

가상 공간에서의 협업을 위한 이종 플랫폼 간 인터페이스의 설계 및 제작*

이석희, 이영호, 우운택
광주과학기술원 U-VR 연구실
{sheelee, ylee, wwoo}@kjist.ac.kr

Interface Design of Heterogeneous Platforms for Networked collaboration

Seokhee Lee, Youngho Lee, Woontack Woo
KJIST U-VR Lab

요약

가상현실 기술의 발전과 함께 몰입형 콘텐츠에 대한 관심이 증가하고 있다. 또한, 네트워크의 급속한 보급과 함께 실시간으로 원격지에서 상호 작용할 수 있는 환경이 구축되고 있다. 하지만 다양한 인터랙션을 지원하는 이종 플랫폼 간 협업 환경을 구축하는 연구는 아직 미흡하다. 따라서 본 논문에서는 이러한 문제점들을 해결하기 위해서 이종 플랫폼간에 다양한 상호 작용을 정의할 수 있는 가상 현실 프레임워크 및 네트워크 인터페이스를 제안한다. 먼저 5W1H 형태의 상호작용 전달 프로토콜을 정의하고 가상 현실 프레임워크의 컴포넌트로서 디바이스 매니저, 인터랙션 매니저, 그리고 렌더링 매니저를 제안한다. 또한 XML 기반의 네트워크 인터페이스로서 가상센서, 컨텍스트 생성자, 그리고 컨텍스트 서비스 매니저를 제안한다. 제안된 가상 환경은 VR 라이브러리를 이용하여 구현 되므로 다양한 디바이스와 디스플레이 환경을 지원한다. 독립적인 인터랙션 매니저는 응용 프로그램 마다 다르게 정의되는 상호작용을 효율적으로 관리할 수 있다. 제안된 네트워크 인터페이스는 이렇게 구현된 가상 환경들을 XML 스크립팅을 사용하여 쉽게 연결할 수 있고 분산 상호작용을 정의할 수 있다. 제안된 시스템의 유용성을 확인하기 위하여 e-AG 상의 가상 문화 유적 답사 시스템에 적용하였다.

Keyword : VR, XML, Interaction, Interface

1. 서론

가상현실 기술의 발전과 함께 몰입형 콘텐츠에 대한 관심이 증가하고 있다. 또한, 네트워크의 급속한 보급과 함께 실시간으로 원격지에서 상호 작용할 수 있는 환경이 구축되고 있다. 따라서 사용자들은 경주 가상 극장과 같은 특정한 장소에서 단순히 가상의 객체를 보는 것 이상을 요구하고 있다.

이런 이유로 다양한 분야에서 가상 협업 시스템 구축에 관한 연구가 진행되고 있다. ImmersaView 는 AGAVE 나 GeoWall 환경에서 마우스나 키보드 디바이스를 통해서 Open Inventor 나 VRML 모델을 공유하면서 협업하는 시스템이다 [1]. COANIM 은 CAVE-like 디스플레이 환경과 AGAVE 환경간의 3D 콘텐츠 기반 협업 시스템이다 [3]. 일반적인 협업 시스템이 아닌 사용자의 목적에 특화된 협업 시스템에 관한

* 본 연구는 KISTI와 KJIST의 지원하에 수행 되었음

연구도 진행되고 있다. 예로 Walkabout 은 지형 관측을 위해 WiggleView 는 지진 관측을 위해 개발된 시스템들이다 [2]. 하지만 ImmeraView 는 모델 사용의 제약과 CAVE-like 시스템에서는 사용할 수 없다는 단점을 가지고 있고 COANIM 은 리눅스 환경에서만 동작하는 단점이 있다. 그리고 Walkabout 과 WiggleView 는 다른 목적의 협업에 사용할 수 없다는 단점을 가지고 있다.

본 논문에서는 이러한 문제점들을 해결하기 위해서 이종 플랫폼간에 다양한 상호 작용을 정의할 수 있는 가상 현실 프레임워크 및 네트워크 인터페이스를 제안한다. 먼저 상호작용 전달 프로토콜을 정의하고 다양한 디바이스와 디스플레이 환경을 지원하는 가상 현실 프레임워크 구현한다. 또한 이렇게 생성된 이종 플랫폼간의 협업을 위한 네트워크 인터페이스를 구현한다.

