

## 이종 플랫폼 간 협업을 위한 분산 가상 환경<sup>1)</sup>

\*이석희, 이영호, 우운택

광주과학기술원 정보통신공학과

e-mail : sheelee@kjist.ac.kr, ylee@kjist.ac.kr, wwoo@kjist.ac.kr

### Distributed Virtual Environment for Collaboration between Heterogeneous Platform

\*Seokhee Lee, Youngho Lee, Woontack Woo

KJIST U-VR Lab

#### Abstract

This paper proposes the distributed virtual environment framework for collaboration between heterogeneous platforms. The proposed system consists of Client, Content Server(CS), and Reflector Server(RS). The Client is composed of VR library(VRL), Interaction Manager(IM), Distributed Interaction Manager(DIM), and Network Interface(NI). The proposed Client structure can support various display environments. The XML-based IM and DIM lets the users define and add the interactions. The proposed Server structure synchronizes the distributed interaction between heterogeneous platforms and provides transparency to the users. Therefore, the proposed framework enables the users to implement virtual collaboration environment for their tasks. Moreover, it can also be used for research and development activities between a large number of the remote users. The usefulness of the framework was verified by applying it to the virtual heritage tour system.

#### I. 서론

가상현실 기술은 사람들이 일상적으로 경험하기 어려운 환경을 직접 체험하지 않고서도 그 환경에 들어와 있는 것처럼 보여주고 조작하게 할 수 있는 기술이다. 따라서 이런 가상현실 기술을 기반으로 서로 떨어

져 있는 장소간의 가상의 협업 시스템을 구축한다면 다양한 분야의 공동연구나 공동개발을 위해서 유용하게 쓰일 수 있다.

이런 이유로 다양한 분야에서 가상 협업 시스템 구축에 관한 연구가 진행되고 있다. ImmersaView는 AGAVE나 GeoWall 환경에서 마우스나 키보드 디바이스를 통해서 Open Inventor나 VRML 모델을 공유하면서 협업하는 시스템이다. [1]. COANIM은 CAVE-like 디스플레이 환경과 AGAVE 환경간의 3D 콘텐츠 기반 협업 시스템이다 [3]. 일반적인 협업 시스템이 아닌 사용자의 목적에 특화된 협업 시스템에 관한 연구도 진행되고 있다. 예로 Walkabout은 지형 관측을 위해 WiggleView는 지진 관측을 위해 개발된 시스템들이다 [2][7]. 하지만 ImmersaView는 모델 사용의 제약과 CAVE-like 시스템에서는 사용할 수 없다는 단점을 가지고 있고 COANIM은 리눅스 환경에서만 동작하는 단점이 있다. 그리고 Walkabout과 WiggleView는 다른 목적의 협업에 사용할 수 없다는 단점을 가지고 있다. 하지만 다양한 디스플레이 환경과 가상 환경 라이브러리를 사용하여 가상 환경 시스템을 구축하고 있는 현시점에서 이종 플랫폼간의 가상 협업을 위한 연구가 필요하다. 또한 사용자에 따라 원하는 인터랙션의 방법이 다양하기 때문에 다양한 사용자 정의 인터랙션을 지원하는 기능 또한 필요하다.

본 논문에서는 이종 플랫폼을 지원 가능한 가상 협업 환경을 제안한다. 제안한 가상 협업 환경은 클라이언트, 콘텐츠 서버, 그리고 리플렉터 서버로 이루어진다. VR 라이브러리, 인터랙션 매니저와 분산 인터랙션 매니저로 이루어진 클라이언트 시스템은 다양한 디스플레이 환경을 지원한다. 그리고 XML 기반의 인터랙션 매니저와 분산 인터랙션 매니저는 사용자들이 협업 목적에 맞게 인터랙션을 정의하고 가상환경에 추가할

1) 이 연구는 KIST와 KJIST 지원에 의해 수행 되었음

수 있도록 만든다. 콘텐츠 서버와 리플렉터 서버는 이종 플랫폼 사용자들 간의 분산 인터랙션을 동기화하고 다수의 클라이언트들 간의 투명성이 보장된 협업을 지원한다. 따라서 제안된 프레임워크는 사용자가 쉽게 사용자 목적에 맞는 가상 협업 시스템을 구현 할 수 있게 하고 이종 플랫폼 환경에 있는 다수의 원격지 사용자와 효율적인 공동연구나 공동개발을 가능하게 한다.

