

멀티미디어 공동 작업 환경에서 오류 상호 인지 및 제어

고응남*, 황대준**

*천안대학교 정보통신학부

**성균관대학교 전기 전자 및 컴퓨터공학부

e-mail : ssken@cheonan.ac.kr

djhwang@yurim.skku.ac.kr

An Error Mutual Awareness and Control on a Multimedia Collaboration Work Environment

Eung-Nam Ko*, Dae-Joon Hwang**

*Division of Information & Communication, Cheonan University

**School of Electrical, Electronics & Computer Engineering, Sungkyunkwan University

요 약

본 논문에서는 멀티미디어 공동 작업 환경에서 오류 상호 인지 및 제어에 대하여 기술한다. 상호 인지란 상호 작업 이해, 참여자 관계 설정 등 결합이라는 구현 기술을 말한다. 결합에는 변화하는 내용을 인지하는 동적 인지와 자주 변하지 않는 정적 인지가 있는 데 제안하고자 하는 오류 제어는 멀티미디어 공동 작업 환경에서 오류 발생시 오류도 하나의 응용으로 취급한다. 인지의 구현 방법에는 파일 공유, 윈도우 복사, 윈도우 오버레이, 또는 윈도우 공유 등이 있는 데 본 연구에서는 윈도우 오버레이 기능에 세션 제어, 발언권 제어 등을 추가한 윈도우 공유 방법을 제안한다. 공동 작업 환경에 참여하는 모든 사용자들은 사용자의 명령, 또는 오류가 발생된 응용들을 제어 가능하며 상호 작용 가능하다.

1. 서론

컴퓨터 성능의 급속한 발전과 고속 네트워크 기술의 발달로 인한 오늘날의 컴퓨팅 환경이 개방화됨에 따라 컴퓨터 지원 협동 작업(CSCW: Computer Supported Cooperative Work) 응용은 화상 회의 시스템, 상호 협력적 원격 교육, 원격 진료, 그룹 게임 분야 등으로 실생활의 거의 모든 분야로 확대되고 있다.[1,2,3]. 최근 들어 이러한 컴퓨터 지원 협동 작업 환경이 증가하고 있는데 반하여 이러한 시스템에서의 접근 제어, 동시성 제어, 상호 인지 등 응용 오류에 대한 연구는 미흡한 실정이다[4,5].

본 논문에서는 멀티미디어 공동 작업 환경에서 오류 상호 인지 및 제어에 대하여 기술한다. 상호 인지란 상호 작업 이해, 참여자 관계 설정 등 결합이라는

구현 기술을 말한다. 결합에는 변화하는 내용을 인지하는 동적 인지와 자주 변하지 않는 정적 인지가 있는 데 제안하고자 하는 오류 제어는 멀티미디어 공동 작업 환경에서 오류 발생시 오류도 하나의 메시지로 취급한다. 인지의 구현 방법에는 파일 공유, 윈도우 복사, 윈도우 오버레이, 또는 윈도우 공유 등이 있는 데 본 연구에서는 윈도우 오버레이 기능에 세션 제어, 발언권 제어 등을 추가한 윈도우 공유 방법을 제안한다. 공동 작업 환경에 참여하는 모든 사용자들은 사용자의 명령, 또는 오류가 발생된 응용들을 제어 가능하며 상호 작용 가능하다.

본 논문의 구성은 2 장에서는 관련 연구, 3 장에서는 제안하는 멀티미디어 공동작업 공간에서 오류 상호 인지 및 제어, 4 장에서는 시스템 평가, 5 장에서는 결론을 기술한다.

2. 관련 연구

본 절에서는 기존의 멀티미디어 공동 작업 환경의 종류 및 구조에 대해서 기술한다. Shastra 는 Purdue 대학교에서 개발된 UNIX 를 기반으로 멀티미디어 협력 작업 설계 환경을 제공하는 시스템이다. 이 시스템은 상호 작용 과정의 모든 동작을 중앙 세션 관리를 통하여 하기 때문에 서버의 부담이 많아진다는 단점이 있다[6]. MERMAID 는 일본의 Kansai C&C Lab 과 NEC 사에서 개발된 분산형 응용 공유 구조를 선택하면서, 공유 이벤트의 분배를 이벤트 발송 부분에서 처리함으로써 다양한 응용의 지원을 고려하는 시스템이다[7]. MMConf 는 미국의 캠퍼리지에서 개발된 분산형 응용 공유 구조를 선택하였으며, X-윈도우즈를 기반으로 설계되어 있다[8]. CECED 는 SRI international 에서 개발된 중앙 집중형 구조와 복제형 구조의 혼합 구조를 지원하며, 화면 공유 개념을 확장하였다[9].