제안된 5W1H 형식의 상호작용 전달 프로토콜은 가상 세계에서 이루어지는 사용자의 다양한 상호작용을 기본적인 데이터 타입으로서 표현할 수 있는 방법을 제공함으로써 각각의 응용에서 정의하는 상호작용을 융통성 있게 처리해 줄 수 있도록 한다. 제안한 가상 협업 환경은 렌더링 매니저, 디바이스 매니저, 인터랙션 매니저로 구성된다. 각각의 컴포넌트들은 CAVELib, VRJuggler, NAVERLib 와 같은 VR 라이브러리를 기반으로 만들어지는 것을 가정하기 때문에 다양한 디바이스와 디스플레이 환경을 지원한다[5][6]. 또한 인터랙션 매니저의 분리는 가상 환경의 목적에 따라 다양하게 상호작용이 재정의될 수 있도록 한다. 이런 로컬 시스템을 기반으로 컨텍스트 서비스 매니저, 가상 센서 그리고 컨텍스트 생성자를 사용하여 네트워크 환경을 구성한다. XML 기반의 컨텍스트 서비스 매니저는 원격의 가상 환경에서부터 전달받은 상호작용 관련 데이터를 분석하여 그에 상응하는 인터랙션 매니저를 호출한다. 가상 센서와 컨텍스트 생성자는 자신의 가상 환경의 변화를

상호작용 전달 프로토콜에 맞추어 기술하고 원격의 가상 환경에 전송한다.

본 논문의 순서는 다음과 같다. 2 장에서는 제안한 가상 현실 프레임워크 및 네트워크 인터페이스에 대해서 설명한다. 3 장에서 가상 문화 유적 답사 시스템으로의 적용을 기술하고 4 장에서는 결론 및 추후 과제에 대해서 언급한다.

2. 이종간 네트워크 가상현실 시스템

2.1 협업을 위한 상호 작용 전달 프로토콜

제안된 상호작용 전달 프로토콜은 가상 세계간에 이루어지는 사용자의 다양한 상호작용을 기본적인 데이터 타입으로서 표현할 수 있는 방법을 제공함으로써 각각의 응용에서 정의하는 상호작용을 융통성 있게 처리해 줄 수 있도록 한다. 가상 환경의 변화는 그림 1 과 같이 5W1H 형태의 상호 작용 전달 프로토콜에 맞추어 정의되어야 된다. WHO 는 가상환경을 사용하는 사용자나 사용자 그룹의 ID 혹은 가상 객체의 ID 이며 HOW 는 취한 동작을 의미한다. WHAT 은 행위의 목적이 대는 가상물체를 지칭하고 WHEN 정보는 데이터의 우선 순위를 정할 때 필요한 프레임 넘버이다. WHERE 는 가상에서의 위치 정보를 포함하고 있다. 예로 가상 세계를 네비게이션하는 사용자 1 의 상호작용은 “User1/*/*x,y,z/*/*move to/*”와 같이 표현될 수 있다. [4].



그림 1. 상호 작용 전달 프로토콜

2.1 로컬 시스템 요구 사항

현재 대부분의 가상 환경들이 그림 2 과 같은 처리 과정을 기반으로 존재한다. 이러한 처리 과정은 크게 디바이스로 부터 입력을 관리하는

부분과 그런 입력된 데이터를 계산하는 부분, 그리고 그것을 사용자에게 디스플레이하는 부분으로 나누어 설명될 수 있다. 따라서 가상 환경 시스템의 컴포넌트를 그림 3 과 같이 디바이스로부터 신호를 받는 디바이스 매니저, 받은 신호를 이용하여 사용자의 의도에 맞게 가상 환경의 변화를 계산하는 인터랙션 매니저, 그런 일련의 과정을 사용자에게 보여주는 렌더링 매니저로 크게 나눌 수 있다.

렌더링 매니저와 디바이스를 구현할 때는 다양한 디스플레이 환경과 상호작용 디바이스를 폭 넓게 고려하는 것이 중요하다. CAVELib, VRJuggler, NAVERLib 같은 가상 현실 라이브러리들은 가상 환경 개발자들이 그러한 컴포넌트들을 쉽게 생성할 수 있도록 기능들을 제공한다.

