

Webber: 네트워크상에서의 가상 환경 멀티미디어 시스템

김 현 준*, 이 현 주**

LG 종합기술원 정보기술(연) 선임연구원*, LG 종합기술원 정보기술(연) 책임연구원**

요 약

Virtual Reality의 많은 응용 분야 중 중요한 한가지는 collaboration 분야이다. collaboration이란 거리적으로 떨어져 있는 사람들이 네트워크를 통해 상호 작용하고 공동의 작업을 할 수 있게 해 주는 것으로, 시스템은 사용자의 의지를 있는 그대로 표출할 수 있도록 다양한 멀티미디어 서비스를 제공해 주어야 한다. 본 논문에서는 다양한 멀티미디어 서비스를 제공하여 네트워크상의 사용자들이 collaboration work를 수행할 수 있게 해 주는 시스템인 Webber에 대해서 설명한다. Webber는 서버/클라이언트 구조를 기반으로 구현되었고, 네트워크로 연결된 사용자들이 가상의 환경에서 키보드, 마우스, 펜 입력 장치 또는 마이크를 사용하여 의사소통을 할 수 있는 환경을 제공한다. Webber는 시스템에서 제공되는 서비스를 담당하는 Service-PUP(Processing Unit Plug-in)이라는 서비스 블록(Service Block)과 이들 PUP들을 관리하고 이들간의 통신을 담당하는 Framework로 구성되어 있다. 각 Service-PUP들은 하나의 기계에 다중 프로세스로 구축되거나 네트워크로 연결된 서로 다른 기계에 분산될 수 있고, 이들 PUP의 독립성으로 인해 서비스의 유지 보수, 새로운 서비스의 추가 삭제가 간편하다.

1. 서 론

Virtual Reality(VR)의 정의를 내리자면 실제와 같은 환경을 시뮬레이션(Simulation)하여 사용자가 그것과 상호작용할 수 있게 하는 인간과 컴퓨터간의 진보된 인터페이스(Interface)라 할 수 있다(1). 정의에서 보듯이 VR에서 가장 중요한 기술 두 가지는 실제와 같은 환경을 시뮬레이션하는 기술과 시뮬레이션된 환경을 인간

의 오감(시각, 청각, 촉각, 후각, 미각)으로 연결해 주는 인터페이스 기술이다. 즉 VR은 실세계와 같이 만들어진 Virtual Environment(VE)와 몰입감(immersion)을 제공하는 입출력 장치들로 구성된다. VR과 관련된 입출력 장치는 인간의 오감을 측정하고 제어하는 연구라 할 수 있는데 현재 많은 연구가 활발히 진행되고 있다. 그러나 우리가 흔히 볼 수 있는 HMD(Head-Mounted Display)에서도 아직 어지러움이나 불편한 착용감 등의 문제를 가지고 있는 것에서도 알 수 있듯이, 아직도 연구의 초기 단계라고 볼 수 있다. 따라서 본 논문에서는 몰입감을 주는 입출력 장치를 제외하고, 현재 사용하고 있는 컴퓨팅 환경하에서 어떻게 가상 환경을 구축하고 멀티미디어 환경을 제공하여, 사용자들이 그 안에서 상호작용할 수 있도록 할 것인지에 초점을 맞추어 기술할 것이다.

VE란 실제나 가상의 공간을 실시간으로 시뮬레이션하여 사용자가 그 공간에서 마치 실제 세계를 경험하는 것처럼 만들어 주는 환경을 말한다(2). 그 가장 간단한 형태가 전화나 BBS라 할 수 있는데, 이것들은 음성이나 문자를 이용하여 공간상으로 떨어져 있는 사람들이 마치 한 공간에 있는 것과 같이 만들어 주므로 전화를 통한 VE의 구축이라고 할 수 있다. 전화나 BBS를 VE라고 볼 때 또 다른 VE로 생각할 수 있는 것이 인터넷(Internet)이다. 초기에 인터넷은 원하는 정보를 시간과 공간의 제한없이 얻고자 하는 목적으로 이용되었으나 근래에 들어서는 사용자들간의 collaboration computing을 지원하는 방향으로 발전해 가고 있다. collaboration computing이란 공간적으로 떨어져 있는 일단의 사람들이 컴퓨터 네트워크를 통하여 오디오나 비디오와 같은 멀티미디어 데이터를 이용하여 의사소통을 하며 공동된 Application에 참가하여 공동의 목적을 이루는 것을 말한다(3). 이때 collaboration computing에서 상호작용(interaction)은 매우 중요한 요소인데 가상 공간과 사

용자간의 상호작용, 사용자와 사용자간의 상호작용이 모두 중요하다.