본 논문의 순서는 다음과 같다. 2장에서는 우리가 제안한 가상 협업 시스템에 관해 설명한다. 3장에서 제안한 시스템의 구현 사항에 대해 설명하고 4장에서 결론 및 추후 과제에 대해서 언급한다.

## II. 이종 플랫폼간 협업을 위한 분산 가상 환경

### 2.1 시스템 프레임워크

본 논문에서는 이종 플랫폼과 다양한 인터랙션에 기반한 네트워크 가상 환경을 위한 시스템을 제안한다. 가상 협업 시스템이 가져야 할 요구 사항은 크게 (a) 다양한 디스플레이 환경을 지원 (b) 협업 중에 동적으로 3D 데이터를 공유, 그리고 (c) 다양한 방식의 협업을 지원 등의 세 가지로 구분할 수 있다. 이러한 요구 사항들을 충족시키기 위해 그림 1과 같은 시스템 구조를 제안한다. 시스템 컴포넌트들로는 클라이언트, 콘텐츠 서버 그리고 리플렉터 서버가 있다. 클라이언트는 인터랙션이 가능한 3D 뷰어를 가지고 있고 각각 독립적인 컴퓨팅 또는 디스플레이 플랫폼을 가지고 구현된 VR 노드들이다. 콘텐츠 서버는 클라이언트들의 정보와 가상 환경 모델 파일을 가지고 있어서 클라이언트들의 가상환경 참여를 관리한다. 또한 리플렉터 서버는 한 클라이언트나 콘텐츠 서버로부터 받은 데이터를 모든 클라이언트와 콘텐츠 서버로 전송하는 역할을 한다.

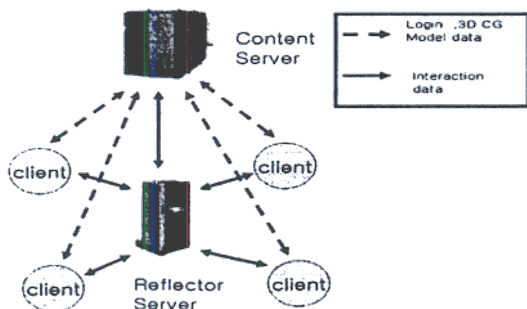


그림 1 시스템 프레임워크

### 2.2 클라이언트 시스템

클라이언트 시스템은 각각의 가상환경 사용자가 실행시키는 프로그램으로써 다음과 같은 기능을 갖는다.

- (a) 콘텐츠 서버에 로그인
- (b) 콘텐츠 서버로부터 데이터 파일과 리플렉터 서버의 주소 다운로드
- (c) 가상환경의 가시화
- (d) 사용자와 입력 디바이스를 통한 다양한 인터랙션
- (e) 콘텐츠 및 리플렉터 서버와의 통신

위와 같은 기능들을 구현하기 위해서 클라이언트 그림 2와 같은 구조의 클라이언트 시스템을 제안한다. VR 라이브러리는 로컬의 VR 노드를 쉽게 구현할 수 있도록 만드는 라이브러리로써 CAVELib, Vrjuggler, NAVERLib등이 있다. VR 라이브러리에는 이미 다양한 디바이스로부터 사용자의 입력을 받아드리는 기능이 있고 데이터를 렌더링하여 디스플레이 장치에 가시화하는 기능이 있다. 첫 번째 요구사항을 충족시키기 위해서는 안정적으로 다양한 디스플레이 장치를 지원하는 VR 라이브러리의 선택이 중요하다. 이렇게 구현이 된 로컬의 VR 노드에 원하는 인터랙션을 정의하고 원격지에 있는 다른 사용자와 상호 작용할 수 있게 하기 위하여 인터랙션 매니저와 분산 인터랙션 매니저, 그리고 네트워크 인터페이스가 필요하다. 인터랙션 매니저는 사용자나 혹은 네트워크로부터 받아들인 정보가 가상환경에 어떻게 반영이 되는지를 정의한다. 위에 언급한 세 번째 요구사항을 충족시키기 위해서는 사용자가 쉽게 인터랙션을 정의할 수 있도록 하는 구조화된 인터랙션 매니저가 필요하다. 분산 인터랙션 매니저는 로컬에서 일어나는 인터랙션을 원격지의 사용자와 공유하기 위해서 원하는 데이터 형식으로 변형시켜주는 역할을 한다. 또한 네트워크로부터 오는 정보를 인터랙션 매니저가 알 수 있는 형태로 변화 시키는 역할도 수행한다. 네트워크 인터페이스는 VR 노드가 리플렉터 서버나 콘텐츠 서버와 통신 할 수 있게 해준다.