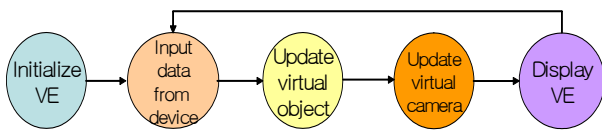


그림 2. 가상 환경의 시뮬레이션 루프

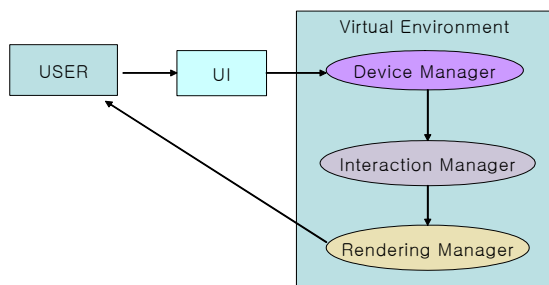


그림 3. 가상 현실 시스템의 구조

인터랙션 매니저는 가상 환경의 목적에 다양한 기능을 가질 수 있고 재정의가 될 수 있기 때문에 일반화된 많은 프리미티브를 제공하는 것과 이를 사용자에게 맞게 재정의 할 수 있도록 하는 기능이 필요하다. 본 논문에서는 그림 4 와 같이 XML 기반의 사용자 정의 인터랙션을 제안한다. 인터랙션 매니저의 예로 가상 객체 매니저와 네비게이션 매니저가 있을 수 있다.

가상 객체 매니저는 가상 환경에 존재하는 모든 객체를 관리하며 객체들의 이동, 회전에 관한 계산을 수행한다. 네비게이션 매니저는 가상 카메라의 이동 및 회전을 계산하여 일인칭 시점에서의 사용자의 네비게이션을 관리한다. 또한 3 인칭 시점에서 가상 공간의 뷰포지션을 관리한다.

```

<module>
  <scope>world,kyungju</scope>
  <name>ObjectManager</name>
  <class>ObjectManager</class>
  <dso>ObjectManager.dll</dso>
  <data>
    <Object>
      <name>KJIST</name>
    </Object>
  </data>
</module>

<module>
  <scope>view,view0</scope>
  <name>Navigator</name>
  <class>Navigator</class>
  <dso>Navigator.dll</dso>
  <data>
    <default_xyz>0.0,0.0,0.0</default_xyz>
    <default_hpr>0.0,0.0,0.0</default_hpr>
    <deltaH>0.1</deltaH>
    <deltaP>0.1</deltaP>
    <initSpeed>0.03</initSpeed>
    <minSpeed>0.01</minSpeed>
    <maxSpeed>0.1</maxSpeed>
    <deltaSpeed>1.02</deltaSpeed>
    <deltaStrafe>0.05</deltaStrafe>
    <deltaZ>0.05</deltaZ>
    <dzone>0.3,0.3</dzone>
  </data>
</module>

```

그림 4. 인터랙션 매니저(네비게이션 매니저, 객체 매니저)를 정의하는 스크립팅 파일

2.2 네트워크 인터페이스

네트워크 가상 환경은 그림 4 과 같이 원격지의 가상 환경으로부터 컨텍스트를 전송하고 전달 받는 추가적인 과정을 필요로 한다. 네트워크에 관련된 작업은 전체적인 가상 현실 프로세스에 오버헤드를 유발할 수 있다. 따라서 가상 환경을 디스플레이 과정과 병렬적으로 수행되어야 한다. 그러기 위해서 그림 5 와 같은 가상 환경 구조를 제안한다.

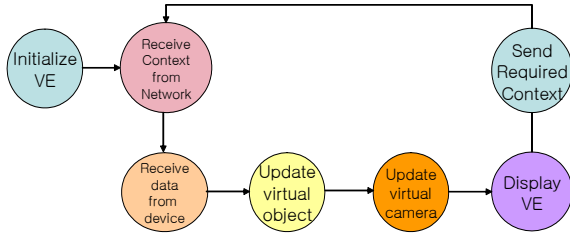


그림 4. 네트워크 가상환경의 시뮬레이션 루프

```
<VirtualSensor>
  <LocationBased_Sensor>
    <range>all</range>
    <object>KJIST_AVATAR</object>
  </LocationBased_Sensor>
</VirtualSensor>
```