지금까지 많은 인터넷 사이트(site)들이 제공하는 가상의 공간들은 텍스트와 그림이 추가 되는 이차원의 VE이었으나, 이는 우리가 살고 있는 실제 공간을 묘사하는데에 한계를 가지고 있다. 이를 해결하기 위해 제안된 것이 VRML(Virtual Reality Modeling Language) (4)인데, VRML version 1.0은 단순히 3차원 공간을 기술할 수 있는 언어인 반면 version 2.0은 사용자와 가상 공간간에 상호작용을 할 수 있는 여지를 주고 있다. 그러나 이에는 한계가 있을 뿐만 아니라 사용자와 사용자간의 상호작용을 지원하지 않기 때문에 VRML은 collaboration computing을 위한 VE구현에는 적합치 않다. 그러므로 collaboration computing을 위한 VE구현에는 3차원 장면을 기술하는 VRML 기능 이외에 사용자들간에 멀티미디어 매체를 이용하여 상호작용할 수 있도록 하는 기능이 필요하게 된다. 이에 본 논문에서는 다양한 멀티미디어 서비스를 제공하여 네트워크상의 사용자들이 collaboration work을 수행할 수 있게 해 주는 Webber 시스템의 구조와 구현 방법에 관해 설명하려 한다.

2장에서는 가상 환경 시스템이 충족해야 할 요구 사항과 이에 대한 기존 연구의 해결책을 알아보고, 3장에서는 Webber의 구조, 특성과 구현 방법을 기술한다. 마지막으로, 4장에서 결론과 추후 과제들을 언급한다.

II. 관련 연구 및 시스템 요구 사항

지금까지 연구되어 온 대표적인 Virtual Environment (VE) 시스템 중에서 collaboration computing에 이용할 수 있는 시스템들을 간단히 살펴보고, 이를 구현하는데 결정해야 할 중요한 사항들과 기존의 시스템들이 이런 사항들을 어떻게 해결하였는지에 대해 살펴보자.

주로 군사 훈련 목적으로 만들어진 NPSNET(5)는 광역망(WAN) 상에서 다중 사용자를 지원하는 응용 시스템을 위한 구조이며, AVIARY(6)는 계층적인 시스템 구조와 동적인 부하 균형(load balancing) 기법을 사용하였다. 이밖에 서버/클라이언트 모델을 사용한 BrickNet(7), peer-to-peer 모델을 사용한 DIVE(2) 등이 있으며, 전용 하드웨어와 통신망을 바탕으로 높은 질의 가상 공간과 서비스를 제공하는 '다이아몬드 공원'

(8)과 같은 시스템도 있다. 또한 'PointWorld' (9), 'OnLive' (10)과 같이 실제로 인터넷을 통해 서비스를 제공하는 사이트들도 있다.

1. 가상 공간의 데이터 관리

VE 시스템이 제공하는 가상 공간에는 다양한 종류의 기하 객체(geometric entity)들이 존재하며, 사용자 객체를 비롯한 각종 활동 가능한 객체들의 운동은 예측 불가능하다. 따라서 VE 시스템은 이와 같은 객체들의 돌발적인 변화 및 생성과 소멸을 효율적으로 관리해야 한다. 또한, 가상 공간에 참여하는 불특정 다수의 사용자들에 관한 관리와 시스템을 구성하는 각 요소들 간의 상호 작용을 처리해야 한다.

VE를 이루는 정보들의 효율적인 관리를 위해 대개의 시스템들은 가상 공간의 기하 객체들을 주로 공간 상의 위치를 기준으로 분류·보관하며, 시스템의 각 프로세스들은 그들이 처리해야 할 데이터와 기능에 따라 구분하여 정의하였다(5, 11).

2. 시스템 연결 구조

VE 시스템은 일반적으로 서버/클라이언트 구조나 일대일 연결 구조, 혹은 이들의 변형 중 하나를 따르게 된다.

서버/클라이언트 구조의 서버는 응용 시스템의 가상 공간 혹은 시스템을 운영하는데 관련된 정보를 관리하여 응용 시스템 내의 정보가 올바른 상태로 유지되도록 하며 여러 클라이언트로부터 전달되는 정보를 중계하고 관리한다. 각 클라이언트들은 서버를 통해 원하는 정보를 받는다. 앞에서 언급한 시스템 중 BrickNet은 서버/클라이언트 구조로 정의되었는데 이러한 구조의 시스템은 서버가 모든 정보를 가공 및 분배하기 때문에, 서버의 처리 부분이 병목 지점(bottleneck)으로 작용할 가능성이 있다. 이를 극복하기 위해, BrickNet은 서버 시스템을 분산하여, 각 서버 시스템들이 가상 공간에 접속하는 사용자들을 나누어 관리하게 하였다.

