

다중 참여자를 지원하는 3차원 가상환경에서 공동작업을 위한 어플리케이션 공유 시스템 설계

탁진현[†] · 이세훈^{††} · 왕창종^{†††}

요약

공동작업을 위한 어플리케이션 공유는 분산되어 있는 개발도구와 어플리케이션을 공유하여 다수의 참여자가 환경적으로 제약 받지 않고 공동 개발할 수 있도록 한다. 그러나 CVE(Collaborative Virtual Environment)에 기존의 어플리케이션 공유 방식을 적용하기에는 기존 방식이 3차원이라는 공간적인 요소를 고려하지 않고 자유스런 상호작용을 방해한다는 문제점을 가지게 된다. 이 연구에서는 이러한 문제점들을 해결하여 CVE에서 효과적으로 어플리케이션을 공유 하기 위한 3차원 가상 환경에서 공동작업을 위한 어플리케이션 공유 시스템을 설계하였다. 설계된 시스템은 어플리케이션 공유 관리기와 그룹 관리자와 통신 관리자로 구성하였다. 시스템은 참여자 사이트로 이동하는 에이전트에 의해 가상 환경에서 참여자의 어플리케이션에 대한 3차원상의 이벤트를 감지할 수 있고 스케줄링에 의해 참여자간의 충돌을 해결할 수 있다.

따라서 설계한 가상환경 공동작업을 위한 어플리케이션 공유 시스템은 가상환경을 위한 CSCW용 응용 개발 시, 기존 윈도우 어플리케이션을 3차원 가상환경에서 쉽게 공동 사용할 수 있도록 함으로써 효율적인 공동 작업을 지원하고 기존 어플리케이션 소프트웨어의 재사용성을 높일 수 있다.

Design of Application Sharing System for Collaborative Works in 3D Virtual Environment supporting Multi-Participants

Jin-hyun Tak[†] · Sei-hoon Lee^{††} · Chang-jong Wang^{†††}

ABSTRACT

Application sharing for collaborative works make multiple participants to be capable of collaborative developments through sharing development tools and applications which is distributed to various system without spatial restriction. However, applying legacy application sharing to CSCW based virtual environment has a problem because it dont consider 3D space and it disturbs the harmonic interaction of participants.

In this study, we designed application sharing system for collaborative works in 3D virtual environment to share application effectively on collaborative virtual environment. The designed application sharing system for collaborative works in 3D virtual environment consist of application sharing manager, group manager and communication manager. This system is able to perceive event of 3D space about application of participant on 3D virtual environment by agent which moves to participant sites and get an ordered events and events solved conflict among participants through scheduling.

Therefore, designed application sharing system for collaborative work in 3D virtual environment can support efficient collaborative works and improve reusability of legacy application through using easily legacy application in 3D virtual environment when CSCW application development for virtual environments.

† 정 회 원 : 인하대학교 대학원 전자계산공학과
†† 정 회 원 : 인하공업대학 전자계산학과 교수
††† 정 회 원 : 인하대학교 전자계산공학과 교수
논문접수 : 1999년 11월 1일, 심사완료 : 2000년 1월 13일

1. 서 론

컴퓨터 성능의 급속한 발전과 고속 네트워크 기술의 발달로 인한 오늘날의 컴퓨팅 환경이 개방화됨에 따라 CSCW(Computer Supported Collaborative Works) 응용은 화상 회의 시스템, 상호 협력적 원격 교육, 원격 진료, 그룹 게임 분야 등으로 실생활의 거의 모든 분야로 확대되고 있다[1, 2, 3]. 특히, 컴퓨팅 능력의 발달로 인한 3차원 가상 환경 개념을 바탕으로 둔 CSCW인 CVE(Collaborative Virtual Environment)는 기존의 CSCW에서 보다 원격지의 참여자들끼리의 상호작용을 보다 현실감 있도록 지원할 수 있다는 측면에서 각광받고 있다[4, 5, 6]. 이러한 3차원 가상 환경에서 공동작업을 위해서는 참여자들 간의 자유로운 대화가 가능하여야 하고, 공동 작업을 위한 도구, 즉 공유 어플리케이션들이 제공되어야 한다[4, 6]. 각각의 참여자들이 이전까지 사용해오던 작업 도구들을 3차원 가상환경상에서 공유할 수 있다면 공동작업이 보다 쉬워지고 효율이 높아질 것이다. 하지만 대부분 개인 사용자를 지원하는 수많은 어플리케이션 소프트웨어들을 공동사용자용으로 확장하거나 새로 개발하는 것은 현실적으로 매우 어려운 일이다.

이에 따라 단일 사용자용 어플리케이션을 네트워크 환경에서 공유하는 연구가 활발히 진행 중에 있다. 기존 어플리케이션 공유에 관한 연구로는 SharedX[7], XpleXer[8], NetMeeting[9], ProShare[10] 등이 있다. 이러한 기존 연구 중에 3차원상에서의 어플리케이션 공유에 대한 연구는 발표된 바 없으며, 또한 기존 연구들은 3차원 가상환경에서 발생하는 사용자의 이벤트처리를 고려하고 있지않기 때문에 CVE가 그대로 사용하기에는 부적합하다. 또한 기존의 어플리케이션 공유에서 사용한 발언권 제어를 통해 참여자의 이벤트를 처리하는 방식은 자유스런 공동작업을 방해하여 효율적이지 못하다. 따라서 2차원상의 어플리케이션 공유 방식을 3차원상에 도입하기 위해 기존 방식을 3차원 환경에 적합하도록 확장하고 보완해야만 한다.

이 연구에서는 3차원 가상 공간의 이벤트를 처리를 포함한 어플리케이션 공유 방식을 제안하고, 시스템을 설계한다.

