

분산 가상 환경 응용을 위한 중간 계층의 설계

고동일, 최양희

서울대학교 컴퓨터공학과

Middleware Design for Distributed Virtual Environments

Dong-il Ko. and Yanghee Choi

Seoul National Univ.

요 약

인터넷 상에서 여러 가지 멀티미디어 정보의 전송과 좀 더 사실적인 사용자 인터페이스의 제공을 목적으로 하는 분산 가상 환경(Distributed Virtual Environment)에 대한 연구가 활발해지고 있다. 하지만 기존의 연구들은 응용 중심적인 접근을 하고 있기 때문에 분산 가상 환경의 일반적인 기능을 제공해 줄 중간 계층(Middleware)에 대한 연구는 부족한 면이 있다. 이에 본 논문에서는 분산 가상 환경 응용을 제작하는데 필요한 일반적인 중간 계층을 설계하고 이를 위한 실행 - 요청 구조를 제안한다.

1. 서론

WWW(World Wide Web)의 출현은 기존의 학술적이고 폐쇄적이던 인터넷의 사용을 누구나 사용할 수 있고 개방적인 인터넷으로 바꾸어 놓았다. 하지만 WWW는 원래의 목적 자체가 서로 분산된 문서 정보의 효율적 검색 및 이용이었기 때문에 문서 정보 이외의 다양한 멀티미디어 정보의 이용이나 사용자 간의 대화 가능 특성(interactivity)이 부족한 단점이 있었다. 이에 이런 WWW의 단점을 극복하고 인터넷 상에서 여러 가지 멀티미디어 정보를 전송하고 대화형 특성을 제공하기 위한 많은 연구와 노력이 진행되어 왔다.

분산 가상 환경(DVE Distributed Virtual Environment)도 이런 연구의 일환으로 시작된 연구이다. 분산 가상 환경이란 '서로 분산되어 있는 여러 사용자들이 네트워크를 통해 일련의 공유된 환경에서 상호 작용을 가능하게 하는 기반 기술'이라고 정의할 수 있다. 분산 가상환경은 다음과 같은 응용 분야를 가지고 있다.

- 군사적 목적의 시뮬레이션(Military Simulation)
- 컴퓨터 이용 협동 작업(CSCW: Computer Supported Cooperative Work)
- 원격 강의 및 교육(Remote lecture, learning)
- 분산 가상 실험(Distributed Virtual Experiment)
- 네트워크 게임(Networked Computer Game)

기존에 여러 분산 가상환경 응용에 대한 많은 연구가 있었으나, 대부분의 연구들은 특정 응용에 대해 특화된 경우가 많아 분산 가상환경에 적합한 일반적인 요구사항과 중간 계층(Middleware)을 정의하는 데는 부족한 면이 많았다. 이 논문은 이런 분산 가상환경 응용을 제작하는 데 적합한 중간 계층을 설계하는 것을 목적으로 하고 있다. 2 장에서는 기존의 분산 가상환경 응용을 구조에 대한 문제들을 중심으로 이야기 하겠다. 3 장에서는 이런 문제들을 해결하기 위해 새롭게 실행 - 요청 구조(Execution - Request Architecture)구조에 대해 논하겠

다. '들의 구성에 대해 설명한다.

2. 분산 가상환경의 구조(Architecture)와 기존 연구

기존의 분산 가상환경 응용에서 말해지는 구조는 크게 두 가지로 말할 수 있는데 하나는 중앙 서버 구조(Centralized Server Architecture)이고, 다른 하나는 완전 분산 구조(Full Distributed Architecture)이다.

중앙 서버 구조는 가상 환경 내의 모든 사용자들이 중앙의 서버에 접속을 하여 이를 통해 다른 사용자들과 상호 작용을 하며, 사용자들의 모든 데이터는 중앙의 서버에서 관리가 되는 구조이다. 이는 사용자 간의 동기화를 구현하기 편하고, 응용의 구현이 간단하며, 보안 문제를 해결하기가 수월한 이점이 있다. 하지만, 중앙 서버 구조의 가장 큰 문제는 중앙 집중에 따른 확장성의 문제이다. 사용자의 요청이 하나의 서버에 집중되기 때문에 네트워크 대역폭의 문제가 커지게 된다. 만일 네트워크 대역폭의 문제가 해결된다 하더라도 이보다 더욱 심각한 것은 서버의 프로세서에 대한 부하이다. 현재 분산 가상 환경에서는 그 기반 구조에 대한 요구사항이 하드웨어의 발전 속도를 훨씬 증가하고 있다.

