 튀어 나오는 등의 자동적인 행동

을 취하게 된다. 지금 현재는 링크, 스마일에 대한 자동 행동 발생 동작만 구현되어 있다. 그러나 이를 응용하여 다양한 자동 행동을 구현 할 수 있다.

3.2.1 클라이언트

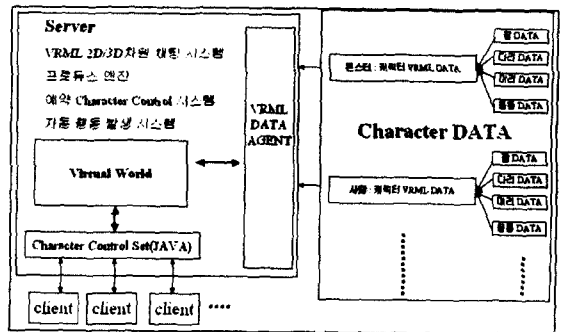
- 방송 참여자로서의 클라이언트

자신이 직접 방송에 참여하여 아바타를 제어하며 자신의 논제를 피력할 수 있다. 즉, 방송에서와 마찬가지로 방송 참여자가 되는 것이다.

- 방송 관람자로서의 클라이언트

다른 참여자들의 행동, 말 하는 것을 관람하며 투표권을 행사할 수 있다. 각각의 방송 참여자들의 행동과 말하는 모든 것을 관람할 수 있으며 방송 관람자끼리의 대화도 가능하다.

또한, 방송 관람자들의 참여를 유도하기 위해 투표 시스템도 있어 자신이 동조하는 아바타에게 투표를 행사할 수 있다. 이 투표 시스템의 경우 한 클라이언트 대해 한번의 기회만 주어진다.



[그림 2] 시스템 구성도

4. 구현 및 고찰

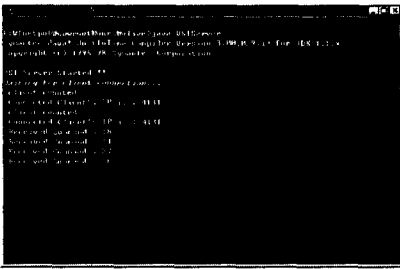
본 시스템은 기존의 VRML 문법을 JAVA 를 이용하여 단순히 제어하고 보여주는 형태가 아닌 JAVA 언어와 EAI 를 통한 연동과 함께 자동행동발생 시스템, 예약 캐릭터 제어를 통해 좀더 발전된 형태의 가상공간을 꾸밀 수 있는 한 형태를 제공하였다.

서버가 동작하면, [그림 4]와 같은 메인 화면이 뜨게 되며, 사용자의 선택에 따라 참여하거나 관람만 할 수 있는 모드를 제공한다. [그림 5]와 [그림 7]에서처럼 참여자는 원하는 아바타를 선택할 수 있으며, 상태창과 아바타를 제어 할 수 있는 콘트롤러가 주어지게 되어, 이 콘트롤러에서 아바타들을 제어할 수가 있다. 또한, [그림 6]과 같이 참여자들은 가상세계에서 자신의 아바타를 통해서 관람자들이 지켜보는 가운데, 토론을 할 수가 있으며, 관람자들은 자신들의 의견을 수렴하여 참여자들에게 피력시킬 수가 있으며, 그들이 참여자 각각에 대해서 투표할 수도 있다.

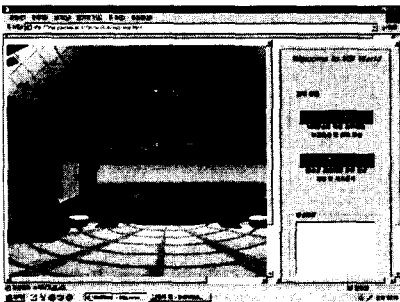
이러한 동적인 동작과 함께 사용자들과의 의사소통을 원활히 할 수 있다.