설계되는 가상환경 어플리케이션 공유 시스템은 다수의 참여자들이 가상 환경상에서 기존 윈도우 어플리케이션을 변경 없이 어플리케이션의 뷰를 통해 공유하

며 작업할 수 있는 환경을 제공한다. 또한 가상공간 브라우저내에 구성된 3차원 환경에서 발생하는 참여자의 어플리케이션에 대한 이벤트를 감지하는 에이전트를 클라이언트 사이트로 이동시킴으로써 3차원상에서 발생하는 이벤트를 실제 어플리케이션에서 발생하는 이벤트로 변경하여 가상 환경 공동작업을 위한 어플리케이션 공유를 가능하도록 한다. 또한 발언권 제어방식이 아닌 다수의 참여자가 자율적인 행위에 의해 발생시켜 전달된 이벤트를 순서 있게 스케줄링하는 기능을 제공하여 원활한 어플리케이션 공유가 가능하게 한다.

2. 기존의 어플리케이션 공유 시스템

2.1 어플리케이션 공유 방식의 분류

기존 어플리케이션 공유 방식은 공유 대상과 구현 방식에 따라 분류할 수 있다[11, 16]. 공유 대상에 의한 분류는 화면공유와 윈도우 공유 방식으로 분류하며 이는 참여자의 프라이버시 보호를 위해 고려되는 사항이다[12].

또한 구현방식에 의한 분류는 그래픽 프리미티브를 가로채는 방법과 화면을 캡처하여 공유 하는 방식으로 분류된다. 그래픽 프리미티브를 가로채는 방식은 X protocol 자체를 고쳐서 새로운 방법으로 어플리케이션을 다시 작성해야 하는 문제점을 가지며 윈도우 환경에서는 GDI(Graphic Device Interface) 메시지를 가로채 후 연결된 호스트에 전달하여 GDI 메시지를 수행함으로써 어플리케이션을 공유한다[8, 11, 13]. 그러나 이 방식은 어플리케이션의 유지보수 및 재사용성에 문제점을 가진다.

화면 캡처 방식은 화면에 나타난 어플리케이션의 영상을 캡처하여 전송하는 방식으로 이 방식은 주로 MS-Windows 환경에서 많이 사용하는 방식이다. 이 방식은 캡처한 이미지를 이용하여 어플리케이션을 공유하기 때문에 전송량이 많아 반응 시간이 느린 단점을 가지지만 구현이 용이하다는 점과 MS-Window의 버전에 영향을 받지 않고 어플리케이션을 공유할 수 있다는 장점을 가진다[9, 10].

2.2 기존의 어플리케이션 공유 시스템

기존의 어플리케이션 공유 시스템으로는 SharedX[7], XpleXer[8], ProShare[10], CollaborationFramework[14] 등이 있다. SharedX, XpleXer는 X Window 상에서 어

플리케이션 공유를 지원하는 시스템들이고 Collaboration Framework, NetMeeting[9]과 ProShare는 MS-windows 환경에서 어플리케이션의 공유를 지원하는 시스템이다. 현재 enduser들의 컴퓨팅환경은 MS-windows 환경이 주종을 이루기 때문에 MS-windows 환경에서 어플리케이션을 공유를 위한 연구가 활발하다[16]. 이 절에서는 MS-windows 환경에서 어플리케이션을 공유하는 대표적인 두 시스템에 대해 고찰하고 문제점을 살펴본다.

2.2.1 CollaborationFramework 시스템

CollaborationFramework는 IBM Tokyo연구소에서 개발한 공동작업을 위한 어플리케이션 공유 툴킷이다 [14]. 공동작업을 위한 어플리케이션은 각각의 참여자 사이트에 복제되어 관리되며 이를 제어하기 위해 각각의 컴퓨터 호스트에서 동작되어 어플리케이션 제어와 공동작업세션을 제공하는 노드 컨트롤러(Node Controller)와 단일사용자용 어플리케이션에서 사용자 연산을 모니터링하는 컨트롤 객체(Control Object)를 제공한다. 이러한 기능들은 응용들에게 API형태로 제공되며, 응용들에서는 이 API를 이용해서 동시에 다중의 참여자 사이에 공유되는 어플리케이션을 제어하고 관리한다.

2.2.2 어플리케이션 뷰 공유 시스템

어플리케이션 뷰 공유 시스템은 어플리케이션 공유를 위한 시스템으로 공유 어플리케이션의 영상을 캡처하여 참여자간의 어플리케이션을 공유한다[17]. 이 시스템은 어플리케이션의 뷰 인터페이스의 주기적인 캡처와 전송으로 인한 네트워크의 부하를 줄이기 위해 어플리케이션 뷰의 셀단위 비교를 수행한다. 셀 비교 후 변환된 부분의 셀만을 전송함으로써 뷰의 데이터를 줄이고 사용자에게 빠른 디스플레이 피드스루(feedthrough)를 제공한다.

그러나 CollaborationFrameWork는 복제모델이기 때문에 모든 참여자 사이트에 존재하는 어플리케이션만을 공유할 수 밖에 없다[14]. 또한 어플리케이션 뷰 공유 시스템은 뷰의 순차적 셀단위 비교의 문제점과 동형 시스템으로 개발되었기 때문에 공동작업 참여자가 불필요한 서버의 기능 까지 가지게 된다. 그리고 CollaborationFrameWork와 어플리케이션 뷰 공유 시스템은 3차원 가상환경에서의 이벤트처리 기능이 없고 발언권 제어로 인해서 가상환경상의 자율적인 사용자 행위를

방해하는 요인으로 작용한다.

