

모빌 에이전트를 이용한 분산 가상 환경 관리자의 설계

°장진윤*, 이대규*, 탁진현*, 이세훈**, 왕창중*
*인하대학교 전자계산공학과, **인하공업전문대학 전자계산기과

Design of Networked Virtual Environment Manager Using Mobile Agent

°J. Y. Chang*, D. K. Lee*, J. H. Tak*, S. H. Lee**, C. J. Wang*

*Dept. of Computer Science & Engineering, Inha University

**Dept. of Computer Engineering, Inha Technical College

요약

기존의 다중 사용자 지원 3차원 공간 정보에서의 가상 공간 관리 방식의 특징은 각 참여자에서 발생하는 이벤트를 중앙 서버에 전달해서 반영된 내용을 다시 각각의 참여자에게 전달하는 방식을 사용한다. 이러한 경우 가상 환경에 대한 상태 정보를 중앙 서버에서 관리하므로 및 일관성 유지가 용이하다는 장점이 있으나 네트워크에 과중한 부담을 주며, 확장성이 부족하다는 단점이 있다.

이 논문에서는 3차원 가상 환경에서 발생하는 이벤트의 형태를 아바타의 이동 이벤트와 그밖의 이벤트로 구분하고, 가장 빈번히 발생하고 신뢰성이 비교적 적게 요구되는 아바타의 이동에 대한 이벤트 전달은 모빌 에이전트간의 통신으로 전달함으로써 중앙서버의 부담을 줄여주며, 일관성을 효과적으로 유지할 수 있는 모빌 에이전트 기반 가상 공간 관리자를 설계하였다.

1. 서론

컴퓨터와 웹을 중심으로 한 네트워크의 급속한 발달은 원격 사용자들간의 공동 작업을 지원하는 기존 CSCW에 대한 연구를 웹을 기반으로 하는 3차원 가상 공간으로의 확장하는 연구로 활발히 진행되고 있다. 특히, 3차원 가상공간은 공동 작업을 위한 상호 작용을 보다 현실감 있도록 지원할 수 있다는 측면에서 각광 받고 있다[1].

이러한 상호작용을 지원하기 위한 3차원 공간으로 확장된 상태 정보의 인식(awareness)은 더욱 복잡하여 지고, 네트워크의 부하를 가중시키므로 이를 효과적으로 처리할 수 있는 연구가 필수적이다[2]. 다중 사용자를 지원하는 3차원 공간 정보에 대한 기존의 관리 방식은 클라이언트-서버 구조와 Peer-to-peer 구조로 나눌 수 있다. 클라이언트-서버 구조는 가상 환경에 대한 상태 정보를 중앙 서버에서 관리하므로 및 일관성 유지가 용이하다는 장점이 있으나 네트워크에 과중한 부담을 주며, 확장성이 부족하다는 단점이 있다. 반면, Peer-to-peer 구조는 중앙 서버의 문제점을 해결할 수 있으나, 참여자의 호스트에서 정보를 관리함으로써 상태정보의 일관성을 유지하는데 어려움이 있다.

따라서, 이 연구에서는 3차원 가상 환경에서 가장 빈번히 발생하고 신뢰성이 비교적 적게 요구되는 아바타의 이동에 대한 이벤트 전달은 모빌 에이전트간의 통신으로 전달함으로써 중앙서버의 부담을 줄이고 가상 환경의 상태 변화에 대한 정보를 관리하는 가상 환경 관리자를 설계함으로써 일관성을 효과적으로 유지할 수 있도록 한다. 가상 환경 관리자의 설계를 위하여 IBM사의 모빌

에이전트 서버인 Aglet[3]을 사용하고, 3차원 공간을 웹상에서 표현하기 위하여 VRML과 EAI(External Authoring Interface)를 사용한다.

2. 기존 시스템의 네트워크 구조

기존의 다중 참여자 가상현실 시스템은 네트워크 구조에 따라 서버-클라이언트 구조와 peer-to-peer 구조로 분류할 수 있다[4, 5, 6].

서버-클라이언트 구조는 가상 공간에 관한 정보는 모두 서버에서 관리하고 참여자는 서버와의 통신을 통해 공유 가상환경에 접근하고 서버의 관리 하에 공유 가상환경에 조작을 가하는 모델이다. 이 모델은 설계 및 구현이 쉽고, 한 곳에서 가상 환경의 조작을 관리하므로 일관성 유지에 유리하다. 하지만 참여자의 증가에 따라 서버에 과부하가 발생하는 단점이 있다[5].