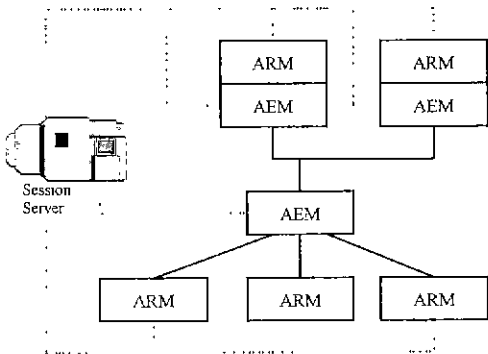
서버 중심 구조는 주로 상업적 응용과 작은 규모의 응용에 이용되어 왔다. 이는 서버 중심 구조 자체가 가지는 확정성 한계에 대한 이유도 크지만, 상업적 응용의 경우 이윤을 위한 과금(billing)과 보안에 대한 고려도 무시 못할 요소이다. 서버 중심 구조의 가장 대표적인 예로 MUD(Multi User Dungeon)을 들 수 있다. MUD는 70년대 후반에 대학을 중심으로 만들어졌으며, 주로 텍스트 위주의 인터페이스를 사용하였다. 이를 그래픽 기반의 인터페이스로 바꾼 그래픽 MUD는 최근에서야 실용화 되었으며, Origin사의 울티마 온라인(Ultima On-line)이 그 대표적인 예로 역시 서버 중심의 구조를 채택하고 있다. 다른 중앙 서버 구조를 사용한 예로는 Virtual Space Teleconferencing System(Vistel)[4]을 들 수 있다.

완전 분산 구조는 가상 환경내의 모든 사용자들은 관심 있는 다른 사용자와 직접 통신을 하며, 모든 데이터는 사용자 스스로 관리하게 된다. 이는 부하가 집중되는 서버가 없기 때문에 필요한 네트워크 대역폭이 줄고, 확장성이 용이한 이점이 있다. 하지만, 각 사용자간의 서로 다른 네트워크 지연과 연산 능력은 사용자간 동기를 유지하기 어렵게 한다. 또한 각 사용자마다 가상 세계에 대한 데이터를 가지고 있고, 이의 일관성(consistency)을 유지하는 것이 문제가 된다. 하지만 완전 분산 구조는 중앙 서버 구조가 가지는 본질적인 확장성 제약 문제에 의하여, WAN 환경의 분산 가상 환경 연구에서 선호되어 왔다.

완전 분산 구조를 이용한 대표적인 예는 DIVE (Distributed Interactive Simulation)[8], MASSIVE (Model, Architecture, and System for Spatial Interaction in Virtual Environment)[6], NPSNET[5] 등이 있다.

3. 실행 - 요청 구조

제안하는 분산 가상 환경을 위한 전체적인 구조는 다음 그림과 같다.



위의 그림에서 세션 서버(Session Server), ARM, AEM 등이 바로 분산 가상 환경을 위한 중간 계층 구성 요소들이다.

세션 서버는 서로 독립된 세션의 생성(creation), 참여(join), 종료(expire)등을 위한 정보를 다른 네트워크 개체에 제공한다. ARM과 AEM을 이 세션 서버를 이용하여, 세션 안내, 세션 초기화 등의 서비스를 분산 가상 환경 응용에 제공할 수 있다.

AEM(행동 실행 모듈, Action Execution Module)은 분산 가상 환경에서 일어나는 행동사건(Action) 들을 받아 이를 동기화 시키고 처리하며, 결과를 관련된 다른 네트워크 개체들에게 전달하는 역할을 한다. AEM의 중요한 특징 중 하나는 어떤 분산 가상 환경 개체에 대하여 신뢰할 수 있는 저장 장소를 가진다는 것이다. 하나의 분산 가상 환경에서 어떤 개체에 대한 일관성(consistency)을 유지한다는 것은 그 개체에 대한 정보를 신뢰 가능 저장소(RES Reliable Entity Storage)의 정보에 일치시키는 것을 말한다.