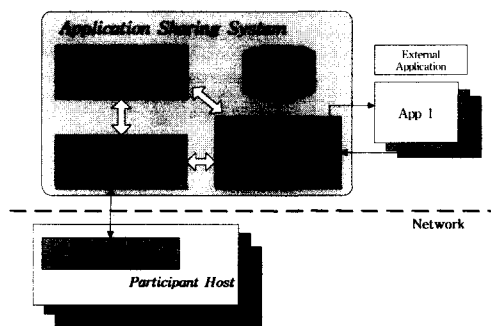
3. 어플리케이션 공유 시스템 설계

이 장에서는 3차원 가상 환경에서 공동작업을 위한 어플리케이션 공유 시스템을 설계한다.

3.1 시스템 설계 개요

이 연구에서는 어플리케이션 공유하기 위한 윈도우 공유 방식과 영상 캡처 방식을 도입하여 가상환경 공동작업을 위한 어플리케이션 공유가 가능하도록 한다. 가상환경상의 다수의 참여자가 발생하는 이벤트를 처리하기 위해 스케줄러를 도입하여 이벤트를 순서 있게 스케줄링하고 뷰 공유를 통한 전송데이터 검출에 대한 성능향상을 위해 메모리상의 뷰 비교방식과 셀단위 변화뷰를 압축하여 전송하는 방식을 채택한다.

(그림 3-1)은 이 연구에서 설계하는 3차원 가상환경상의 공동작업을 위한 어플리케이션 공유 시스템의 구성이다. 설계하는 어플리케이션 공유 시스템은 서버에서 공유되는 어플리케이션 관리와 어플리케이션의 뷰를 관리하는 어플리케이션 공유 관리자(Sharing Application Manager)와 공동작업에 참여하는 사용자를 관리하는 그룹 관리자(Group Manager), 참여자의 통신/이벤트 메시지의 스케줄링과 어플리케이션 뷰를 전송을 담당하는 통신 관리자(Communication Manager)와 참여자 호스트로 이동되는 노드 에이전트(Node Agent)로 구성된다.



(그림 3-1) 시스템의 구성도

3.2 어플리케이션 공유 관리자

어플리케이션 공유 관리자는 공동작업에서 사용할 어플리케이션의 생명주기를 담당하며 참여자로부터 전

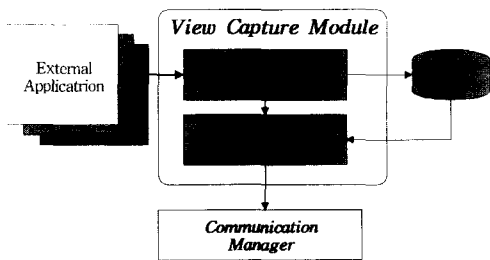
달되는 어플리케이션에 대한 이벤트 반영과 변화되는 어플리케이션의 뷰를 캡처하는 역할을 담당한다.

공동작업에서 사용할 어플리케이션은 공동작업에서의 그룹 의장이 공유 엔진에서 제공하는 어플리케이션의 리스트에서 선택되며, 어플리케이션 공유 관리자는 등록된 어플리케이션을 구동함으로써 어플리케이션의 공유를 시작하게 된다. 또한 어플리케이션 공유 관리자는 어플리케이션 공유 시에 필요한 어플리케이션에 대한 정보를 관리한다.

참여자로부터 전달되는 이벤트에 대한 반영은 통신 관리자로부터 먼저 전달되어오는 메시지에서 이벤트가 적용될 어플리케이션을 찾아 해당하는 어플리케이션의 이벤트를 실제 어플리케이션에서 발생한 이벤트처럼 적용한다. 또한 이벤트 반영은 참여자별로 이벤트 타임을 관리함으로써 동일 어플리케이션 환경에서 작업할 수 있는 작업환경을 제공한다. 그리고 이벤트 반영 시 통신 관리자를 통해 어플리케이션을 사용하는 참여자 아이디를 전송하여 참여자의 작업 시에 다른 참여자의 작업중인 것을 인식할 수 있도록 정보를 제공한다.

3.2.1 어플리케이션 뷰 관리 모듈

어플리케이션에 대한 뷰 관리는 참여자의 이벤트 반영으로 인해 변경된 어플리케이션에 대한 뷰 캡처와 뷰에 대한 변경된 부분을 찾는 역할을 담당한다. (그림 3-2)는 뷰 관리 모듈의 내부 구성도이다.



(그림 3-2) 뷰 관리 모듈의 내부 구성도

일정주기 간격으로 어플리케이션의 뷰를 캡처하여 모든 어플리케이션의 뷰를 참여자에게 전송한다는 것은 시스템의 상당한 오버헤드(overhead)이다. 또한 공동작업에 참여하는 참여자수의 증가도 어플리케이션 공유 엔진에서 전송해야 할 데이터의 양이 증가하는 원인이 된다. 이 연구에서는 변경된 뷰의 데이터량의

감소뿐만 아니라 검색시간 감소를 위한 메커니즘을 사용한다. 먼저 전송하는 데이터의 양을 줄이기 위해 어플리케이션에 대한 뷰를 2차원의 격자로 구분하여 관리한다. 이는 격자로 어플리케이션의 뷰를 분할함으로써 스케줄러가 이벤트 충돌을 검사하는 기준을 제공하며, 격자로 구성되는 가상환경의 어플리케이션 뷰와 어플리케이션의 공유 시 공유 시스템과 참여자간의 동기화의 유연성을 제공한다.