Peer-to-peer 구조는 각 참여자 노드가 다른 참여자에 관한 정보를 개별적으로 관리하며 변경되는 내용은 네트워크를 통하여 다른 참여자들에게 직접 전달되는 방식이다. 전달해야 하는 정보들이 서버로 집중되지 않으므로 병목현상을 해결할 수 있지만, 다수의 가상세계 정보간의 일관성 유지가 어려운 점이 있다[4, 6].

기존의 다수 참여자 시스템 중에서 서버-클라이언트 구조를 사용한 시스템에는 AVIARY, Co-CAD가 있고, peer-to-peer 방식에는 DIS, NPSNET 등이 있다. 표 1은 두 구조를 간단히 비교한 것이다 [4, 5, 6].

하지만 [표 1]에서 나열한 시스템들도 일관성 유지나 서버의 과부하를 완전히 해결하지는 못하였다.

[표 1] 네트워크 구조의 비교

입력성 유지	쉬움	어려움
대용량 전송	생김	없음
경쟁 관리	서버 사용	복제 관리
대표적 시스템	AVIARY, Co-CAD	DIS, NPSNET

이 논문에서는 클라이언트-서버 방식을 기반으로 하고 서버의 과부하를 줄이기 위해 통신 방식에 멀티캐스트 방식을 도입하여 시스템을 설계하고 모빌 에이전트를 도입하여 시스템의 효율성을 높이도록 하였다.

3. 가상 환경 관리자의 설계

이 논문에서는 다중 참여자의 공동작업이 이루어지는 분산 가상환경을 유지하고 관리하는 가상환경 관리자를 설계한다. 가상 공간에서 공동작업이 그룹단위로 이루어 질 때 하나의 가상공간이 형성되며 설계되는 가상 환경 관리자는 웹상에서 VRML로 표현되는 가상 공간에서 참여자들의 공동작업이 효과적으로 수행될 수 있도록 관리한다.

3.1 가상 환경 관리자의 구성

가상 환경 관리자의 구성은 그림 1과 같다.

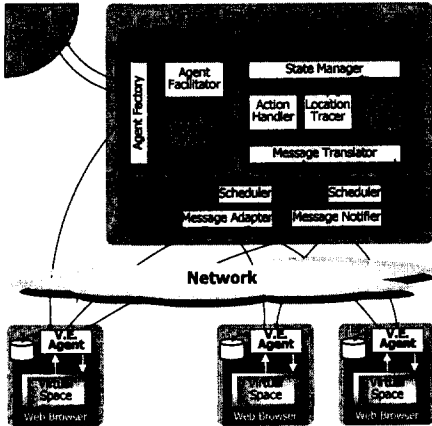


그림 1 가상 환경 관리자 구성도

가상 환경 관리자는 참여자의 가입, 탈퇴에 따른 가상 환경의 변화 및 그룹 관리자[7]에 의해 전달되는 그룹 정책에 따라 참여자에 의해 발생하는 조작, 생성, 이동, 삭제 등을 관리하고 참여자들에게 전달한다. 또한, 참여자의 가상 환경 에이전트와 통신하여 참여자들 사이에 발생하는 이벤트의 상충여부 판단과 동시 조작 문제를 조정하는 역할은 한다. 가상 공간에서 일어나는 이벤트에 대한 실제 연산은 클라이언트의 가상 환경 에이전트에 의해 처리되고 서버측의 가상 환경 관리자는 가상 공간 전체에 대한 정보 유지와 이벤트의 전파를 담당한다.

3.2 에이전트 관리자 모듈

• 에이전트 팩토리

에이전트 팩토리는 그룹관리자로부터의 요청에 따라 가상환경 에이전트의 생성과 소멸을 담당한다. 그룹관리자에 의해 승인되는

참여자의 가입에 대하여 그룹관리자로부터 전달된 참여자의 정보와 객체 조작 권한 및 상태 관리자가 유지하고 있는 가상 공간의 정보를 가지고 참여자에 특성화된 가상 환경 에이전트를 생성시킨다. 생성된 가상 환경 에이전트는 에이전트 퍼실리테이터에 등록된 후 참여자의 호스트로 이주된다. 그림 2는 가상 에이전트의 생성과정을 보여준다.