NPSNET, DIVE, AVIARY 등 대부분의 기존 시스템은 일대일 연결 구조로 구축되었다. 일대일 연결 구조는 응용 시스템의 모든 구성 요소들끼리 물리적인 연결 상태를 유지하는 형태이다. 따라서 시스템의 모든 구성 요소들은 원하는 상대와 직접 정보를 주고 받을 수 있는데 이를 위해서는 시스템의 각 구성 요소들이 각각

전체 시스템 정보를 지니고 있어야 한다. 이 때 각 구성 요소에 중복되어 있는 정보에 대해 시스템 전체적으로 일관성(consistency)을 유지하기가 어렵다.

3. 분산 네트워크

현실감과 일의 효율을 증가시키기 위해서는 VE 시스템이 다양한 기능들을 제공해야 하며 이는 많은 시스템 자원을 요구하게 된다. 따라서 시스템 구성 요소들을 적절한 하드웨어에 분산, 구축한다면 시스템을 좀 더 효율적으로 구현할 수 있을 것이다. 이 때 분산된 각 시스템 요소들이 네트워크를 통해 주고 받는 정보량은 VE 시스템의 규모와 시스템이 제공하는 기능들이 어떤 것인가에 따라 결정되는데 만약 네트워크를 통해 전달되는 정보양이 많아져 정보 전달이 어려워진다면 현실감이 손상되므로 바람직하지 않다. 예를 들어, 음성 데이터를 지원하는 시스템의 경우, 네트워크 통신에 의한 지연과 데이터 처리에 의한 지연을 모두 합하여 100ms 이상이 걸리면 현실감이 떨어진다(2).

이와 같은 정보 전달의 지연 현상을 막기 위해, 네트워크 통신량을 적절한 수준으로 유지하고 이에 적절한 네트워크 프로토콜을 고안하는 것이 바람직하다.

NPSNET은 분산 시뮬레이션 시스템을 위해 미 국방성에서 개발한 DIS 프로토콜을 사용하였다. DIS는 분산 시스템의 구성 요소들이 통신을 위해 지원해야 할 데이터 포맷, 상호 작용적인 시뮬레이션 관리, 라디오 통신, 보안, 성능 측정 기준 등 광범위한 문제를 다루고 있다. 이 프로토콜은 안정성을 보장하나 상황이 늘 변하는 VE 시스템을 지원하기에는 제한적인 것으로 알려져 있다(5).

적절한 네트워크 사용량의 유지는 네트워크 프로토콜에 의한 제어 방법뿐만 아니라, 응용 시스템의 정보 처리기법에 의해 달성할 수도 있다. 이 중에서 기존의 거의 모든 시스템이 사용하는 대표적인 방법으로 dead-reckoning(5)방법이 있다. Dead-reckoning은 클라이언트 시스템의 자체 시뮬레이션 기능을 확보하여 서버 시스템의 부하를 줄이고 서버와 클라이언트 간에 변화 내용만을 메시지 형태로 전달하는 방법으로 서버와 클라이언트간에 교환해야 할 정보량을 줄이는 방법이다.

4. 서비스의 추가, 삭제, 유지, 보수

공간적으로 떨어져 있는 사람들이 공동의 디자인 작

업을 하는 경우를 생각해 보자. 이 작업을 위해서는 단순한 텍스트 채팅 보다는 오디오나 비디오를 이용한 의사 소통이 효율적이며, 여기에 화이트 보드나 3차원 모델을 보면서 토론을 하고 디자인을 변형하면 좋을 것이다. 이와 같이 collaboration computing을 수행하기 위해서 시스템이 제공하는 기능, 즉 시스템이 제공하는 서비스들이 다양 할수록 일은 효율적으로 진행될 것이다. 그런데 기술이 진보해 감에 따라 추가할 수 있는 서비스들은 더욱 증가해 갈 것이며, 기존의 서비스들도 더 효율적인 방법들이 개발 될 것이다. 따라서 collaboration system들은 새로운 서비스를 추가하거나 삭제, 유지, 보수가 쉬워야 하는데 기존의 시스템들은 이런 고려가 불충분하다.

III. WEBBER

1. 개요

Webber는 공간적으로 떨어져 있는 다중의 사람들이 네트워크를 통해 구성된 삼차원 가상 공간 내에서 collaboration할 수 있도록 해 주는 응용 시스템이다.