그리고 변화뷰의 검색시간을 감소하기 위해 격자로 관리되는 어플리케이션 뷰는 이전에 캡처한 어플리케이션 뷰와의 변화를 검사하기 위해 이전의 뷰의 장치 독립비트맵(DIB : Device Independent Bitmap)과 현재 뷰의 DIB를 메모리상에서 Exclusive-OR 연산을 수행하여 변화 부분을 추출한다. 메모리상의 Exclusive-OR 연산은 공유엔진에서의 이미지 비교수행에 대한 오버헤드를 줄인다. 변화부분 검출이 이루어진 후에 이미지는 다시 격자형의 2차원 배열에서의 변화부분만을 JPEG 이미지로 변환한다. <알고리즘 3-1>은 어플리케이션 뷰의 변화를 탐지하기위한 알고리즘이다.

<알고리즘 3-1> 뷰 변화 탐지 알고리즘

메소드 이름 : 뷰 변화 탐지
 실인자 : 어플리케이션 아이디
 결과 : 어플리케이션 뷰 변화 부분을 통신 관리자에게 제공

Begin Method

```

changeCellofView := new
ChangeCellofView[Width][Height];
previousView := currentView;
currentView := captureView (ApplicationID);
changedView := exclusive_OR(previousView,
currentView );
If isSameViewPrevious() is False Then
    For(x=0; x < cellWidth; x ++)
        For(y=0; y < cellHeight; y ++)
            If isChangedCell(x,y) is TRUE Then
                send_to_CM (x, y, celltoJPEG(x,y));
            End If
        End For
    End For
End For
End If
End Method
    
```

캡처된 어플리케이션 전체 이미지는 동적인 사용자의 참여를 위해 가상환경 저장소에 저장된다.

3.3 그룹 관리자

이 절에서는 가상환경 어플리케이션 공유 시스템에서 사용자 관리를 담당하는 그룹 관리자를 설계한다. <표 3-1>은 이를 위한 기능들을 정의하고 있다.

<표 3-1> 시스템에서의 그룹 관리 기능

연산의 종류	연산의 기능
Basic Operations	Create Group
	Destroy Group
Group Management Operations	Join Group
	Leave Group
	Share Applications
	Destroy Application
Information Management Operation	Manage Group Context
	Manage and Retrieve User/Group Information

그룹 관리자는 참여자의 요구에 따라 해당되는 그룹을 생성하고, 생성된 그룹들을 그룹 풀에서 관리한다. 그룹 의장에 어플리케이션 공유에 대한 이벤트는 공유 어플리케이션 관리자에게 전달되어 어플리케이션 공유 관리자가 어플리케이션을 구동할 수 있도록 한다.

3.3.1 그룹 관리

그룹 관리의 참여자들로 구성되는 그룹의 문맥(context)을 관리하고, 참여자들 간의 상호 작용에 필요한 기능들을 수행한다. 그룹에 대한 정보는 그룹 문맥을 이용하여 유지된다. 그룹 문맥에는 현재 그룹에 참가하고 있는 참여자들의 정보와 현재 그룹의장의 정보, 공유 어플리케이션에 대한 정보 등을 갖는다. 그룹 관리자는 최근에 처리한 사건에 해당하는 그룹 문맥을 유지하는데, 만약 새로운 참여자의 참여나 참여자 탈퇴, 어플리케이션의 이벤트 반응자와 같은 경우 해당하는 문맥으로 문맥교환이 이루어진다. 문맥교환이 발생되면 통신 관리자를 이용해서 해당 사실이 참여자들에게 통보된다.

그룹의 의장은 그룹 참여자들의 역할을 부여할 수 있는 권한 외에 그룹의 강제 종료와 그룹에 대한 정보를 변경할 수 있는 권한을 가지며, 그룹내의 어플리케이션을 제어할 수 있는 권한을 가진다. <알고리즘 3-2>는 그룹 관리를 위한 사용자 요구 처리 알고리즘이다.

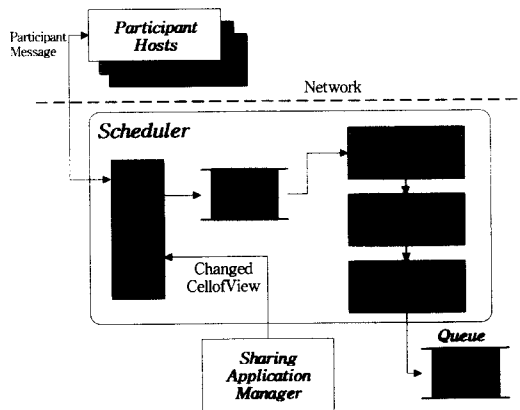
<알고리즘 3-2> 사용자 요구 처리 알고리즘

메소드 이름 : 사용자 요구 처리
 실인자 : 사용자 요구
 결과값 : 요구한 해당된 결과

```

UserRequest := getUserRequest();
Begin Method
Case UserRequest
GroupJoin :
    User Role is determined by GroupChairman;
    Update GroupContext->List;
    Send signal to CommunicationManager;
GroupLeave :
    Update GroupContext->List;
    Send signal to CommunicationManager;
ShareApplication :
    If Requestor is GroupChairman then
        Update GroupContext->hasToken;
        Send signal to Sharing Application Manager;
    EndIf
DestroyApplication :
    If Requestor is GroupChairman then
        Update GroupContext->hasToken;
        Send signal to Sharing Application Manager;
    EndIf
GetChairmanInfo, ApplicationInfo, GetGroupmemberInfo;
    Send informations with Requestor to
    CommunicationManager;
SendMessageToAll :
    Send Message and Target Userlist to
    CommunicationManager;
End Case;
End Method;
    
```

3.4 통신 관리자



(그림 3-3) 통신관리자의 내부 구성도

통신 관리자는 공동작업에서 발생하는 참여자 간의 통신 채널을 제공하고 참여자 호스트로부터 전달되는 이벤트를 스케줄링하여 공유 어플리케이션 관리자에게 제공하는 역할을 담당한다. 또한 공유 어플리케이션 관리자로부터 제공되는 셀단위 변화부와 메시지를 그룹에 참여한 참여자 사이트로 브로드캐스팅하는 역할을 담당한다. (그림 3-3)은 통신관리자의 구성도 이다.