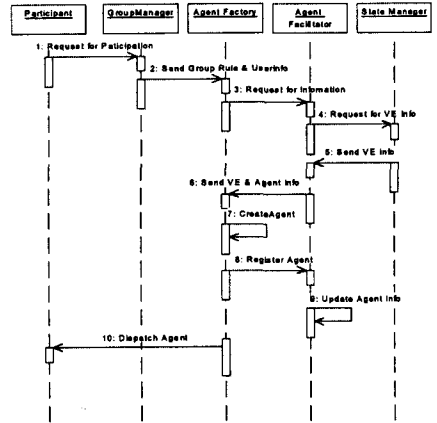


그림 2 가상 환경 에이전트의 생성

가상 환경 에이전트의 소멸 시, 팩토리는 가상 환경 에이전트로부터 작업종료 메시지를 받고 가상 환경 에이전트를 복귀시켜 상태정보와 유지하고 있던 로그정보를 퍼실리테이터에게 전달한 후 소멸시킨다.

• 에이전트 퍼실리테이터

에이전트 퍼실리테이터는 해당 그룹에 속하는 그룹에 속한 가상 공간 에이전트의 정보를 유지하며 그룹 정책과 개인 권한 등급 및 가상 공간에서 참여자를 대리하는 아바타의 상태 정보 등을 관리한다. 그림 3은 에이전트 정보에 대한 자료구조를 설명한다.

AgentID	Agent Proxy	Participant ID	Participant Name	Avatar Info.	Authorization	IsActive	
		AvatarID	SourceURL	Behavior	Location	Direction	Velocity

그림 3 에이전트 정보 데이터

아바타의 정보는 소스코드의 위치와 현재 행위와 위치, 그리고 행위가 이동인 경우 방향과 속도 정보로 구성된다.

• 가상 환경 에이전트

에이전트 관리 모듈로부터 가상 공간의 상태 정보와 존재하는 다른 가상 환경 에이전트의 정보를 가지고 이주되는 가상 환경 에이전트는 참여자의 호스트에 상주하면서 참여자의 행위에 의해 발생하는 이벤트와 외부에서 전달되는 이벤트를 처리한다. 즉, 아바타를 포함하는 객체의 이동정보를 dead-reckoning 알고리즘[4]을 사용하여 메시지화하고 EAI를 사용하여 VRML로 표현되는 가상 공간에서 발생하는 이벤트를 처리하고 전파되어야 하는 이벤트를 필터링하여 메시지화한다. 이때, 아바타의 이동에 대한 이벤트는 가상 환경 관리자를 통하여 전파하지 않고 다른 참여자의 호스트에 있는 가상 환경 에이전트와 직접 통신함으로써 서버의 부하를

줄인다. 아바타 이동 이벤트는 가상 공간의 상태정보 유지를 위하여 가상 환경 관리자에게 보내어, 관리자에 의해 직접 전파되지 않는다.

가상 환경 에이전트는 전달되는 메시지들 중 참여자의 관심(interest) 부분에 해당하는 메시지를 필터링하고 전체 가상 공간에서 LOD(Level Of Detail)와 같은 기준에 의해 영역에 해당되는 이벤트만을 반영시키고 나머지 영역에서 발생하는 이벤트는 그 정보만을 유지하는 방법으로 멀티캐스팅되는 메시지 처리의 부하를 줄인다.

3.3 가상 공간 관리 모듈

가상 공간 관리 모듈은 가상 환경 에이전트에서 전달되는 메시지를 기반으로 가상 환경의 현재 상태를 유지한다. 그러나, 가상 공간 관리 모듈은 동적으로 생성, 삭제, 조작되는 객체를 관리하기 위하여 그림 4 와 같은 정보를 관리한다.

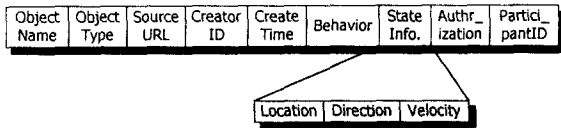


그림 4 가상 공간의 동적 객체 정보 테이블

객체이름은 가상 공간에서 객체를 식별할 수 있는 유일한 식별자이며, 객체타입은 개인 객체(private object)와 개인-공유 객체(private-shared object), 공유 객체(shared object) 등의 타입정보를 기술한다. 권한은 객체를 핸들링 할 수 있는 권한정보를 기술하며 이는 그룹관리자로부터 전달된다. 그 외에 객체의 코드정보가 있는 URL 정보, 객체 생성자와 생성시간, 현재 객체의 핸들을 가지고 있는 참여자의 ID 정보를 유지한다. 객체에 대한 조작을 관리하는 액션 핸들러는 객체의 동적인 삽입, 삭제를 수행한다. 객체 조작 관리란 조작을 위한 연산을 수행하는 것이 아니라 객체 사용자간의 동시성 제어와 권한 관리 역할을 담당한다. 또한, 객체의 이동 상태 정보는 dead-reckoning 알고리즘으로 표현하므로 객체에 대한 위치, 방향, 속도로 표시한다. 위치 추적기는 이러한 객체 이동정보를 시간에 따른 경로로 변환하여 상태관리자에게 전달한다. 상태관리자는 액션 핸들러와 위치 추적기를 통해 전달되는 정보를 서버가 관리하는 공간 정보에 반영시킨다. 상태관리자는 단지 현재 상태정보만을 관리하며, 이전의 상태정보는 그룹관리자에 의해 관리된다.