Webber는 객체 지향 개념을 도입하여 전체 구조를 계층적이고 모듈화 하여 구성하였다. 따라서 개발자는 좀 더 효율적인 모듈 단위를 구현하는데 노력을 집중할 수 있고, 시스템의 구성 모듈을 서로 상이한 하드웨어 시스템에 분산하기에 용이하다. 이와 같은 개념을 바탕으로 Webber의 개념도를 그려보면 그림 1과 같다. 그림 1에서 보듯이 Webber는 서비스 모듈들을 관리하고 이를 필요에 따라 발생시키거나 제거하는 Framework과 각 서비스를 관리하는 Service-PUP(Service Plug-in Unit)으로 구성되어 있다. Framework은

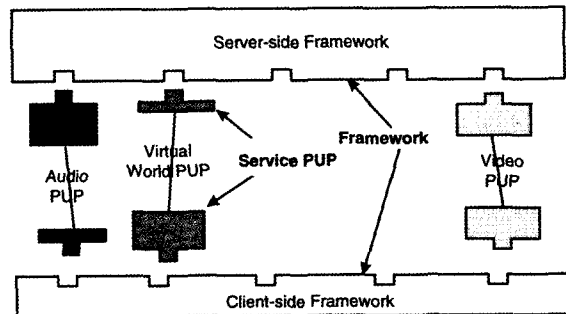


그림 1. Webber의 개념도

표 1. 현재 Webber에서 제공하는 PUP들

이름	설명	비고
VRML-PUP	VRML 1.0 Browsing 기능	VRML 1.0 spec 이외에 내부적으로 정한 avatar에 관한 spec 포함
Navigation-PUP	가상공간 Navigation 기능	fast rendering을 위한 preprocessing/culling 기능을 포함
Chat-PUP	Text Chatting	
ACS-PUP	Audio Conference System	Audio mail 기능 포함
WhiteBoard-PUP	White Board	Text 파일을 비롯한 몇 가지 종류의 파일 포맷을 read/save 기능 포함
PenR-PUP	Hand-Script Recognition	문자 인식기
VCS-PUP	Video Conference System	개발중

Service-PUP와의 인터페이스를 위한 내부적인 표준과 각 서비스들간의 통신이나 필요한 정보를 얻기 위한 다양한 라이브러리를 제공한다. 그러므로 각 Service-PUP을 구현하기 위해서 PUP개발자가 전체 시스템에 대해서 알아야 할 것은 Framework과의 인터페이스와 라이브러리뿐이다.

이와 같은 구조로 현재 Webber에서 제공하고 있는 PUP들은 표 1과 같으며, 이를 통해 사용자들은 3차원의 가상 공간에서 다른 사용자와 상호작용할 수 있다. 즉 가상 공간내에서 다른 사용자들을 대표하는 가상의 인물(avatar)들을 보면서 텍스트 채팅을 하거나, 스피커와 마이크를 사용하여 상대방의 음성을 들으면서 이야기를 나눌 수 있다. 또한 가상의 이차원 공간인 화이트 보드를 통하여 관심있는 파일을 보거나 그림을 그리며

정보를 주고 받을 수 있다.

2 기본 구조

그림 2에서 살펴 보았듯이 서버/클라이언트 구조는 서버가 시스템 전체의 광역 정보를 관리하므로 시스템 전체적으로 일관성을 유지하기는 용이하나, 서버의 처리 부분이 병목지점으로 작용할 가능성이 있다. 한편 일대일 연결 구조에서는 각 구성 요소 간의 데이터 전달이 비교적 단순한 반면, 다수의 클라이언트 사이에 발생할 수 있는 동시성 제어(concurrency control)가 어려워, 시스템 내의 중복된 정보들의 일관성을 보장하는데 많은 노력을 들여야 한다. 일대일 연결 구조의 이와 같은 단점으로 해서 Webber는 시스템 구성 요소 간의 연결에 있어 서버/클라이언트 구조를 채택하였다.