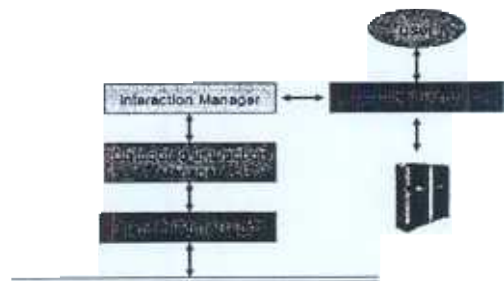


그림 2 클라이언트 시스템의 구조

### 2.3 콘텐츠 서버 시스템

두 번째 요구 조건을 충족시키기 위해서 그림 3과 같은 콘텐츠 서버 시스템 구조를 제안한다. 네트워크 인터페이스는 클라이언트 혹은 리플렉터 서버와 통신을 할 수 있게 만들며 로그인 매니저는 가상 환경에 참여한 클라이언트들을 관리한다. 처음 클라이언트가 콘텐츠 서버에 접속하면 로그인 테이블에 클라이언트를 등록하고 협업을 위한 CG 콘텐츠를 전송한다. 전송을 마치면 리플렉터 서버 주소를 클라이언트에 보내고 리플렉터 서버에 참가한 클라이언트의 주소를 보낸 후 클라이언트들과 접속을 끊는다. 인터랙션에 의해서 변화된 가상환경의 정보는 리플렉터 서버에 의해서 보내어지며 실시간으로 변화를 반영하여 이후에 참여하는 클라이언트에게 최신의 가상환경 정보를 제공한다.

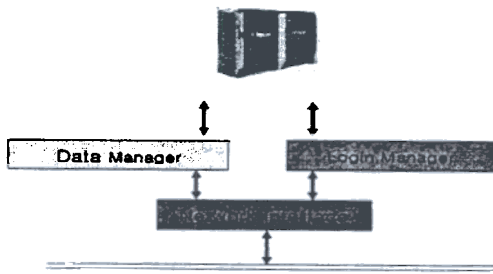


그림 3 콘텐츠 서버 시스템의 구조

### 2.4 리플렉터 서버 시스템

리플렉터 서버는 클라이언트들 간의 투명성을 보장한다. 제안된 리플렉터 서버의 구조는 그림 4와 같다. 네트워크 인터페이스는 두 개의 컴포넌트를 갖는다. 클라이언트와 콘텐츠 서버로부터 데이터를 받는 리시버와 현재 접속되어있는 모든 클라이언트와 콘텐츠 서버에 데이터를 주는 센터가 있다. 리플렉터 서버는 가상 환경에 참여한 노드들에게 데이터를 멀티캐스팅하는 역할 외에 가상환경의 변화를 동기화 시켜주는 작업을 해야 한다. 동기화 매니저는 리시버로부터 받은 데이터의 유용성을 검사하고 적절하다고 판단되면 센터에게 넘겨주는 역할을 한다.