3.4.1 이벤트 스케줄링

이 연구에서는 각각의 참여자들 사이에서 발생하여 공유 시스템으로 전송되어오는 이벤트를 적절히 순서화 있게 스케줄링하여 실제 어플리케이션에 반영하도록 함으로써 발원권 제어 방식보다 3차원 환경에서 더욱 유연한 작업환경을 제공하도록 한다.

<알고리즘 3-3>은 적절한 이벤트 순서를 얻기 위한 스케줄링 알고리즘이다.

<알고리즘 3-3> 이벤트 순서화 스케줄링

메소드 이름 : 이벤트 순서화 스케줄링
 실인자 : 스케줄러 내부 버퍼
 결과값 : 순서화된 어플리케이션 이벤트 메시지

```

Begin Method
While True Do
    If buffer == NULL Then continue();
    initialize CollisionWindowCell();
    buffer_data := Read(buffer);
    Case data_type
        WM_LBUTTDOWN :
            insert changeEvent(buffer_data) to queue;
            get UserID and ApplicationID;
            check collisionWindowCell;
            insert changeEvent(buffer_data) with
                same userID and ApplicationID to
                queue
            until buffer_data is
WM_LBUTTONUP;
        WM_RBUTTONDOWN :
            insert changeEvent(buffer_data) to queue;
            Get UserID and ApplicationID;
            Check collisionWindowCell;
            insert changeEvent(Buffer_data) with
                same userID and ApplicationID to
                queue
            until buffer_data is
            WM_RBUTTONUP;
        WM_LBUTTONDOWNBLCLK :
            insert changeEvent(buffer_data) to queue;
    
```

```

KEYDATA :
insert changeEvent(buffer_data) to queue;
End Case
EndWhile
End Method
    
```

순서화 스케줄링은 먼저 CollisionWindowCell을 초기화 한다. 이는 이벤트 충돌 검사에서 사용할 배열로 이벤트가 반영되는 어플리케이션의 격자를 나타낸다. 버퍼로부터 데이터를 가져와 데이터 타입에 따라 데이터 처리를 달리한다. 특히, 마우스 이벤트 메시지일 경우 같은 참여자ID와 어플리케이션ID를 가진 이벤트 패턴 메시지가 모두 다른 이벤트는 버퍼에 그대로 남겨지게 된다. 이는 참여자의 이벤트가 반영되는 도중에 다른 참여자에 의해 변경되는 것을 막아준다.

또한 가상 환경에서 공유 어플리케이션 사용 시에 각 참여자의 행위에 대한 이벤트가 다른 사용자에게 의해 방해 받지 않도록 하기 위해 이벤트 충돌 검사를 한다. 이벤트 충돌은 이벤트 순서화 스케줄링에 의해 작성되는 CollisionWindowCell의 격자 좌표값과, 현재 큐에 전달되어온 메시지 데이터의 격자 좌표에서의 위치값을 비교함으로써 충돌여부를 판단한다. 충돌이 발생하면 이전에 입력된 이벤트 패턴이 같은 이벤트를 버퍼에서 제거시키고 CollisionData로 등록시켜 이벤트 패턴이 같은 이벤트 메시지를 버퍼로 전송되는 것을 막는다.

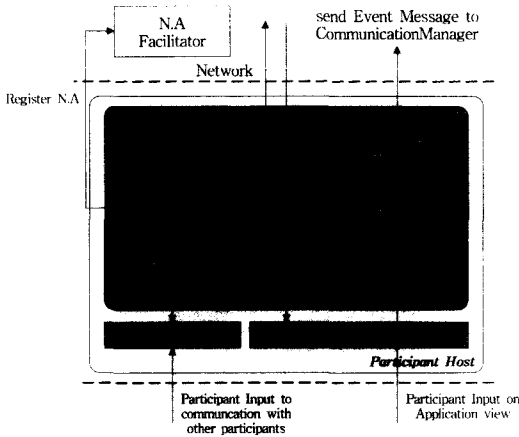
3.4.2 뷰의 동기화

통신 관리자는 일정 주기로 캡처되어 변화된 부분 뷰만을 참여자에게 제공하여 참여자간 뷰 동기화를 담당한다. 또한 동적 참여자에 의한 그룹 관리기의 문맥이 변경될 경우 현재의 가상환경 정보와 현재의 어플리케이션 뷰를 제공함으로써 동적 참여자에게 WYSIWIS (What You See Is What I See)를 제공한다. 또한 새로운 참여자로 인한 가상환경의 정보는 참여자마다 다른 뷰포인트를 가지는 가상환경 정보를 제공하고 이로써 실제 공동작업과 같은 작업환경을 구성한다.

3.5 노드 에이전트

노드 에이전트는 참여자 사이트에서 참여자 그룹간의 통신과 3차원 가상환경상의 다수의 어플리케이션에 대한 이벤트를 서버로 전송하는 역할, 서버로부터 전송되어오는 이미지를 가상환경에 반영하는 기능을 갖

는다. 에이전트의 이동은 공유 시스템에 위치하는 퍼실리테이터(facilitator)에 의해 관리된다. (그림 3-5)는 노드 에이전트의 내부 구조이다.