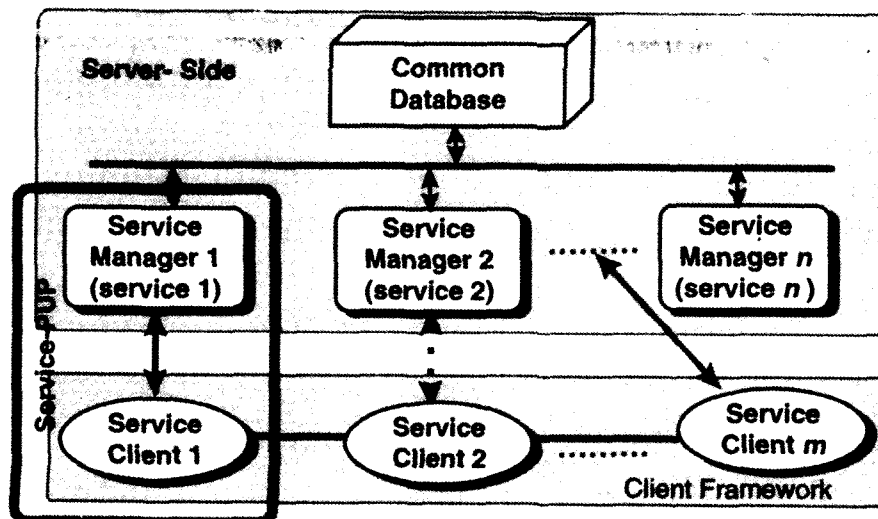


그림 2. Webber의 기본 구조

그림 2에 Webber의 기본 구조가 나타나 있다. 그림에서 보듯이 서비스 PUP들은 서버와 클라이언트 쪽의 서비스 매니저(Service Manager:SVM)와 서비스 클라이언트(Service Client:SVC)로 구성된다. 한 PUP에 있는 SVM와 SVC는 해당 서비스 범위에 관련되어 있는 데이터 베이스를 중복하여 보유할 수 있다. 따라서 SVC에서 이 데이터 베이스를 참조하여 로컬 시뮬레이션을 할 경우 해당 SVM과 SVC가 데이터 베이스와 관련된 수정 메시지만을 주고 받으므로 네트워크 사용량을 줄일 수 있다.

서버 시스템은 SVM들에 해당하는 다중 프로세스로 구성되며, 여러 시스템에서 분산 배치될 수 있다. 시스템에 연결될 클라이언트 시스템은 SVC들의 집합으로 하나의 기계에 설치되며, 각 SVC들이 해당 SVM과 개별적으로 연결한다. 이와 같은 구조는 응용시스템에서 서비스별로 특정 네트워크 프로토콜을 요구하는 경우에 적합하고, 사용자로 하여금 선택적으로 서비스를 제공할 수 있도록 한다. 또한 각 SVM간의 독립성을 보장하여 각 SVM들을 서로 상이한 시스템(Heterogeneous System)에 분산 구축하거나 하나의 SVM이 포화 상태에 이르렀을 경우 해당 SVM을 중복 생성할 수도 있다. 이를 통해 시스템은 서버의 병목 현상을 해결하고 각 서버간의 일의 균형을 맞출 수 있다.

3. 시스템의 분산 구축

앞에서도 언급했듯이 Webber 서버 시스템의 프로세스들은 한 기계 내에 존재할 수도 있으나, 각 SVM은 그 특징에 따라 필요로 하는 자원과 성능이 서로 다르므로 각각을 가장 적절한 하드웨어에 분산 구축하는 것이

더 효율적이다.

Webber의 Framework는 서버 시스템을 분산 구축하기에 용이한 구조와 구성 요소를 제공한다. 우선 서비스 매니저들의 정보가 담길 SVM 테이블을 정의하였다. 이 테이블은 전체 시스템이 구축된 뒤에, 각 SVM들이 각각 어느 기계에 어떤 포트를 이용하여 통신하게 될 것인가에 관해 서술하며, 서버 시스템과 클라이언트 시스템들이 참조하게 된다. 또한 Webber의 서버쪽 Framework에서는 서버 시스템의 분산 구축을 위해 그림 3과 같은 구조를 택하였다.

서버 시스템을 구성하는 각 기계에는 서비스 조정자(Service Coordinator : SC)라는 데몬프로세스가 있어, 이미 정의된 서비스 매니저 테이블을 참조하여 해당 프로세스를 생성하여 가동한다. SVM들이 여러 기계에 분산되어 있을 수 있으므로, 서비스 조정자는 다른 기계의 서비스 조정자를 통해 원격리에 있는 SVM간의 통신을 중재하게 된다. 이 방법은 SVM에게 그 위치에 상관없이 통일된 인터페이스를 제공하고 구체적인 통신 방법과 인터페이스를 분리하여 시스템을 유지, 보수하기 쉽게 한다.

4. 상세 구성도 및 구현

Webber에서는 내부에서 자동으로 만들어지는 LogIn-PUP이라는 서비스 PUP이 있다. 이는 주 서버 쪽에서 만들어지는 것으로 현재 Webber를 사용하고 있는 사용자에게 대한 정보와 여러 SVM이 함께 참조해야 할 글로벌 데이터 베이스를 관리하고 타 SVM의 요구에 따라 이에 대한 정보를 제공한다. 따라서 Webber의 서버 시스템은 글로벌 데이터베이스와 LogIn-PUP의

