

# 다중 참여자를 지원하는 VRML 기반의 어플리케이션 공유 엔진 설계

○이원재\*, 탁진현\*, 이세훈\*\*, 왕창종\*

\*[won75, tak]@selab.inha.ac.kr, cjwangse@inha.ac.kr, 인하대학교 전자계산공학과  
\*\*seihoon@truc.inhac.ac.kr, 인하공업전문대학 전자계산기과

## Design of VRML based Applications Sharing Engine for Supporting Multi-Participants

○W. J. 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

공동작업을 위한 어플리케이션 공유는 여러 시스템에 분산되어 있는 개발 도구와 어플리케이션을 공유하여 다수의 참여자가 공간적으로 제약 받지 않고 공동 개발할 수 있도록 한다. 그러나 가상공간에 개념을 둔 CSCW에 기존의 어플리케이션 공유 방식을 적용하기에는 3차원이라는 공간적인 요소를 고려하지 않은 문제점을 가지게 된다. 또한 기존의 CSCW에 비해서 가상 공간에 기반을 둔 CSCW 어플리케이션 개발은 가상 공간과 다중 참여자를 고려해서 개발하여야 하기 때문에 개발자에게 매우 어려운 작업이다.

따라서, 이 논문에서는 다중 참여자를 지원하는 VRML 기반 어플리케이션 공유 엔진을 설계하였다. 설계된 VRML 기반 어플리케이션 공유 엔진은 참여자 사이트로 에이전트를 이동시켜 웹 브라우저 내의 VRML 가상 공간에서 참여자의 어플리케이션 대한 3차원상의 이벤트를 2차원적인 상대값으로 바꾸고 실제 어플리케이션상의 이벤트로 변환하는 매커니즘을 도입하여 VRML 가상공간에서 어플리케이션을 공유를 가능하게 하였다. 또한 참여자 이벤트를 처리하는 스케줄러를 도입하여 참여자의 이벤트를 순서적이고 참여자 간의 충돌을 해결할 수 있었으며 가상환경이 VRML로 구성되기 때문에 운영체제에 독립적이고 웹상에서 어플리케이션 공유가 가능하였다. 그리고 가상환경을 위한 CSCW 응용 개발에 대한 오버헤드를 줄이고 기존 윈도우 어플리케이션을 가상공간에 사용함으로써 기존 어플리케이션의 재사용성을 높일 수 있었다.

### 1. 서론

컴퓨터 성능의 급속한 발전과 고속 네트워크 기술의 발달로 인한 오늘날의 컴퓨팅 환경이 개방화됨에 따라 CSCW 응용은 화상 회의 시스템, 상호 협력적 원격 교육, 원격 진료, 그룹 게임 분야 등으로 실생활의 거의 모든 분야로 확대되고 있다[1]. 특히, 컴퓨팅 능력의 발달로 인한 가상 공간 개념을 바탕으로 둔 CSCW는 기존의 CSCW에서 보다 원격지의 참여자들끼리의 상호작용을 보다 현실감 있도록 지원할 수 있다는 측면에서 각광 받고 있다[2]. 그 중에서 어플리케이션 공유는 여러 시스템에 분산되어 있는 개발 도구와 어플리케이션을 공유하여 다수의 참여자가 공간적으로 제약 받지 않고 공동 개발할 수 있도록 지원하는 주요 수단이다 [3,4,5].

그러나 기존의 어플리케이션 공유 방식이 가상공간 개념을 둔 CSCW에 적용되기에는 가상공간이라는 3차원 공간 개념을 고려하지 않고 설계되었기 때문에 적합하지 않다. 또한 기존의 CSCW에 비해서 가상 공간에 기반을 둔 CSCW 어플리케이션 개발은 가상 공간과 다중 참여자를 고려해서 개발하여야 하기 때문에 개발자에게 매우 어려운 작업이다.