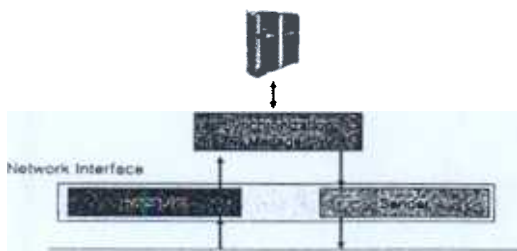


그림 4 리플렉터 서버 시스템의 구조

### 2.5 데이터 타입의 정의

본 시스템은 일반적인 가상 협업을 전제로 하기 때문에 보편적으로 사용될 수 있는 데이터 타입의 정의가 필요하다. 따라서 그림 5와 같은 데이터 타입을 정의하였다. 각각 '/'를 분리자로 하여 5개의 데이터가 있다. WHO는 가상환경을 사용하는 사용자나 사용자 그룹의 ID이며 HOW는 취한 동작을 의미한다. WHAT은 행위의 목적이 대는 가상물체를 지칭하고 WHEN 정보는 데이터의 우선순위를 정할 때 필요한 프레임 넘버이다. WHERE는 가상에서의 위치 정보를 포함하고 있다.



그림 5 데이터 타입

## III. 구현

제안된 시스템의 유용성을 검증하기 위해 그림 6과 같이 분산 환경에서의 가상 문화 유적 답사시스템에 적용하였다. 원격지에 떨어져있는 다수의 사용자들이 한 지역에 있는 듯한 느낌을 가지고 함께 가상 문화 유적을 답사할 수 있었다. 그림 6과 같이 Access Grid시스템과 3D 디스플레이가 연동된 e-AG 시스템에서는 서로 얼굴을 보면서 이야기하며 문화 유적에 대해서 토론할 수 있었다.

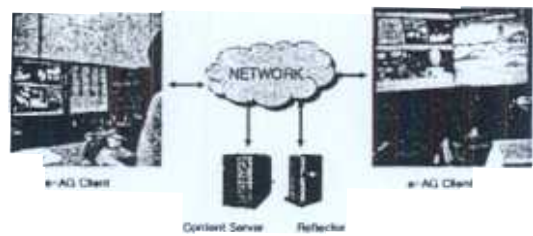


그림 6 e-AG 시스템간의 가상 문화 유적 답사

분산 협업을 위한 네트워크 인터페이스의 구현을 위해서는 네트워크 라이브러리인 Quanta를 사용하였다 [4]. Quanta는 이미 Reflector 클래스를 제공한다. 이를 기반으로 동기화 매니저를 추가하여 리플렉터 서버를 구현하였다. 또한 파일 업로드와 다운로드를 지원하는 remoteFileIO, ParallelFileIO 클래스를 사용하여 콘텐츠 서버와 클라이언트 측 네트워크 인터페이스를 구현하였다.

다양한 사용자 정의 인터랙션을 지원하기 위해서 OOP 컨셉과 XML을 사용하였다. 인터랙션들 마다 공통적으로 가져야할 변수와 메소드를 정의해서 Interaction 베이스 클래스를 만들었고 사용자들은 이를 상속 받아 자신이

원하는 인터랙션을 정의 할 수 있도록 하였다. XML기반 인터랙션 매니저는 인터랙션 테이블을 가지고 있어서 XML 설정 파일에 정의된 인터랙션들을 등록, 관리하며 주어진 시간에 인터랙션을 처리한다. 또한 XML기반의 분산 인터랙션 매니저를 두어서 네트워크를 통해 들어오는 원격지의 데이터를 원하는 로컬의 인터랙션과 파라미터로 바꾸어 줄 수 있게 한다. 그림 7과, 그림 8 은 마우스를 사용하여 앞으로 전진하는 분산 인터랙션의 예이다. 분산 인터랙션의 결과 사용자1과 사용자2는 동적으로 같은 뷰 포지션을 가지면서 네비게이션 할 수 있다.