ARM(행동 요청 모듈: Action Request Module)은 AEM에 어떤 행동을 요청하는 부분이다. ARM이 하는 요청은 어떤 분산 가상환경 개체의 요청으로, 하나의 ARM

은 하나의 분산 가상환경 개체에 관계하게 된다. 이는 AEM과 통합되어 내부 인터페이스로 AEM에게 이를 요청할 수 있으며, 네트워크를 이용하여 요청할 수도 있다. 하나의 ARM은 반드시 하나의 AEM과 연결(connection)을 이루게 된다. 하지만, 하나의 ARM은 동시에 여러 개의 AEM과 연결될 수는 없다. 어떤 ARM이 AEM에 연결되어 있다는 말은 그 ARM이 관여하는 분산 가상환경 개체의 RES가 그 AEM에 존재한다는 뜻이다.

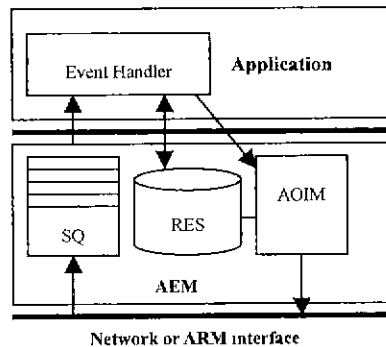
중간 계층 구성 요소들은 하부 네트워크 계층과, 상부 응용 계층 사이에 위치하며, 분산 가상 환경 응용을 제작할 때 필요한 일반적인 서비스, 즉, 세션의 시작, 종료, 참여를 위한 정보의 제공, 확장성 제공, 동기화, 가상 환경 데이터의 관리 등의 기능을 제공해야 한다.

이 구조를 실행 - 요청 구조라고 정의한다. 이는 분산 가상 환경 내에서 어떤 행동을 요청하는 것과 이를 수행하는 쪽을 분리하여 생각하는 것을 그 핵심으로 하고 있다.

AEM과 ARM은 하나의 모듈로 구성될 수도 있고, 서로 다른 모듈로 분리할 수도 있다. 모든 AEM과 ARM이 하나의 모듈로 구성 되어 있다면, 이는 기존의 완전 분산 구조와 같이 생각할 수 있다. 만일 하나의 AEM과 ARM이 분리되어 있다면 이는 기존의 중앙 서버 구조와 같이 생각할 수 있다. 실행 - 요청 구조는 이렇게 기존의 분산 가상 환경 구조를 통합하여, 혼합(Hybrid) 구조를 구성하는 것을 가능하게 한다. 이 중간 계층 요소간의 인터페이스만 통일 되어 있다면, 응용 제작자는 응용의 구조에 신경을 쓰지 않고 응용을 제작할 수 있으며, 응용의 목적, 네트워크 구조와 성능, 사용자 시스템의 성능 등을 고려하여 필요한 분산 가상 환경의 구조를 재 설정할 수 있다. 또한 이와 같은 동적 구조 조정을 통해 분산 가상 환경의 확장성을 확보하기 용이한 이점도 있다.

4. 행동 실행 모듈(AEM: Action Execution Module)의 구성

AEM은 다음과 같이 구성된다.



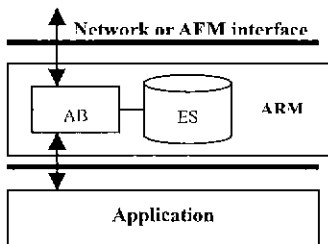
SQ (Synchronization Queue): 네트워크나 별도의 ARM과의 인터페이스에서 행동 메시지(Action Message)를 받는 역할을 한다. SQ는 동기화를 제공하기 위한 일련의 작업을 수행한 후 이벤트를 응

용의 Event Handler 에게 발생시키게 된다

- ✓ **RES (Reliable Entity Storage):** 분산 가상환경에서 어떤 개체의 속성에 대한 신뢰할 수 저장 장소를 제공한다. 한 분산 가상환경의 개체는 단 하나의 RES 를 가진다. 바로 이 RES 의 정보만이 전체 분산 가상환경상의 그 개체의 상태를 나타낼 수 있으며 분산 가상환경은 이 RES 의 복사본들을 갖게 만들기 위해 메시지를 주고 받는다고 할 수 있다.
- ✓ **AOIM (Area Of Interest Manager) :** 각각의 개체는 그 속성으로 AOI(관심 영역 Area Of Interest)를 가진다. 이는 한 세션의 모든 개체가 이 개체의 정보에 관심을 가질 필요는 없기 때문에 이 개체의 정보가 필요한 다른 개체들의 집합을 각각의 개체별로 관리하는 것이다. AOIM 은 응용의 Event Handler 가 처리한 개체의 행동을 바탕으로 이 AOI 를 새롭게 구성하며, 그 결과를 필요하다면, AOI 를 바탕으로 다른 개체들에게 전달하게 된다.
- ✓ **사건 처리기(Event Handler) :** 이는 엄밀히 말해 AEM 의 구성 요소는 아니다. 응용에 구현되는 사건 처리기는 AEM 이 발생시킨 사건을 처리하고 그 결과를 AOIM 에게 알려줄 의무가 있다. 사건 처리기는 정의된 인터페이스를 이용하여, RES 의 정보를 사용할 수 있다.