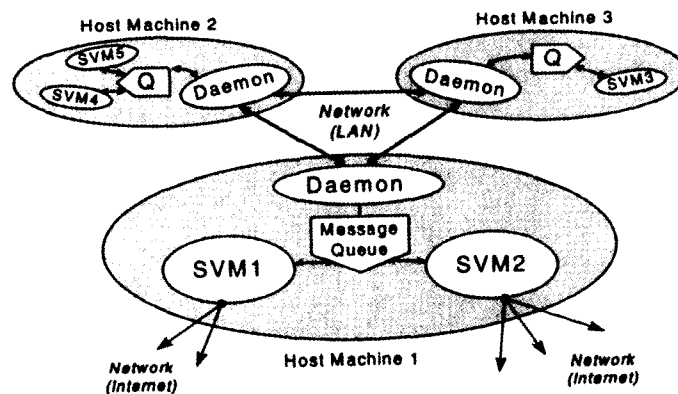


그림 3. 서버 시스템의 분산 구축

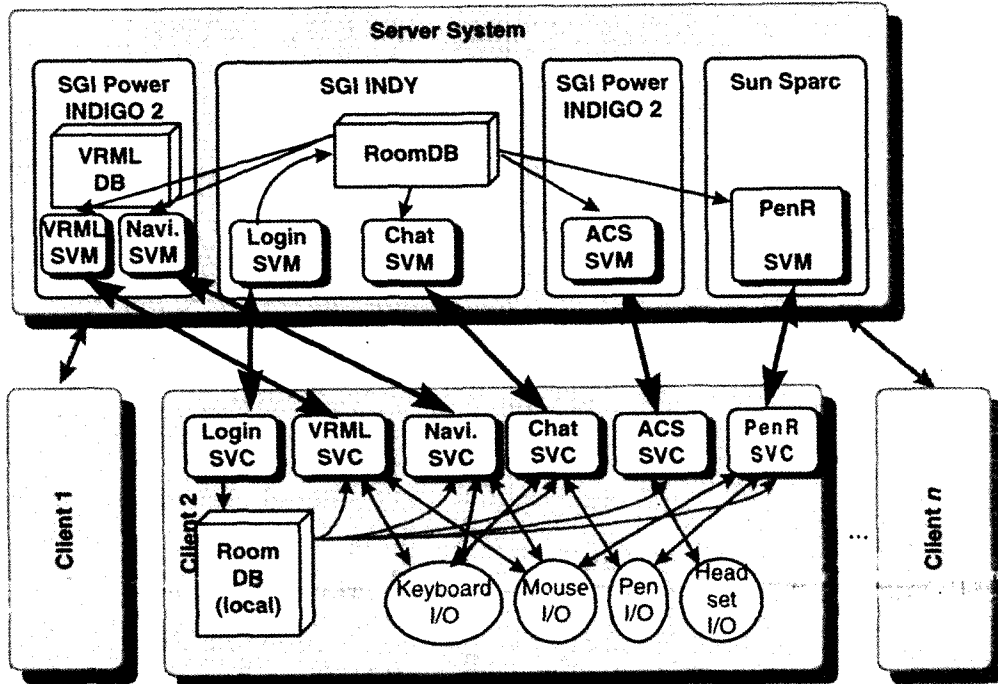


그림 4. Webber 시스템 구현

SVM을 포함한 각 서비스 PUP들의 SVM으로 구성되며, 클라이언트 시스템은 각 서비스 PUP들의 SVC와 이들을 하나의 응용 시스템인 것처럼 만들어 주는 Framework로 구성된다. 표 1에서 언급한 서비스 PUP들을 이용하여 구현된 Webber의 상세 구조가 그림 4에 나타나 있다.

LogIn-SVM는 가상 공간에 참여한 사용자들을 관리하는 것으로, 사용자의 출입을 제어하며 그들의 신상 정보를 담아두는 데이터 베이스를 관리한다. VRML-PUP과 Navigation-PUP은 VRML파일을 읽어 가상 공간을 구축하고 가상 공간 내의 기하 객체를 관리하며 그들의 움직임을 제어한다. VRML-PUP에서는 VRML 1.0파일을 읽어 가상의 세계를 구축하는 기능 이외에, 객체들에게 운동을 부여할 수 있는데 SVM과 SVC 사이의 정보 전달량을 줄이기 위해 객체들의 운동은 SVC에서 시뮬레이션 한다. 또한 이 PUP에서는 가상 공간에서 사용자를 대표하기 위한 아바타(avatar)라는 기하 객체를 제공한다. 아바타는 가상 공간 내에서 사용자의 명령에 따라 몇 가지 표정이나 동작을 표현하고 사용자가 음성 채팅을 할 경우 입을 움직이게 된다. 아바타의 표정을 만들기 위해서 파라미터를 이용한 표정 애니메이션 기법(12)을 도입하였으며 좀 더 현실감 있는