(그림 3-5) 노드 에이전트의 내부 구조

3.5.1 가상환경에서의 이벤트 감지

노드 에이전트는 참여자가 3차원 가상환경에 위치하는 어플리케이션에 적용하는 이벤트에 대해 감지해야 한다. 이벤트 후킹(hooking)은 윈도우에서 발생하는 모든 이벤트에 대해 관여할 수 있으며, 필요한 경우에는 이벤트의 내용을 바꾸거나 삭제도 가능하게 하는 기술이다. 그러나 이벤트 후킹은 단지 마우스 이벤트와 키보드 이벤트의 발생과 이들의 2차원상의 좌표값을 얻는 후킹만이 가능하다. 즉, 가상환경상의 어플리케이션에서 발생하는 이벤트 좌표에 대한 감지는 할 수가 없다.

또한 가상환경에서 공유 어플리케이션이 3차원상에 위치하고 참여자는 2차원으로 표현되는 가상공간 브라우저 내에 나타나는 공유 어플리케이션 화면에 마우스와 키보드를 통해 작업하기 때문에 2차원상에서 입력되는 참여자의 어플리케이션에 대한 이벤트 좌표에 대한 2차원에서 3차원으로 변환이 필요하게 된다.

2차원에서 3차원 좌표값으로의 매핑은 3차원 환경상에 위치한 어플리케이션의 평면과 2차원 브라우저의 참여자 뷰포인트(viewpoint)와 2차원 좌표값을 이용하여 다음과 같이 계산된다. 먼저 브라우저의 뷰포인트는 이벤트가 행해지는 방향이고 좌표값은 방향의 시작점이 된다. 즉, 3차원상의 직선의 벡터와 직선 위의 점

이 된다. 따라서 3차원상의 어플리케이션 뷰를 평면이라고 가정하고, 직선과 만나게 되는 평면상의 점을 찾게 되면 2차원상에서 발생한 이벤트의 3차원 좌표값을 얻을 수 있다.

따라서 이 연구에서는 가상환경상의 어플리케이션에서 발생하는 이벤트를 감지하기 위해 이벤트 후킹과 센서를 이용한다. 이벤트 후킹은 마우스와 키보드에 대한 이벤트 종류를 검사하고 센서는 3차원상의 이벤트 좌표를 얻는데 사용된다. 가상환경 어플리케이션의 이벤트 메시지는 후킹과 센서에 의해 어플리케이션 내의 3차원 좌표를 구한 뒤 실제 어플리케이션 좌표인 2차원 좌표값으로 변환하여 전송한다. <알고리즘 3-4>는 센서와 이벤트 후킹에 의해 이벤트 메시지를 생성하는 알고리즘이다.

<알고리즘 3-4> 이벤트 메시지 생성

메소드 이름 : 이벤트 메시지 생성
 실인자 : 이벤트의 발생
 결과값 : 이벤트 메시지

```

Begin Method
  hostInfo := GetInfo about ClientID and ApplicationID;
  If Event about Mouse Then
    3DPosition := getPositionValue();
    2Dposition := convert3Dto2D(3DPosition);
    eventType := getEventhookingtype();
    msg := createMessage(hostInfo, 2DPosition,
    eventType);
  End If
  If Event about keyboard Then
    Value := getKeyValue();
    msg := creatMessage(hostInfo, Value, "KEYDATA");
  End IF
  return msg;
End Method
    
```

또한 노드 에이전트는 가상환경에 위치하는 어플리케이션, 참여자의 뷰포인트 그리고 참여자와 어플리케이션과의 거리에 따라 이벤트를 발생시키도록 하는 정책을 가지게 된다. 이는 참여자가 효율적용 어플리케이션을 사용할 수 있도록 해주며 에이전트가 3차원 좌표를 좀더 쉽게 검출할 수 있도록 하는 기준을 제공한다. 이러한 기준은 가상공간 브라우저 상에 나타나는 어플리케이션 뷰와의 거리와 뷰포인트가 허용하지 않는 값을 가지게 되면 노드 에이전트는 해당 이벤트를 발생시키지 않도록 한다.

4. 실험 및 평가

이 장에서는 설계한 시스템을 이용하여 CAD 어플리케이션의 공유를 통한 공동작업의 예를 보임으로써 설계한 시스템이 어플리케이션의 공유와 그룹 관리 기능, 스케줄링 기능들을 제공함을 보인다.

4.1 공동작업 시나리오

참여자는 웹 브라우저를 실행하여 시스템에 접속한다. 로그인 대화상자가 나타나게 되면 참여자 이름과 패스워드를 입력하고, 이러한 로그인 과정을 마치면, 그룹 뷰(group view)가 나타나게 되고 참여자는 자신이 참여해야 할 공동작업 그룹을 선택하여 그룹에 참여한다. 참여자가 그룹에 참여하면 서버는 그룹 관리자를 통해 현재 문맥이 변경되고 통신관리자는 해당 작업 그룹의 가상환경 정보와 CAD 어플리케이션의 현재 뷰를 제공한다.

공유된 어플리케이션을 통한 공동작업은 가상공간 브라우저를 통해 이루어지며 채팅창을 통한 참여자간의 의사소통이 이루어진다. 한 참여자의 어플리케이션에 대한 행위는 서버로 전송되어 스케줄링을 거친 뒤 어플리케이션에 반영되게 되고 현재 이벤트 반영 중이라는 사실을 통신 관리자를 통해 다른 참여자에 알린다. 공동작업을 진행하는 동안 그룹의장은 공동작업의 진행상황에 따라 어플리케이션 사용을 제한 할 수 있으며 특정사용자에게만 이벤트 반응을 허용할 수 있다. 또한 어플리케이션 사용은 발언권 제어 방식이 아닌 서버의 스케줄링을 통해 이루어 지기 때문에 다중 참여자의 작업을 가능하게 한다.