그림 6. 가상 환경 모든 범위에서 해당 객체에 위치 정보를 알려주는 가상 센서 스크립트

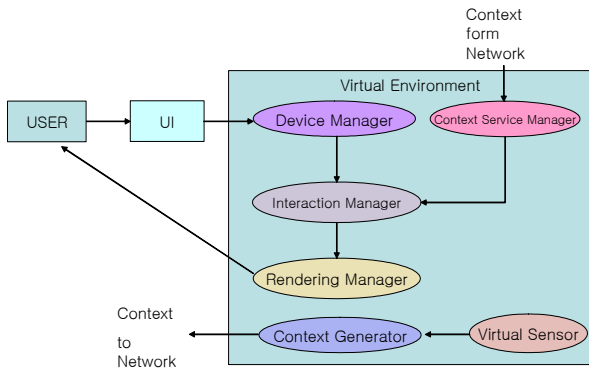


그림 5. 네트워크 가상 환경의 시스템 구조

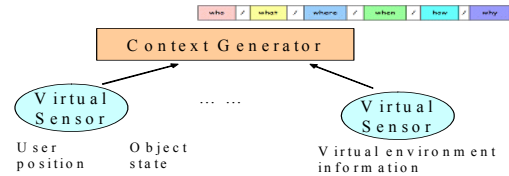


그림 7. 가상 센서의 예 및 컨텍스트 생성자 사이의 관계

2.2.1 가상 센서와 컨텍스트 생성자

로컬의 가상 환경의 변화를 다른 가상 환경과 공유하려면 가상센서와 컨텍스트 생성자를 둔다. 가상의 센서는 사용자의 필요에 의해서 필요한 가상의 환경의 위치에 놓일 수 있다. 컨텍스트 생성자는 이런 가상의 센서로부터 정보를 취합하여 이미 정의된 상호작용 전달 프로토콜인 5W1H 형식의 데이터 유형으로 변형 시켜준다. 이런 통일된 데이터 유형은 원격지에 있는 컨텍스트 서비스 매니저가 일관성있게 데이터를 처리할 수 있도록 도와준다. 그림 6 은 KJIST_AVATAR 의 위치 정보를 가상 환경 전체에 걸쳐 알려주는 가상 센서의 XML 스크립팅 예이다. 알려진 정보는 그림 7 과 같이 컨텍스트 생성자에 의해서 상호작용 전달 프로토콜 형태로 만들어져서 다른 가상 환경에 제공된다. 이때에 VR 라이브러리 마다 서로 다른 좌표 시스템을 사용하기때문에 Where 정보를 전송 할 때는 월드 좌표 시스템에 기반한 위치정보로 통일한다.

2.2.2 인터랙션 매니저 와 컨텍스트 서비스 매니저

인터랙션 매니저는 로컬 시스템에서의 역할과 같이 가상 환경에 관련된 변화를 계산하는 역할을 한다. 추가로 네트워크 가상환경에서의 인터랙션 매니저는 컨텍스트 서비스 매니저와 통신을 한다. 컨텍스트 서비스 매니저의 역할은 다른 가상 환경으로 부터 들어오는 상호전달 프로토콜을 분석하여 인터랙션 매니저를 호출 하는 것이다. 그림 8 은 Who 로 Avatar ID, How 로 Move 를 입력 받았을때 What 과 Where 를 파라미터로 객체 매니저를 호출하도록 하는 XML 스크립팅 예이다. 객체 매니저는 객체 What 을 Where 의 위치에 옮기는 일을 수행하게 된다. 표 1 은 상호작용 전달 프로토콜과 그에 대응 될 수 있는 인터랙션 매니저 클래스의 예를 보여준다.

```
<context_based service>
  <Interaction_manager>
    <name>Object_Manager</name>
    <context>
      <who>Avatar_ID</who>
      <how>Move</how>
    </context>
    <para>how, what, where</para>
  </Interaction_manager>
</context_based service>
```