따라서, 이 논문에서는 기존의 어플리케이션 공유 방식을 가상 공간상에서 도입하여 어플리케이션 공유를 가능하게 하고 다중 참여자를 지원하는 VRML 기반 어플리케이션 공유 엔진을 설계한다. 설계되는 VRML 기반 어플리케이션 공유 엔진은 다수의 참여자들이 가상 환경상에서 윈도우 어플리케이션의 뷰를 공유하며 작업할

수 있는 환경을 제공한다. 또한 웹 브라우저 내에 VRML로 구성된 3차원 공간에서 발생하는 참여자의 어플리케이션에 대한 이벤트를 감지하는 에이전트를 클라이언트 사이트로 이동시킴으로써 3차원상에서 발생하는 이벤트를 실제 어플리케이션에서 발생하는 이벤트로 변경하여 가상 공간에서 어플리케이션 공유를 가능하도록 한다. 그리고 기존의 윈도우 어플리케이션을 가상공간에 사용함으로써 기존 어플리케이션의 재사용성을 높일 수 있을 것이다.

### 2. 기존의 어플리케이션 공유 시스템 비교

컴퓨팅 환경에서 어플리케이션을 공유하고자 하는 시도는 오래 전부터 이루어져 왔다. 어플리케이션 공유란 엑셀이나 워드프로세서와 같이 개인용으로 만들어진 어플리케이션을 수정 없이 화상회의와 같은 공동작업에 이용하는 것이다[6].

어플리케이션 공유의 기본 과정은 호스트에 있는 어플리케이션의 출력에 대한 정보를 다른 호스트로 전송함으로써 해당 어플리케이션을 공유할 수 있게 한다. 그리고 다른 호스트에 있는 참여자의 입력을 어플리케이션의 입력으로 전달하여 마치 어플리케이션이 수행되고 있는 호스트에서 입력이 발생한 것 같이 해주는 기능을 가지고 있다. 특히 어플리케이션 공유 방식 중에 어플리케이션의 윈도우를 이미지로 캡처하여 공유하는 윈도우 공유 방식에 대한 연구가 활발히 진행 중이며 대표적인 시스템으로는 [3,4,5,6]이 있다. 표 1은 어플리케이션 공유 시스템들간의 기능을 비교한 표이다.

표 1에서 참고문헌[3,4,5,6]의 어플리케이션 공유 시스템은 가

상 환경에서 이루어지는 3 차원 공간 개념을 고려하지 않았기 때문에 가상공간상의 공동작업에 이용되기에는 문제점을 가지며 인터넷상에서 웹 브라우저를 통한 공유가 불가능하다.

표 1. 기존 윈도우 어플리케이션 공유 시스템의 기능 비교

기능 \ 연구	[3]	[4]	[5]	[6]
기존 어플리케이션 공유	○	○	○	○
이질 환경에 대한 확장성	X	X	X	○
동적 참여자에 대한 현재 상태 제공	X	X	X	○
시스템 동기화	○	X	X	○
충돌 해결	○	○	○	○
3 차원 공간의 고려	X	X	X	X
웹 지원	X	X	X	X

### 3. 어플리케이션 공유 엔진 설계

이 장에서는 다중 참여자를 지원하는 VRML 기반 어플리케이션 공유 엔진을 설계한다. 설계되는 어플리케이션 공유 엔진은 웹 브라우저 내의 VRML로 구성되는 3 차원 가상 공간상의 어플리케이션 뷰(View)에 대한 참여자의 이벤트를 입력받는 에이전트가 참여자 사이트로 이동하여 참여자 이벤트를 받아 서버에 전송하여, 공유 되는 어플리케이션에 반영하고 반영된 어플리케이션의 뷰를 참여자 사이트에 전송하여 뷰를 공유하는 방식을 취한다.

#### 3.1 어플리케이션 공유 엔진의 구성

VRML 기반 어플리케이션 공유 엔진의 구성은 그림 1 과 같다.