하지만, JAVA 애플릿은 바이트 코드 형태로 되어 있어 JAVA 애플릿을 실행 시키고자 한다면 바이트 코드를 실행 시킬 수 있는 코드로 바꾸어 주는 인터프리터가 필요하며, 이 때문에 C 와 같은 다른 언어로 작성한 프로그램 보다 상당한 속도 저하를 가져온다.

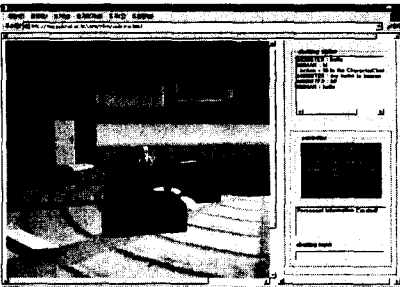
그러므로 속도면에서 상당한 시스템을 요구로 한다. 또한 VRML 역시 모델링툴(Lightwave)을 사용하여 모델링한 데이터를 3DMAX2.0 에 의해 VRML 로 컨버트하는 방식을 사용하였기 때문에, 그 데이터의 양도 방대하다. 앞으로의 시스템은 좀더 데이터의 양이 작고 표현력이 강한 VRML 처리를 위한 연구와 EAI 를 통해 제어하는 VRML 의 좀더 효율적인 연구가 필요로하며, 아바타의 동작을 좀 더 세분화 시켜 행동 제어를 좀 더 자유롭게 할 수 있는 방법에 관한 연구가 필요하다.



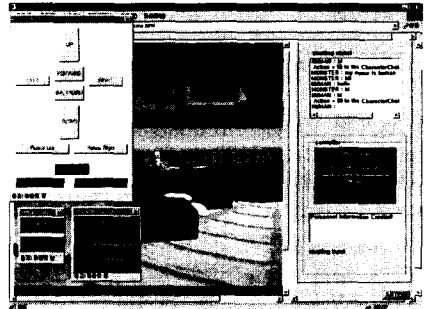
[그림 3] 서버동작 상황 표시



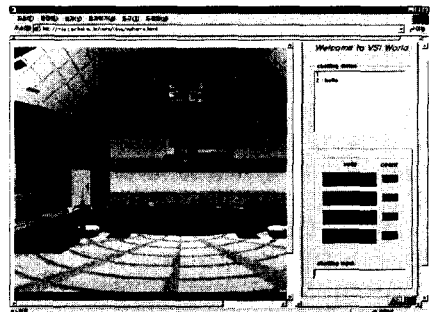
[그림 4] 가상스튜디오의 사용자 인터페이스



[그림 5] 방송참여자 아바타간의 채팅



[그림 6] 액션 및 컨트롤러



[그림 7] 일반 방송관람자 인터페이스

5. 결론

VRML 과 JAVA 의 발전은 사용자들의 요구에 따라 꾸준히 발전하고 있으며, 현재 구현된 3 차원 가상현실 시스템의 경우 많은 제약이 따르며 고급 시스템 사양을 추구하지만, VRML 과 JAVA 의 빠른 발전 속도와 초고속 정보화 사업들을 기반으로 점차 이러한 한계는 극복되어 나가고 있다. TVHTML 로 기술되는 웹 TV 와 같은 좀 더 편리하고 쉽게 사용할 수 있는 매체들도 등장하고 있다. 이 가상스튜디오는 이러한 가상공간을 활용하여 사이버정치를 한다거나 가상투표 시스템등의 사회의 다양한 통합을 목적으로 웹 기반의 가상적인 사회를 건설하려는 연구로써 발전되어질 수 있을 것이다.

참고문헌

- [1] <http://www.activeworlds.com/community/index.html>
- [2] Multi-User VRML Environments by Bruce Damer
<http://www.vrmlsite.com/apr97/a.cgi/spot2.html>
- [3] <http://www.sonypic.com/>
- [4] Stephen, N 외. Special Edition USING VRML
- [5] <http://www.web3d.org/vrml/vrml.htm>
- [6] <http://user.chollian.net/~vrml/java/>