그림 8. 컨텍스트 서비스 매니저를 위한 XML 스크립팅 예

표 1. 상호작용 전달 프로토콜과 인터렉션 매니저 클래스의 예

Class	Who	What	Where	When	How
Navigation Manager	User ID		Location after moving	Virtual Time	Move to
	User ID		Different Position(Foward,Backward)	Virtual Time	Move
	User ID		Different Position(Right, Left)	Virtual Time	Rotate
Object Manager	Avatar ID	Target Object	Scene Change	Virtual Time	Grab
	Avatar ID	Virtual Object	Different Position(Foward,Backward)	Virtual Time	Move
	Avatar ID		Different Position(Right, Left)	Virtual Time	Rotate

컨텍스트 서비스 매니저의 또 다른 중요한 역할은 서로 다른 플랫폼간의 동기화이다. 먼저 서로 다른 VR 라이브러리간의 데이터의 동기화이다. 각각의 VR 라이브러리마다 독립적인 좌표 시스템을 사용하여 가상 환경을 구현하기 때문에 월드 좌표를 해당 로컬 좌표로 바꾸어 주는 것이 필요하다. 이르기 위해서는 컨텍스트 서비스 매니저는 위치와 회전 변환 매트릭스를 저장하고 있어야 하면 입력된 월드좌표에 역행렬을 취해 줌으로써 원하는 로컬 좌표로 변환한다

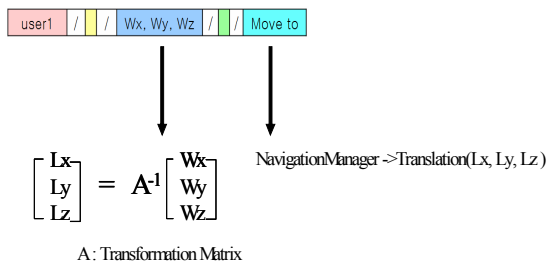


그림 9. 월드 좌표를 로컬 좌표로 변환하는 과정

다음으로는 장비의 이질성으로 인하여 생기는

사용자간의 뷰의 불일치를 동기화 시켜야 한다. 이를 위해서 컨텍스트 서비스 매니저에 상호작용 메시지의 버퍼를 둔다. 또한 상호작용 메시지의 종류를 바로 실행 가능 상호작용, 특정 시간 후에 실행해야 할 상호작용으로 나눈다. 장비에 독립적으로 계산량이 적은 상호작용은 바로 인터렉션 매니저로 전해져 수행이 되고 계산량이 많은 인터렉션은 XML 파일로 정의된 재생 지연시간(Playout delay)까지 버퍼에 저장하여 모든 사용자들이 동시에 상호작용에 참여할 수 있는 환경을 제공한다. 하지만 이러한 접근은 컴퓨팅 자원의 많은 낭비를 초래한다. 따라서 가장 컴퓨팅 파워가 좋은 플랫폼에 맞추어서 다른 플랫폼의 렌더링 수준을 동적으로 조절하는 기능도 추가되어야 한다.

3. 시스템의 구현

제안된 시스템의 유용성을 검증하기 그림 9 과 같이 e-AG 기반의 분산 가상 문화 유적 답사시스템에 적용하였다. e-AG 시스템은 Access Grid 시스템에 3D 디스플레이를 연동시킨 화상 회의 시스템이다. Access Grid 를 사용하여 서로 얼굴을 보면서 이야기하며 문화 유적에 대해서 토론할 수 있었고 3D 디스플레이를 사용하여 원격지에 떨어져있는 다수의 사용자들이 한 지역에 있는 듯한 느낌을 가지고 함께 가상 문화 유적을 답사할 수 있었다. 특정 가상 유물에 대해서는 사용자의 조작을 허용하기 때문에 공동연구의 가능성을 시험해 볼 수도 있었다.

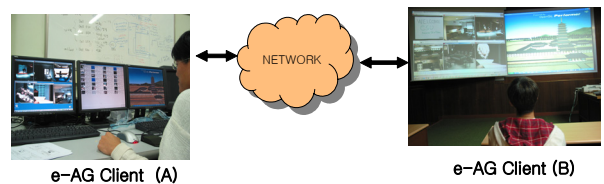


그림 9. e-AG 기반의 가상 문화 유적 답사 시스템

클라이언트 A 는 NAVERLib 기반의 가상 환경 시스템으로써 일반 데스크탑 환경에서 구현되었다. 또한 마우스와 키보드 기반의 상호작용을 지원하였다.