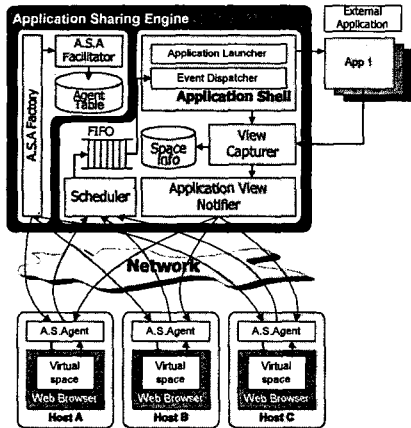


그림 1. 어플리케이션 공유 엔진 구성도

어플리케이션 셸(Application Shell)은 참여자 그룹으로부터 들어 오는 어플리케이션 구동의 요구를 받아 서버에 존재하는 어플리케이션을 구동시키고 공동작업동안 발생하는 참여자의 이벤트를 실제 서버 사이트에서 발생한 것처럼 어플리케이션에 반영시키는 역할을 담당한다. 뷰 캡처기(View Capturer)는 참여자의 이벤트가 반영된 어플리케이션의 뷰를 캡처하는 기능과 가상 공간 정보를 갱신하는 기능을 담당한다. 어플리케이션 뷰 전달자(Application View Notifier)는 캡처된 어플리케이션의 뷰를 각각의 참여자 사이트로 전송하는 역할을 담당한다. 스케줄러(Scheduler)는 참여자 사이트로부터 전달되는 이벤트 메시지를 해석하여 실제 윈도우에서 발생한 이벤트 형식으로 변환한 뒤 큐에 메시지를 삽입한다. 어플리케이션 공유 에이전트(A.S.A: Application Sharing Agent)는 참여자 사이트

로 이동하여 참여자의 웹 브라우저내의 가상 환경상에서 발생하는 이벤트를 캡처하여 서버로 전송하는 역할과 서버에서 전송되어 오는 이미지를 클라이언트에 반영하는 역할을 담당한다.

### 3.2 어플리케이션 셸

어플리케이션 셸은 참여자 그룹에서 사용할 어플리케이션의 실행과 종료기능인 생명주기 기능을 담당하며 스케줄러를 통해 전달되는 이벤트를 어플리케이션에 반영하는 기능을 갖는다. 또한 어플리케이션 셸은 공유 어플리케이션 정보인 어플리케이션의 핸들과, 윈도우 영역의 좌표 등을 관리한다. 그리고 어플리케이션 셸은 서버 측에서 사용할 수 있는 어플리케이션을 리스트로 관리하며 각각의 참여자로부터 전달되는 이벤트를 적절한 어플리케이션에 반영할 수 있도록 한다.

### 3.3 뷰 캡처기

뷰 캡처기는 참여자의 이벤트에 의해 변경되는 어플리케이션의 뷰를 캡처하는 역할을 담당한다. 어플리케이션 셸은 참여자의 이벤트가 반영된 이후나 주기적으로 뷰 캡처기에 어플리케이션 윈도우를 캡처하도록 메시지를 전달한다. 뷰 캡처기는 윈도우의 뷰를 DIB(Device Independent Bitmap)형식의 비트맵 이미지로 캡처한 뒤 JPEG 형식의 이미지로 저장한다.

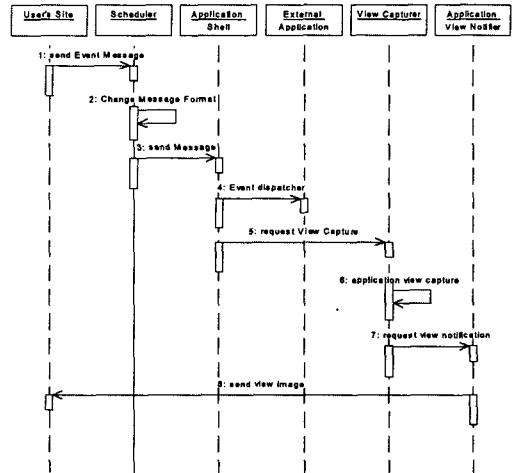


그림 2. 어플리케이션 공유 엔진에서 뷰 공유 동작 과정

### 3.4 어플리케이션 뷰 전달자

어플리케이션 뷰 전달자는 가상공간에 참여한 참여자들에게 동일한 뷰를 제공하기 위한 기능을 갖는다. 뷰 캡처기에서 주기적을 생성되거나 이벤트가 반영된 뒤에 이미지 파일을 참여자 사이트로 전송하여 참여자에게 WYSIWIS(What You See Is What I See)를 제공한다. 그림 2 는 참여자의 이벤트가 반영되고 어플리케이션 뷰가 참여자 사이트로 전송되는 과정을 나타내고 있다.