기존의 멀티미디어 공동작업 환경의 구조는 응용 구조에 따라 집중형(Centralized), 분산형(Distributed), 복제형(Replicated)으로 구분할 수 있다. 집중형 구조는 모든 구성 요소가 하나의 워크스테이션에서만 실행되는 구조이다. 이는 모든 요소가 하나의 워크스테이션에서만 실행되기 때문에 가장 간단한 방법이다. 분산형 구조는 구성 요소가 분산되어 있으며, 이들은 여러 워크스테이션에 걸쳐서 존재하는 구조이다. 복제형 구조는 분산형 구조의 변형된 형태로서, 대응하는 구성 요소가 아주 동일하거나 복제된다. 이는 각자의 워크스테이션에 실행에 필요한 모든 요소가 존재하는 구조이다.

공동 작업자들간의 관계, 작업의 종류를 알기 위한 방법으로 상호 인지(mutual awareness), 즉 결합(coupling)이라는 개념이 필요하다. 기존 시스템의 결합을 구현하는 방법에는 공유 대상과 구현 방식에 따라 분류할 수 있다[10,11]. 기존 시스템에서의 구현 방법은 파일 공유, 윈도우 복사, 윈도우 오버레이 등이 있다. 파일 공유는 다수의 사용자가 동일한 응용을 사용하며, 모든 사용자가 동일한 파일에 접근하고, 한번에 한 사용자만 읽기 쓰기가 가능하고 나머지는 읽기만 가능하다. 공유를 위해서 다른 조정 기법들이 제공되지 않기 때문에, 사용자들이 작업하는 데 필요한 조정은 모두 오디오 채널 등의 다른 미디어를 통해서 이루어져야만 하므로 다른 미디어가 반드시 필요하다. 윈도우 복사는 한 사용자만 응용을 사용하며, 다른 사용자들에게 윈도우 이미지를 주기적으로 단순히 복사함으로써 공동 작업 환경을 구현하는 방법이다. 모든 사용자들은 동일한 뷰, 즉, 응용 프로그램의 처리 결과를 가진다. 윈도우 오버레이는 복사된 윈도우 위에 간단한 스케치 도구를 겹쳐 놓는 방법으로, 오버레이 방법을 통해서 모든 사용자들은 동작할 수 있다.

기존 시스템들은 윈도우 오버레이 방식으로 구현한 경우가 많다. 한 순간에 한 사용자 만이 적극적인 사용이 가능하며 신뢰도와 호환성 문제 등을 처리해야 할 필요성이 있다.

3. 멀티미디어 공동 작업 환경에서의 오류 상호 인지 및 제어

본 논문에서는 멀티미디어 공동 작업 환경에서의 오류 상호 인지 및 제어에 대하여 제안한다.

본 논문에서는 윈도우 공유 방법을 제안한다. 공동 작업 환경에 참여하는 모든 사용자들은 사용자의 명령, 또는 오류가 발생된 응용들을 제어 가능하며 상호 작용 가능하다.

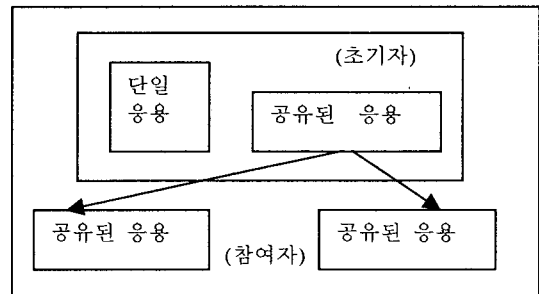
3.1 멀티미디어 공동 작업 환경

그룹통신을 지원하기 위한 방법은 TCP/IP 나 UDP/IP 를 이용하고 전송계층의 프로그램 지원으로 그룹 통신을 지원하는 방법과, 멀티캐스트를 이용하는 방법이 있을 수 있다. 본 연구의 제안 모델에서는 IP 계층에서 호스트에 제공하는 멀티캐스트를 이용하였다. UDP/IP 브로드캐스팅도 다수의 호스트에 동시 전송이 가능하지만 호스트를 지정할 수 없어 그룹 전송을 하지 못하고 그룹의 가입과 탈퇴가 자유롭지 못하기 때문에 IP 멀티캐스트를 사용한다.