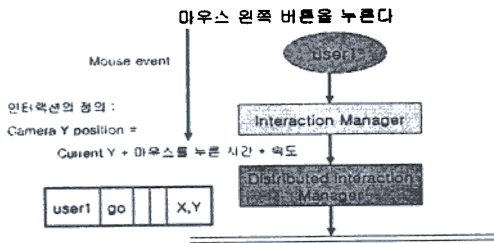


그림 7 사용 정의 인터랙션과 데이터 일반화 예

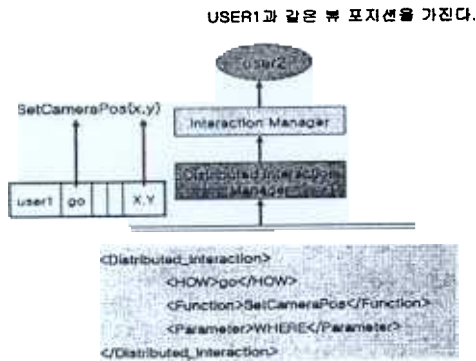


그림 8 분산 인터랙션의 XML 스크립팅 예

다양한 디스플레이 환경을 지원하기 위해서 클라이언트 시스템의 VR 라이브러리로서 NAVERLib과 CAVELib을 사용하였다 [5][6]. NAVERLib은 현재 KIST에서 개발중이 가상 환경 라이브러리로서 XML기반 시나리오 스크립팅 기능을 비롯한 다양한 기능을 지원하고 있다. 하지만 많은 VR 노드들이 CAVELib을 이용하여 구현되어지고 있는 실정이다. 따라서 AGAVE나 Geowall환경에서는 NAVERLib을 CAVE-like환경에서는 CAVELib을 이용하는 클라이언트 시스템을 구현하였다. 그림 9는 구현에 사용된 라이브러리와 시스템 구조를 보여준다.

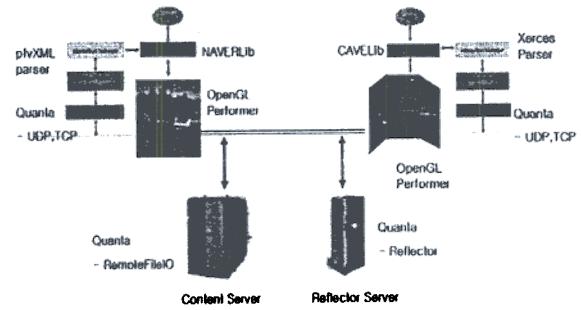


그림 9 가상문화유적답사 시스템 구현을 위한 라이브러리

#### IV. 결론 및 향후 연구 방향

본 논문에서는 이종 플랫폼간의 협업 가상 환경을 구현하기 위한 프레임워크와 각각의 컴포넌트들을 정의하였다. 그리고 그 응용으로 NAVERLib과 CAVELib기반 클라이언트 시스템과 XML기반의 인터랙션 및 분산 인터랙션 매니저 그리고 Quanta기반 네트워크 인터페이스를 가지는 가상 문화 유적 답사 시스템을 구현함으로써 그 유용성을 입증하였다.

본 시스템은 일반적인 협업 환경을 제안하였기에 추후에 더 많은 응용을 통하여 인터랙션의 일반화가 이루어져야 한다. 또한 많은 실험을 통해 안정성 및 확장성을 확고히 하고 클라이언트들 간의 완전한 동기화 문제도 해결해야 할 과제이다.

#### 참고문헌

- [1] [www.evl.uic.edu/cavern/agave/immersaview](http://www.evl.uic.edu/cavern/agave/immersaview)
- [2] [www.evl.uic.edu/aej/macagave/walkabout.html](http://www.evl.uic.edu/aej/macagave/walkabout.html)
- [3] [www.evl.uic.edu/cavern/agave/coanim.html](http://www.evl.uic.edu/cavern/agave/coanim.html)
- [4] He, E., Alimohideen, J., Eliason, J., Krishnaprasad, N. K., Leigh, J., Yu, O., DeFanti, T. A., "Quanta: a toolkit for high performance data delivery over photonic networks"
- [5] C.Park, H.Ko, C.Cho, H.Ahn, Y.Han, T.Kim "NAVER: Design and Implementation of Networked Virtual Environments Based on PC Cluster" Korean Society for Emotion and Sensibility (KOSES), 2002
- [6] [www.vrco.com/products/cavelib/cavelib.html](http://www.vrco.com/products/cavelib/cavelib.html)
- [7] [www.evl.uic.edu/cavern/agave/wiggleview](http://www.evl.uic.edu/cavern/agave/wiggleview)