### 3.5 스케줄러

스케줄러는 참여자 사이트로부터 들어오는 메시지를 큐에 저장한 뒤 메시지를 받아 실제 어플리케이션에서 발생한 이벤트와 같은 메시지 형식으로 변환하는 역할을 담당한다. 또한 스케줄러는 참여자 메시지 버퍼를 두어 동일한 참여자의 이벤트가 다른 참여자의 이벤트에 의해 끊임없이 순서적으로 처리되도록 한다. 이는 버퍼에 들어온 메시지의 타입과 참여자 아이디를 검사하여 이벤트

의 쌓이 순서 있게 큐에 삽입되도록 하는 방식을 취한다. 알고리즘 1은 이벤트 쌓을 만들어 이벤트가 순차적으로 처리되게 하는 알고리즘이다.

알고리즘 1. 이벤트 스케줄링 알고리즘

```

Input : Buffer_Data
Output : FIFO input
Begin Method
While( True )
If(Read Buffer_data is NULL) continue();
else{
switch(Eventtype){
case MouseEvent:
Retrive EventPair with same Client
And Event_type from Buffer;
insert EventPair with ApplicationID to FIFO;
case KeyboardEvent:
Retrive EventPair with same Client
And Event_type from Buffer;
insert EventPair with ApplicationID to FIFO;
}
}
EndIf
EndWhile
End Method
    
```

### 3.6 어플리케이션 공유 에이전트

어플리케이션 공유 에이전트는 참여자 사이트로 이동하여 웹 브라우저 내에 가상 공간상에 표현된 다수의 어플리케이션에 대한 이벤트를 서버로 전송하는 역할과 서버로부터 전송되어오는 이미지를 가상환경상의 반영시키는 기능을 갖는다. 어플리케이션 공유 에이전트는 참여자 사이트로 이동할 때 어플리케이션 공유 에이전트 퍼실리테이터(Facilitator)에 프록시를 등록한 뒤 이동하여 어플리케이션 공유 시에 발생할 수 있는 참여자의 이벤트는 마우스와 키보드 이벤트를 처리한다.

마우스 이벤트 메시지는 마우스의 클릭이나 이동에 대한 마우스 이벤트 종류와 마우스 좌표값으로 구성된다. 이중 마우스 좌표값은 3차원상에 존재하는 어플리케이션 뷰에 대한 마우스 좌표값이기 때문에 참여자의 뷰 포인트와 어플리케이션의 3차원상의 위치 그리고 마우스의 위치에 따른 어플리케이션 뷰상의 상대 좌표값으로의 변환이 필요하다. 또한 키보드 이벤트 메시지는 키보드 이벤트의 종류와 해당 키보드의 아스키 값으로 구성되며 마우스 이벤트와 달리 이벤트가 발생할 때 마다 전송되는 것이 아니라 엔터 값과 일정시간을 동안 버퍼에 저장하였다가 값을 전송하는 메카니즘을 갖는다.

또한 어플리케이션 공유 에이전트는 해당 어플리케이션의 3차원상의 위치와 참여자의 뷰 포인트간의 관계를 검사하여 일정 수준의 각을 넘어서 참여자의 어플리케이션에 대한 이벤트를 서버로 전송하지 않아 이벤트를 어플리케이션에 반영되지 않도록 한다.

### 4. 실험 및 평가

이 연구에서 설계한 다중참여자를 지원하는 VRML 기반 어플리케이션 공유 엔진을 위해 서버는 MS-Windows NT 운영체제하에 Visual C++ 6.0 과 IIS 를 사용하였으며 클라이언트는 웹브라우저 내에 Cosmo Player 2.11 을 addon 하였다. 그리고 IBM 사의 Aglet 을 사용하여 구현되었다. 그리고 실험을 위한 네트워크 환경의 서버측은 ATM 155MB 와 클라이언트는 100base-TX Ethernet LAN 환경상에서 이루어졌다.