표정을 연출하기 위해 FACS(13)의 연구 결과를 이용하였다. Navigation-PUP에서는 가상 공간에 대한 전처리(preprocessing)를 수행하여 매 프레임마다 필요 없는 객체들을 제거(culling)해 내서 빠른 렌더링(rendering) 속도를 유지 한다. Chat-PUP과 WhiteBoard-PUP는 텍스트 채팅과 화이트 보드 기능을 서비스한다. ACS-PUP은 다자간의 대화 기능을 갖고며 가상 공간 내의 사용자들의 위치를 파악하여 음향의 전체 level 및 좌우 level을 조정한다. PenR-PUP는 온라인 문자 인식 서비스를 제공한다. 이 PUP은 사용자가 마우스 혹은 펜으로 쓴 문자 정보를 인식하고 그 결과를 알려 주는데, 이 결과는 Chat-PUP으로 전달되어 텍스트 채팅에도 이용될 수 있다.

Webber의 서버 시스템은 UNIX 계열의 운영체제 위에 구축되었으며, 그림 4에 보이는 바와 같이, 서버 시스템의 SVM들은 Sun's Sparc 및 SGI 계열의 기계 위에 분산되어 있다. Webber의 클라이언트 시스템은 IBM 호환 기종의 펜티엄 급 PC에서 Windows95 운영 체제 위에 구축되었으며, 음성의 녹음 및 재생은 사운드 블러스터 호환 카드를 이용하였다. 삼차원 그래픽스 렌더링을 위해서는 사용자의 선호도에 따라 Direct3D나 OpenGL 라이브러리를 사용한 PUP을 선택할 수 있다.

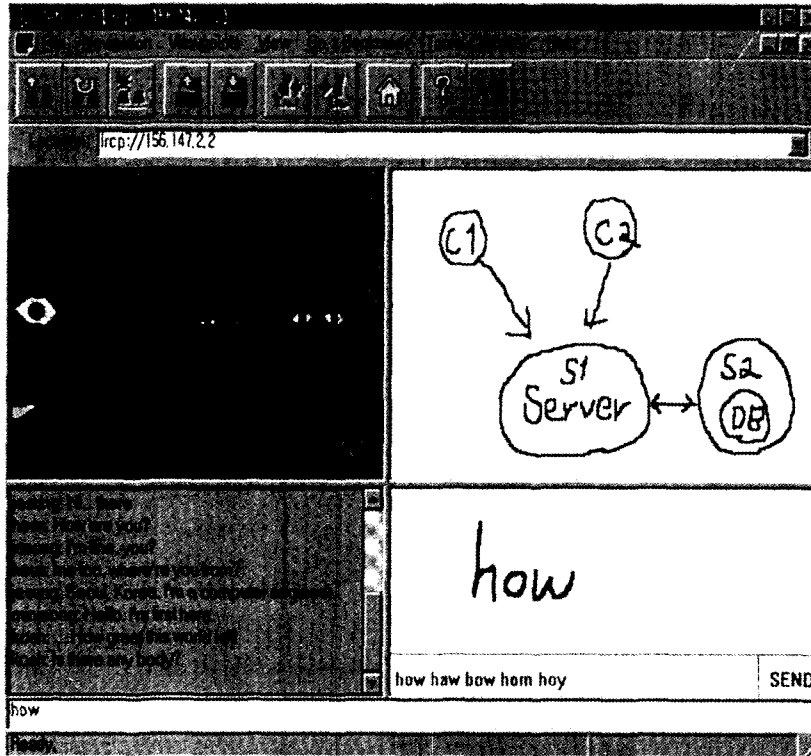


그림 5. Webber 클라이언트 시스템

그림 5는 사용자가 Webber의 서버 시스템에 성공적으로 로그인 하였을 때의 모습을 보여주고 있다.

IV. 결론 및 추후 문제

본 논문에서는 네트워크 환경에서 구현된 가상 환경 시스템인 Webber의 구조와 특징들을 설명하였다. Webber는 Framework과 PUP의 집합으로 구성되며, 각 PUP은 서버/클라이언트 구조를 가진다. 각 PUP은 서로 독립적이기 때문에 네트워크로 연결된 분산 시스템에 쉽게 구축될 수 있고, 서비스 별로 독립적인 개발이 용이하며 새로운 PUP의 추가, 삭제, 유지, 보수가 용이하다. 그러므로 클라이언트 시스템을 구축할 때 보다 융통성이 있는 선택 사양을 제공할 수 있다.