3.3 메시지 전송 모듈

메시지 전송 모듈은 가상 공간 에이전트와의 통신을 수행한다. 가상 공간 관리자에 의해 전달된 메시지를 해당 가상 환경 에이전트에게 전달한다. 가상 환경 관리자로 부터 가상 공간 에이전트로 전달되는 메시지는 아바타의 상태 정보를 제외한 공유되는 객체에 대한 이벤트 처리에 관한 것이므로 발신자 기반의 Reliable Multicast Protocol(RMP)을 사용한다.

3.4 메시지 포맷

참여자들이 메시지 관리 모듈에 보내는 메시지의 포맷은 그림 5 와 같다.

Source Proxy	Target Proxy	Object Name	Event Type	Time Stamp	State Info.	Parameter
--------------	--------------	-------------	------------	------------	-------------	-----------

그림 5 메시지 포맷

소스 프락시와 타겟 프락시는 메시지의 전송측과 수신측을 표시하며, 이벤트 타입은 객체가 발생시키는 이벤트의 종류를 나타낸다. 타임스탬프는 이벤트의 발생시간 즉, 메시지의 전송 시간이고, 위치정보는 이벤트가 일어난 위치이다. 이벤트 타입은 [표 2]와 같다.

[표 2] 지원되는 이벤트 타입

AddObject	객체의 삽입
RemoveObject	객체의 삭제
MoveObject	객체의 이동
JoinAvatar	아바타의 가상 환경 참여
MoveAvatar	아바타의 이동 이벤트
AvatarBehavior	아바타의 행위 이벤트
LeaveAvatar	아바타의 가상 환경 탈퇴
UserDef	사용자 정의 이벤트

4. 결론

이 연구에서는 모빌 에이전트를 이용하여 다중 사용자의 협업을 지원하는 웹 기반의 분산 가상 환경관리자를 설계하였다.

설계한 가상 환경 관리자는 그룹에 참여한 사용자들의 가상 공간으로의 참여와 탈퇴, 가상 공간 안에서의 상호작용에 대한 일관성을 유지함으로써 중앙 집중식 방식의 장점을 살렸다. 또한, 참여자들간의 통신구조를 이원화 함으로써 네트워크에 대한 부담을 줄여 중앙 집중형 방식의 문제점을 해결하고, 분산 제어 방식이 가질 수 있는 사용자의 부담을 최소화 시키고 가상 공간의 일관성 유지문제를 해결하였다.

참고문헌

- [1] 성운재, 심재한, 원광연, "다중 참여자 네트워크 가상현실 시스템을 위한 복수 멀티캐스트 통신구조," 한국시뮬레이션학회 논문지, 7 권, 1 호, 1998
- [2] John A. Mariani, "SISCO: Providing a Cooperation Filter for a Shared Information Space," Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, 1997.
- [3] D.B. Lange, Programming and Deploying Java Mobile Agents with Aglets, Addison-Welsley, 1998
- [4] Macedonia, M.R., M. J. Zyda, D.R. Pratt, T.Barham, and S.Szewitz, "NPSNET: A Network Software Architecture for Large-scale Virtual Environments," PRESENCE, Vol. 3, No. 4, MIT press, Fall 1994.
- [5] Gisi, M. A., Cristiano Sacchi, "Co-CAD: A Collaborative Mechanical CAD System," PRESENCE, Vol. 3, No. 4, MIT press, Fall 1994.
- [6] snowdon, D.N., Adrian J. West, "AVIARY: Design Issues for Future Large-Scale Virtual Environments," PRESENCE, Vol. 3, No. 4, MIT press, 1994.
- [7] 이승근, 탁진현, 이세훈, 왕창중, "분산 가상 환경을 위한 모빌 에이전트 기반 참여자 그룹관리," 한국정보과학회 추계 학술논문발표집, Vol. 26, No. 2, 1999.10.