클라이언트 B 는 CAVELib 기반의 가상 환경 시스템으로써 passive stereo 디스플레이가 지원되는 워크스테이션 환경에서 PDA 와 같은 새로운 UI 를 사용하여 상호작용하도록 구현 되었다. 각각 모두 제안된 로컬 시스템의 요구조건을 따르기 때문에 디바이스 매니저, 인터랙션 매니저, 렌더링 매니저를 포함하고 있었다. 이런 이중 가상 환경간의 협업 환경의 구성을 위해서 사용자의 위치를 추적하는 가상 센서를 가상 환경 전체 영역에 그림 10 과 같이 선언을 하였다. 그리고 그림 11 과 같이 컨텍스트 생성자를 선언을 해준다. 컨텍스트 생성자 선언 시는 컨텍스트 정보를 보낼 위치와 소켓 타입을 명시해 주어야 한다. 또한 원격지의 정보를 받고 나서 컨텍스트 서비스 매니저가 하여야 할 작업도 정의를 해주어야 한다. 현재의 가상 문화 유적 답사 시스템은 서로 같은 Viewpoint 를 가지고 네비게이션을 하는 시나리오를 가지고 있으므로 그림 12 와 같이 컨텍스트 기반 서비스를 정의를 해준다.

```
<VirtualSensor>
  <LocationBased_Sensor>
    <range>all</range>
    <object>KJIST</object>
  </LocationBased_Sensor>
</VirtualSensor>
<VirtualSensor>
  <LocationBased_Sensor>
    <range>all</range>
    <object>KISTI</object>
  </LocationBased_Sensor>
</VirtualSensor>
```

그림 10. 사용자의 위치 정보를 알리기 위한 가상 센서

```
<Context_Generator>
  <IP>123.237.52.XX</IP>
  <Port>7000</Port>
  <Socket_Type>UDP</Socket_Type>
</Context_Generator>
```

그림 11. 컨텍스트 생성자의 XML 스크립팅

```
<context_based service>
  <Interaction_manager>
    <name>Navigation_Manager</name>
    <context>
      <who>User_ID</who>
      <how>Move to</how>
    </context>
    <para>how, where, when</para>
  </Interaction_manager>
</context_based service>
```

그림 12. Who 와 How 컨텍스트를 분석하여 네비게이션 매니저를 호출하는 컨텍스트 기반 서비스 스크립팅, 파라미터로는 Where, When 이 된다.

6. 결론 및 추후 과제

본 논문에서는 이중 플랫폼간의 협업 가상 환경을 구현하기 위한 가상 현실 프레임워크와 네트워크 인터페이스를 정의 하였다. 그리고 그 응용으로 e-AG 상에서 NAVERLib 과 CAVELib 기반으로 가상 문화 유적 답사 시스템을 구현함으로써 그 유용성을 입증하였다. 본 시스템은 일반적인 협업 환경을 제안하였기에 추후에 더 많은 응용을 통하여 인터랙션의 일반화가 이루어 져야 한다. 또한 많은 실험을 통해 안정성 및 확장성을 확고히 하고 클라이언트들간의 완전한 동기화 문제도 해결해야 할 과제이다. 또한 지금은 단순한 유니 케스트 형식의 Peer-to-Peer 구조를 이용하여 네트워크를 구현하였는데 Atlas 와 같은 네트워크 가상 현실 라이브러리를 이용하여 네트워크 효율을 증진시켜야 하겠다. 더 나아가 유비쿼터스 컴퓨팅 흐름과 같이 하여 서비스 디스커버리에 의한 브로드캐스팅 구조로 전환도 고려되고 있다.

7. 참고 문헌

[1] www.evl.uic.edu/cavern/agave/immersaview/index.html
 [2] www.evl.uic.edu/aej/macagave/walkabout.html
 [3] www.evl.uic.edu/cavern/agave/coanim.html
 [4] 조수호, 원광연 “삼차원 상호작용을 고려한 네트워크 가상현실 플랫폼” HCI 2001 학술대회, pp 255-260, 2001
 [5] C.Park, H.Ko, H.Ahn, J.Kim “NAVER : design and implementation of XML-based VR Framework on a PC cluster”, Virtual Systems and MultiMedia(VSMM) pp 967 - 975, 2002
 [6] 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