4.2 실험 환경

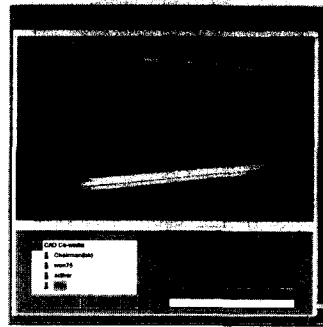
실험을 위한 공유 시스템의 서버는 MS-Windows NT/IIS를 이용하였으며 어플리케이션 공유 시스템은 Visual C++ 6.0를 사용하여 구현 하였다. 또한 3차원 가상환경은 VRML을 기반으로 하였다[15]. 클라이언트는 MS-Windows 98 운영체제하의 웹브라우저 내에서 Cosmo Player 2.11을 addon 하였으며, 클라이언트로 이동되는 어플리케이션 공유 에이전트는 자바 애플릿과 EAI(External Authoring Interface), ActiveX를 이용하여 구현하였다.

4.3 CAD 공동작업

이 실험은 그룹 단위의 어플리케이션 공유를 통한

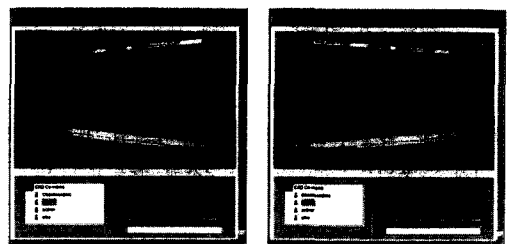
공동작업을 검증하기 위한 것으로 그룹 의장 1명과 참여자 3명으로 구성된 공동작업 그룹을 구성하였으며, 가상환경상의 어플리케이션 공유를 검증을 위해 VRML 가상환경을 구성하여 사용했다. 공동작업에서 사용할 어플리케이션으로 CAD 어플리케이션을 공유하였으며 참여자는 특정한 그룹에 참가하여 CAD 어플리케이션 공유 하여 작업하게 된다.

참여자는 해당 공동작업 그룹을 선택하여 뷰 상에서 마우스 더블 클릭을 하여 해당 그룹의 구성원으로 공동작업에 참여하게 된다. (그림 4-1)은 참여자가 그룹에 참여한 후의 참여자 브라우저 화면이다.



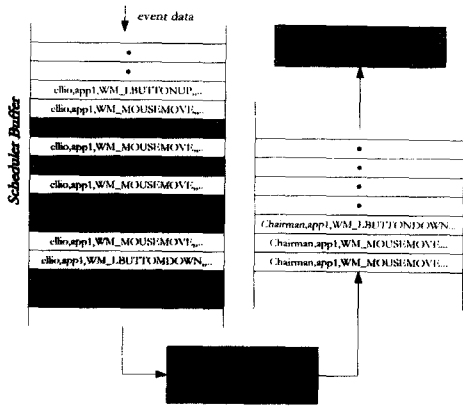
(그림 4-1) 그룹 참여 후의 참여자의 브라우저

(그림 4-2)는 공유된 CAD 어플리케이션 화면에 참여자 1이 이벤트를 발생시키는 동안의 참여자간의 뷰가 일치됨을 보여준다. 참여자 1에 의해 이벤트가 발생되면 서버로 이벤트 메시지가 전송된 뒤 스케줄링을 거쳐 이벤트가 반영되고 사용중인 참여자 아이디와 정보는 참여자 리스트와 채팅창에 정보가 표현되며 어플리케이션의 변화 뷰를 통해 다른 참여자들은 참여자 1이 어플리케이션에 이벤트를 반영하고 있음을 알 수 있다.

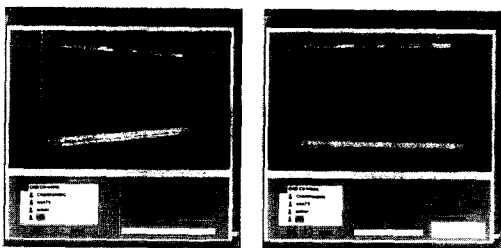


(그림 4-2) 참여자간의 뷰 일치

참여자들간의 동시작업은 그룹의장이 작업 도중에 참여자 3이 다른 영역에 대해 이벤트를 발생시키면 통신관리자에서 충돌 체크와 이벤트 스케줄링을 통해 이벤트가 반영된다. (그림 4-3)과 (그림 4-4)는 스케줄링 알고리즘에서 사용되는 내부 자료 구조와 이벤트 반영후의 참여자3과 그룹의장의 뷰 일치함을 보여준다.



(그림 4-3) 스케줄링 내부 자료구조



(그림 4-4) 동시작업 후의 참여자간의 뷰 일치

실험을 통하여 설계한 3차원 가상환경에서 공동작업을 위한 어플리케이션 공유 시스템이 어플리케이션 뷰를 공유함으로써 가상환경에서의 공동작업이 가능함을 보였고 참여자가 이벤트를 반영할 때 반영중인 사용자의 정보를 제공함으로써 현재 작업중인 참여자의 정보를 참여자들 간에 공유할 수 있음을 보였다. 또한 이벤트 스케줄링을 통하여 발언권제어 방식을 사용하지 않고도 어플리케이션 공유가 이루어짐과 동시작업이 가능함을 보였다. 이와 같이 3차원 가상환경에서 공동작업을 위한 어플리케이션 공유 시스템은 실제 공동작업 공간과 같은 3차원 가상환경에서 공동작업이 이루어

어 지기 때문에 공동작업 중에 실제 참여자들 간에 face-to-face 미팅을 통한 작업이 가능하며, 자신을 대신하는 아바타(avatar)를 공동작업 공간에 참여 시킴으로써 보다 풍부한 인식정보를 제공받을 수 있을 것이다.