CAI 는 Common Application Interface, GSM 은 Global Session Manager, LSM 은 Local Session Manager, MCP 는 Multichannel Port, UDP 는 User Datagram protocol, IP 는 Internet Protocol 의 약어이다. 여기서 LSM 과 MCP 를 포함하는 계층은 DooRaeMTP 라 불리며 동시에 다수의 독립적인 세션을 개설할 수 있는 다중 세션을 지원하고 또한 하나의 세션은 미디어 데이터 별로 각각의 채널을 갖는 다채널 방식을 지원한다. 특히 오류 발생시 이것을 제어할 수 있는 채널도 할당한다.

3.2 윈도우 공유 방식

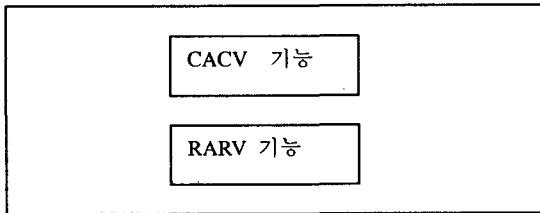
윈도우 공유 방식은 (그림 1)처럼 공동 작업 환경에 참여하는 사용자들에게 각자의 워크스테이션에 독립적인 작업을 할 수 있도록 보장한다. 이는 공유 영역을 윈도우 단위로 설정할 수 있도록 하므로, 각자의 개인적인 공간을 유지할 수 있고 공유가 설정된 윈도우만 다른 참여자들이 접근 가능하므로 보안성의 측면에서도 유리하다.



(그림 1) 윈도우 공유 방식

3.3 오류 상호 인지 및 제어

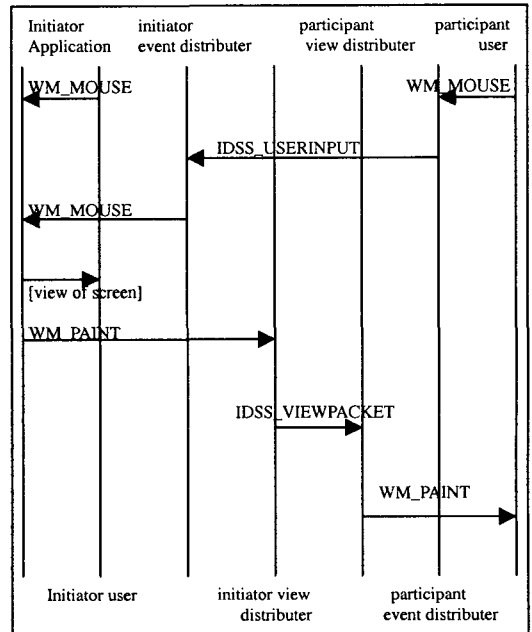
제안하는 오류 상호 인지 및 제어 시스템은 윈도우 공유 방식을 사용한다. 응용 및 오류 공유 구조로는 분산 복제형 구조(CARV)이다. 분산 복제형 구조는 (그림 2)처럼 중앙 집중형 구조(CACV)와 복제형 구조(RARV) 두개의 장점을 취합한 형태로 모든 시스템이 프로세스를 수행할 수 있는 능력을 가지고 있지만 복제형 구조와 같이 완전한 대칭적인 구조를 가지고 있지는 않다. CACV 구조의 기능은 View 분배 기능을 중점적으로 지원한다. 또한, 생성자 측의 응용으로 입력을 전달하기 위해서 자신의 인스턴스 영역 내의 사건을 판별하여 생성자 측의 응용 공유 인스턴스의 입력으로 하는 기능을 가지며, 자신과 관계 없는 참여자의 사건을 인식해서 무시하는 기능도 있다. 또한 참여자가 발생시킨 사건을 공유 되는 응용의 입력으로 사용하기 위해서 가공하는 기능도 가진다. RARV 구조의 기능은 이벤트 분배 기능과 공유 되는 응용들간의 일치화 문제를 해결한다. 특히, 다른 참여자의 사건을 동시에 자신의 응용 프로그램으로 전달해서 모든 참여자 사이의 응용에 대한 일관된 처리를 행하며, 참여자들 간에 공유 되는 응용들의 일관성을 감시하여 항상 일관성을 유지하도록, 공유 윈도우 이동, 크기 조정, 사건 사상 함수 변환 등을 행한다. 특히, 사건 사상 함수는 모든 참여자들 간에 공유되는 응용이 반드시 같은 위치에 있지 않아도 공유가 원활히 이루어질 수 있도록 지원한다. 즉, 각각의 복제된 응용들은 서로 상대적인 위치를 유지하더라도 복제된 윈도우 내의 위치만 일정하게 유지가 되면 공유에 지장이 없다. 이렇게 함으로써 각자의 사용자는 응용성 있는 응용의 배치를 가능케 하며, 각자의 독립적인 작업을 방해하지 않으면서 공동 작업을 수행해나갈 수 있다.