현재 Webber는 몇 가지 방향으로 계속 연구되어 나갈 예정인데 그 중 하나가 JAVA환경으로의 전환이다. 현재의 클라이언트 시스템으로써 일반 PC만을 고려하고

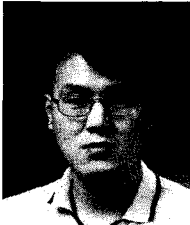
있지만 앞으로 네트워크 환경이 대중화 될 경우 TV, 전화기와 같은 수많은 하드웨어들이 네트워크에 접속하게 된다. 따라서 단말기의 종류에 상관없이 클라이언트 시스템을 설치할 수 있도록 하기 위해 JAVA로의 전환을 연구중에 있다.

또 다른 한가지 방향은 새로운 네트워크 환경이나 프로토콜의 도입이다. 가상 공간의 현실감을 높이기 위해서는 많은 멀티미디어 데이터를 처리할 필요가 있는데, 멀티미디어 데이터의 특성이나 크기를 고려할 때 현재의 TCP/IP기반의 네트워크 환경은 적합하지 않다. 따라서 멀티미디어 데이터 통신에 적합한 프로토콜이나 새로운 네트워크 환경이 필요하게 된다. 현재 이와 관련하여 학계에서는 Mbone(Multicast backbone) 등 연결 망에 관련된 연구, 실시간 데이터 전송을 위한 RTP(Realtime Transport Protocol), RMP(Reliable Multicast Protocol) 등의 프로토콜에 관한 연구와 ATM 망과 같은 차세대 네트워크의 연구가 진행되고 있다. 따라서 Webber에 새로운 네트워크 환경이나 이들 프로토콜을 적용하는 것을 연구중이다.

참고 문헌

- [1] J. N. Latta and D. J. Oberg, "A Conceptual Virtual Reality Model," *IEEE Computer Graphics & Application*, vol. 14, no. 1, pp. 23-29, Jan. 1994.
- [2] Olof Hagsand, "Interactive Multiuser VEs in the DIVE System," *IEEE Multimedia*, vol. 3, no. 1, pp. 30-39, Spr. 1996.
- [3] C. Egelhaaf, E. Moeller, and P. Schoo, "Developing Distributed Multimedia Telecommunication Application," *IEEE Multimedia*, vol. 4, no. 3, pp. 76-81, Win. 1996.
- [4] VRML Architecture Group, The Virtual Reality Modeling Language Specification Version 2.0, <http://webspacesgi.com/moving-worlds/>, 1996.
- [5] M. R. Macedonia, M. J. Zyda, D. R. Pratt, P. T. Barham, and S. Zeswitz, "NPSNET: A Network Software Architecture for Large Scale Virtual Environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 3, no. 4, pp. 288-308, 1994.
- [6] D. N. Snowdon and A. J. West, "AVIARY: Design Issues for Future Large-Scale Virtual Environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 3, no. 4, pp. 288~308, 1994.
- [7] G. Singh, L. Serra, W. Ping, and H. Ng, "BrickNet: A Software Toolkit for Network-Based Virtual World," *PRESENCE: Teleoperators and Virtual Environments*, vol. 3, no. 1, pp. 19~34, 1994.
- [8] D. B. Anderson, J. W. Barrus, J. H. Howard, C. Rich, C. Shen, and R. C. Waters "Building Multiuser Interactive Multimedia Environments at MERL," *IEE Multimedia*, vol. 2, no. 4, pp. 77-82, 1995.
- [9] Black Sun Inc. PointWorld, <http://www.blacksun.com>, 1996.
- [10] OnLive! Technologies, OnLive, <http://www.onlive.com>, 1996.
- [11] R. Kazman, "Load Balancing and Latency Management in a Distributed Virtual World," *3rd International Conference on Cyberspace*, 1993.
- [12] F. Parke, "Parametrized Models for Facial Animation," *SIGGRAPH*, pp. 61~68, 1982.
- [13] P. Ekman and W. Friesen, *Investigators Guide for the Facial Action Coding System*, Consulting Psychologist Press, Palo Alto, Calif., 1978.

필자 소개



김 현 준

- 1989년 2월 아주대학교 전자공학과 학사
- 1991년 2월 한국과학기술원 전기 및 전자공학과 석사
- 1996년 2월 한국 과학기술원 전기 및 전자공학과 박사
- 1996년 3월 ~ 현재 LG 종합기술원 정보기술(연) 선임연구원
- 주관심분야 : Computer Animation, Rendering



이 현 주

- 1983년 2월 경북대학교 전자공학과 학사
- 1985년 2월 연세대학교 전자공학과 석사
- 1990년 2월 연세대학교 전자공학과 박사
- 1995년 6월 ~ 1996년 7월 : UC, Berkeley, Industrial Fellow
- 1990년 1월 ~ 현재 LG 종합기술원 정보기술(연) 책임연구원
- 주관심분야 : 멀티미디어통신, Distributed Object Computing