그림 3 은 서버 측에 공유하기 위해 그림판과 메모장이 구동되었을 때 클라이언트측에서 발생한 이벤트가 실제 어플리케이션에

반영되고 각각의 클라이언트의 윈도우 어플리케이션 뷰가 동일하게 변하는 것을 보여주는 그림이다.

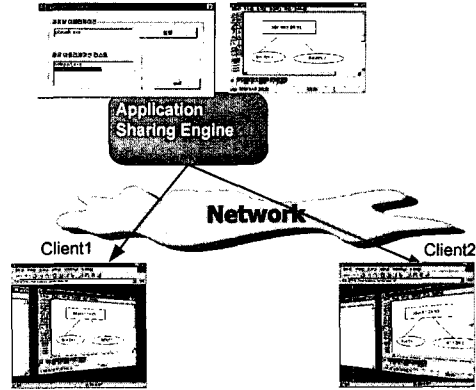


그림 3. 어플리케이션 공유 엔진을 통한 뷰의 공유

[표 1]에서 언급한 기존 어플리케이션 공유 시스템에서 지원하는 기능에 대해 이 연구에서 설계한 VRML 기반 어플리케이션 공유 엔진은 다음과 같이 비교될 수 있다. 먼저 기존의 어플리케이션의 수정없이 가상공간 내에서 어플리케이션의 재사용을 할 수 있으며, 웹브라우저 상에서 어플리케이션 공유가 가능하기 때문에 운영체제에 의존적인 방식이 아니다. 또한 어플리케이션이 윈도우 어플리케이션 뷰로 공유 되기 때문에 동적 참여자에 대한 현재 상태의 제공이 가능하다. 그리고 스케줄러를 통한 이벤트의 순서화 및 참여자의 충돌을 해결할 수 있었다. 또한 VRML 을 통한 가상공간과 Cosmo player 가 addon 된 브라우저에서 어플리케이션 공유를 할 수 있었다.

### 5. 결론

이 논문에서는 다중참여자를 지원하는 VRML 기반 어플리케이션 공유 엔진을 구현하였다.

설계된 가상환경상의 어플리케이션 공유 엔진은 참여자 사이트로 에이전트를 이동시켜 웹 브라우저 내의 3차원 가상 공간에서 참여자의 어플리케이션에 대한 3차원상의 이벤트를 2차원적인 상대값으로 바꾸고 실제 어플리케이션상의 이벤트로 변환하도록 하여 3차원 공간에서 어플리케이션을 공유를 통한 공동작업을 가능하도록 하였다. 또한 참여자의 이벤트를 순서적이고 충돌해결을 위한 스케줄러를 도입하여 해결하였으며 가상환경이 VRML 로 구성되기 때문에 운영체제에 독립적이고 웹상에서 어플리케이션 공유가 가능하다. 그리고 가상환경을 위한 CSCW 응용 개발에 대한 오버헤드를 줄이고 기존의 윈도우 어플리케이션을 가상공간에 사용함으로써 기존 어플리케이션의 재사용성을 높일 수 있었다.

### 참고 문헌

- [1] A. Versey and Ajay Paul Sravana, CASE as Collaborative Support Technologies, *Communication of the ACM*, Jan. 1995, pp.83-94
- [2] Schooler, E. M., Conferencing and Collaborative Computing, *ACM Multimedia Systems*, Vol. 4, No.5, pp. 210-225(1996)
- [3] D. Garfinkel, B. B. Welti and T. W. Yip, "HP SharedX: A Tool for Real-Time Collaboration," *Hewlett-Packard Journal*, April, 1994.
- [4] H. Abdel-Wahab and M. Feit, "XTV : A Framework for Sharing X Window Clients in Remote Synchronous Collaboration," *Proceedings IEEE Tricomm'91*, August, 1991.
- [5] M. Altenhofen, B. Neidecker-Lutz and P.Tallett, "Upgrading a Window System for Tutoring Fuctions," *Proceedings of International workshop on Distributed Window System*, December, 1991.
- [6] 김상욱 외 4명, " T.SHARE 기반 공동작업 분산제어엔진의 개발," *정보과학회논문지(C) 제 5권 제 1호*, 1999.2