5. 행동 요청 모듈(ARM: Action Request Model)의 구성

ARM 은 다음과 같이 구성된다.



- ✓ **AB (Action Broker) :** AB 의 역할은 응용에서 오는 Action 요청을 관계된 AEM 에게 네트워크나 AEM interface 를 통해 요청하는 것이다.
- ✓ **ES (Entity Storage) :** 이것은 분산 가상환경 개체들의 속성 저장소이다. RES 와는 달리 ES 의 내용은 복사본으로 ARM 에 관계된 응용이 이용하기 위해 사용된다. 단일 AEM 과 ARM 이 하나로 구현되는 환경이라면, ES 와 RES 는 하나로 구현될 수 있다.

6. 결론

기존의 응용 중심적인 분산 가상 환경에 대한 접근에서 벗어나 일반적인 분산 가상 환경 응용 제작을 위한 중간 계층의 제작이 필요하다. 이를 통해 유사한 분산 가상 환경들의 기능을 통합하고, 분산 가상 환경 응용의 개발을 촉진 시킬 수 있다. 분산 가상 환경 중간 계층은 다음과 같은 시시스를 응용에 제공할 수 있어야 한

다

- 분산 가상 환경 세션의 개시, 종료, 참여를 위한 정보를 제공해야 한다.
- 각 분산 가상 환경 개체에 대한 신뢰할 수 있는 정보를 제공해야 한다.
- 각 분산 가상 개체간의 동기화를 지원할 수 있어야 한다.
- 분산 가상 환경의 확장성은 보장하기 위한 방법이 제공되어야 한다.

이를 위해 본 논문에서는 실행 - 요청 구조를 바탕으로 한 분산 가상 환경 중간 계층을 제안하였다. 이를 이용하면 위에서 논한 중간 계층 서비스를 응용에 제공할 수 있으며, 동적인 분산 가상 환경 구조의 조정을 통해 확장성을 제공하기도 용이하다.

향후에는 이를 이용하여 분산 가상 환경을 위한 실험 환경을 조성하고, 실험 응용을 제작하여 분산 가상 환경을 위한 다른 요소, 즉, 응용 프로토콜 가상 개체의 묘사 방법, 성능 측정을 위한 특성 정의 등의 연구가 필요할 것이다.

7.참고 문헌

- [1] Bernie Roshu "Distributed Virtual Reality - An Overview" <http://sunce.uwaterloo.ca/~broehl/distrib.html>
- [2] Richard C. Waters John W. Barus. 'The Rise of Shared Virtual Environments' IEEE Spectrum March 1997
- [3] Michael R. Macedonia, Michael J. Zyda "A Taxonomy for Networked Virtual Environments" IEEE Multimedia January-March 1997
- [4] J. Ohya et al., "Real-time Reproduction of 3D Human Images in Virtual Space Teleconferencing" Proc VRAIS 93, IEEE Press Piscataway NJ, 1993
- [5] Macedonia, Michael R., et al. "NPSNET: A Network Software Architecture for Large-Scale Virtual Environment" Presence: Teleoperators and Virtual Environments, 3(4), Fall 1994
- [6] C. Greenhalgh and S. Benford, "MASSIVE: A Collaborative Virtual Environment for Teleconferencing" ACM Trans. Computer-Human Interaction, Vol 2, No 3, 1995
- [7] Karl O'Connell, Vinny Cahill, Andrew Condon, Stephen McGerty, Gradimir Starovic and Brendan Tangney, Distributed System Group Dept. of Computer Science, Trinity College Dublin Ireland "The VOID Shell - A Toolkit for Development of Distributed Video Games and Virtual Worlds"
- [8] Olof Hagsand Swedish Institute of Computer Science. "Interactive Multiuser VEs in the DIVE System" IEEE Multimedia Spring 1996
- [9] Sandeep K. Singhal "Effective Remote Modeling in Large-Scale Distributed Simulation and Visualization Environments" Ph.D Paper of Stanford Univ. August 1996