(그림 2) 분산 복제형 구조의 구성

CACV 구조에서의 응용 및 오류 공유 과정은 (그림 3)처럼 먼저, 참여자가 공유되는 응용 및 오류 사건을 발생시키는 데서 시작한다. 사용자가 발생시킨 사건은 윈도우 메시지의 형태로 참여자 측 사건 분배기로 재지향 되고, 이는 다시 생성자 측 사건 분배기로 네트워크를 통해서 전송된다. 생성자 측 사건 분배기는 수신한 사건을 다시 공유되는 응용 프로그램으로 재지향 한다. 여기까지가 참여자 측에서 생성된 사건이 생성자 측 응용까지 전달되는 경로이다. 생성자 측의 응용은 전달 받은 사건을 수행해서 수행 결과인 view 를 화면 상에 생성하며, 이를 각자의 화면으로 출력한다. 이렇게 해서 사용자들은 각자의 워크스테이션에 응용 프로그램을 가지고 각자 생성한 사건을 사건 분배기를 통해서 분배하는 방법으로 각자의 응용 프로그램을 가지고 공동 작업을 수행한다.

view 를 화면 상에 생성하며, 이를 생성자 측 뷰 분배기가 감지해서 전송할 수 있는 형태로 압축한 후, 이를 여러 개의 전송 가능한 패킷으로 만들어서 참여자 측으로 보낸다. 참여자 측에서는 뷰 분배기가 생성자 측에서 보낸 패킷 들을 받아서 다시 하나의 뷰 정보로 재구성해서 압축을 해제한 다음, 참여자 측 화면으로 출력한다. 이렇게 해서 참여자는 응용 프로그램을 가지지 않고서도 생성자의 응용 프로그램을 통해서 공동 작업을 수행할 수 있다.



(그림 3) CACV 구조에서 응용 및 오류 공유 과정

RARV 구조에서의 응용 및 오류 공유 과정은 (그림 4)처럼 한 사용자가 발생시킨 응용 및 오류 사건은 윈도우 메시지의 형태로 사건 분배기로 재지향 되고, 이는 다시 다른 사용자들의 사건 분배기로 네트워크를 통해서 전송된다. 다른 사용자들의 사건 분배기는 수신한 사건을 다시 공유되는 응용 프로그램으로 재지향 한다. 여기까지가 한 사용자 측에서 생성된 사건이 다른 사용자들이 소유한 공유 응용까지 전달되는 경로이다. 다른 사용자들의 응용은 각자 전달 받은 사건을 수행해서 수행 결과인 view 를 화면 상에 생성하며, 이를 각자의 화면으로 출력한다. 이렇게 해서 사용자들은 각자의 워크스테이션에 응용 프로그램을 가지고 각자 생성한 사건을 사건 분배기를 통해서 분배하는 방법으로 각자의 응용 프로그램을 가지고 공동 작업을 수행한다.