5. 결 론

이 연구에서는 3차원 가상환경에서 공동작업을 위한 어플리케이션 공유 시스템을 설계하였다. 설계한 시스템은 어플리케이션 공유를 통해 3차원 가상환경에서 공동작업을 제공할 수 있으며 발언권 제어방식이 아닌 스케줄링을 통한 참여자의 이벤트를 처리함으로써 효과적인 어플리케이션 공유를 제공할 수 있다. 또한 어플리케이션 뷰에 대한 메모리상의 뷰 비교와 쉘단위 전송, 그리고 변화부의 압축을 통해 어플리케이션 뷰 변화에 대한 검출 오버헤드를 줄이고 전송 데이터 양의 감소를 이룰 수 있었다. 따라서 설계한 가상환경 공동작업을 위한 어플리케이션 공유 시스템은 가상환경을 위한 CSCW응용 개발에 대한 오버헤드를 줄이고 기존 윈도우 어플리케이션을 가상환경에 사용함으로써 기존 어플리케이션의 재사용성을 높일 수 있었다. 또한 CVE시스템 개발자들은 설계한 어플리케이션 공유 시스템의 모듈을 사용하여 간단하게 기존 어플리케이션 공유를 통한 공동작업 시스템을 설계할 수 있다.

향후 연구 방향은 기존 어플리케이션 공유 방식과의 연동과 몰입형 가상환경에 사용할 수 있도록 시스템의 확장이 필요하다.

참 고 문 헌

- [1] A. Versey and Ajay Paul Sravana, "CASE as Collaborative Support Technologies," *Communication of ACM*, pp.83-94, Jan. 1995.
- [2] Ellis, C., Gibbs, S. and Rein, G., "Groupware Some Issues and Experiences," *Communication of ACM*, Vol.34, No.1, Jan. 1991.
- [3] James D.Palmer and N. Ann Fields, "Computer Supported Cooperative Work," *IEEE Computer*, pp.15-17, May 1994.
- [4] Chris Greenhalgh, Steven Benford, "MASSIVE : A Collaborative Virtual Environment for Teleconferencing," *ACM Transactions on Computer-Human Interaction*, Vol.2, No.3, pp.239-261, September 1995.

- [5] Hagsand, Kista, "Interactive Multiuser VEs in the DIVE System," IEEE MultiMedia 1996.
- [6] Schooler, E. M., "Conferencing and Collaborative Computing, *ACM Multimedia Systems*, Vol.4, No.5, pp.210-225, 1996.
- [7] Gust, P., "SharedX : X in a Distributed Group Work Environment," *Proc. Of 2nd Annual X Technical Conference*, Jan. 1998.
- [8] Wladimir, M., "The Application Sharing Technology." <http://landru.unx.com/DD/advisor/docs/jun95/jun95.minenko.shtml>
- [9] NetMeeting, <http://www.microsoft.com/ie/ie3/netmtg.htm>
- [10] ProShare, <http://www.intel.com/comm-net/proshar/techinfo/index.htm>
- [11] Wolf, K. H., Froitzheim, K., and Schulthess, P., "Multimedia Application Sharing in Heterogeneous Environment," *ACM Multimedia 95*, San Francisco, California, Nov 5-9, 1995.
- [12] Lori, M., "Dialing in to Win 95," *PC Week*, Feb. 1996.
- [13] WinDD User Information, http://www.tek.com/Network/_Displays/Support/windd/windduser/main.htm
- [14] Sakairi, T., Shinozaki, M., and Kobayashi, M., CollaborationFramework : A toolkit for Sharing Existing Single-User Applications without Modification, in Proceedings of the Third Asia-Pacific Computer and Human Interaction, July 1998.
- [15] The VirtualReality Modeling Language International Standard ISO/IEC 14772-1:1997, Copyright©1997 The VRML Consortium Incorporated. <http://www.vrml.org/Specifications/VRML97/>
- [16] 정진호, 박현진, 양현승, "ShareIT : 영상 캡처를 이용한 이중 윈도우 시스템에서 응용프로그램 공유 시스템," 정보과학회논문지(C) 제4권 제6호, pp.865-875, 1998.12.
- [17] 김희선, 김상욱, "어플리케이션 뷰 공유 시스템에서 셀 비교를 통한 다중 사용자 인터페이스 관리," 정보과학회논문지(B) 제25권 제11호, pp.1677-1685, 1998.11.



탁진현

e-mail : tak2023@hanmail.net

1992년 인하대학교 전자계산공학과 학사

1994년 인하대학교 전자계산공학과 석사

1995년~현재 인하대학교 전자계산공학과 박사과정

1994년~1998년 (주)중앙네트웍시스템 기술부 과장

1999년~현재 (주)IB Internet 개발부장

관심분야 : 멀티미디어, 웹기반교육, 분산가상환경, CSCW, 인터넷방송



이세훈

e-mail : seihoon@true.inhac.ac.kr

1985년 인하대학교 전자계산학과 학사

1987년 인하대학교 전자계산학과 석사

1996년 인하대학교 전자계산공학과 박사

1987년~1990년 해병대 전산실 분석장교

1991년~1993년 비트컴퓨터 기술연구소 선임연구원

1999년~현재 (주)IB Internet 기술고문

1993년~현재 인하공업전문대학 전자계산기과 교수

관심분야 : 분산객체컴퓨팅, 멀티미디어, 소프트웨어공학, 원격교육



왕창종

e-mail : cjwangse@inha.ac.kr

1964년 고려대학교 물리학과 학사

1975년 성균관대학교 경영학과 석사

1979년~현재 인하대학교 전자계산공학과 교수

관심분야 : 소프트웨어공학, 분산객체컴퓨팅, 원격교육