

## 원격 공동 실험을 위한 플랫폼 (A Platform for Remote Collaborative Experiment)

김 상 욱 <sup>†</sup> 진 민 <sup>\*\*</sup> 손 종 경 <sup>\*\*\*</sup> 김 우 년 <sup>\*\*\*\*</sup> 김 정 미 <sup>\*\*\*\*</sup>  
(Sangwook Kim) (Min Jin) (Jongkyung Sonn) (Woonyon Kim) (Jeongmi Kim)

**요 약** 본 논문은 원격 공동 실험을 위한 플랫폼인 PCS(Platform for Collaborative System) 개발에 관한 연구이다. PCS는 특정 응용 분야를 목적으로 하지 않고, 모든 분야의 공동 실험 시스템을 개발할 수 있는 범용의 공동 실험 플랫폼이다. PCS는 공동 실험에 필요한 객체인 통신, 세션, 사용자, 어플리케이션, 미디어, 메시지 객체와 이들에 대한 관리 메커니즘 함수를 가진 관리 객체로 구성된다. 관리 객체는 공동 실험 객체와 이들을 관리하는 오퍼레이션으로 구성되며, 관리 객체 사이의 상호작용을 통하여 제어 정보 및 데이터를 처리한다. 또한, PCS는 공동 실험을 위한 어플리케이션 공유 기술을 통하여 단일 사용자 어플리케이션을 다중 사용자 어플리케이션으로 활용할 수 있도록 하고, 공유 어플리케이션을 통한 원격 기기 제어 기능을 제공한다. PCS 플랫폼은 원격 공동 실험을 위한 환경을 제공하며, 여러 공동 실험 시스템의 하부 구조로 사용될 수 있다.

**Abstract** This paper is concerned with the development of PCS(Platform for Collaborative System), which is a platform for development of remote collaborative experiment systems. Platform does not aim at the development of special-domain applications. This system implies the common development environments which can develop all-domain applications. PCS consists of collaborative experimental objects such as communication, session, user, application, media, message object and management objects that have management functions. Management objects are made up of collaborative experimental objects and operations which manipulate control information and data. It also supports application sharing for making single user interface of experimental applications to multi-user interface. Application sharing also supports instruments control on the remote site. PCS platform supports total environments for remote collaborative experiment and can be used as infrastructure to all kinds of collaboratory systems.

### 1. 서 론

본 논문은 원격 공동 실험을 위한 플랫폼인 PCS

(Platform for Collaborative System) 개발에 관한 연구이다. 본 논문은 플랫폼 개발을 위하여 일반적인 공동 작업 시스템에 필요한 구성요소를 정립하고, 실험 시스템에 필요한 기능을 정의함으로써 공동 실험 플랫폼인 PCS를 구성한다. 또한, 개발된 플랫폼 기반에서 원격 실험 기기를 이용하여 분산 공동 실험을 수행할 수 있는 시스템을 구현한다.

원격 공동 실험 시스템은 화상 회의[1, 2], 원격 교육[1, 2], 공동 편집[3], 멀티미디어 메일, 작업 흐름[1, 2] 지원과 같은 기존의 그룹웨어 시스템 및 그룹웨어 플랫폼[4]을 기반으로 한다. 그러나, 기존의 그룹웨어 플랫폼과 시스템들은 사람들과의 공동 작업에 중점을 두고 개발되었기 때문에 실험 기기가 추가된 실험과 같은 환경에 적절하지 않다. 따라서, 원격지에서 실험을

· 본 연구는 1997년도 학술진흥재단 학제간연구(97-012-E0002)의 연구결과임

<sup>†</sup> 종신회원 : 경북대학교 컴퓨터과학과 교수  
swkim@cs.knu.ac.kr

<sup>\*\*</sup> 종신회원 : 경남대학교 정보통신학부 교수

<sup>\*\*\*</sup> 비회원 : 경북대학교 생물교육과 교수

sonnjik@kyungpook.ac.kr

<sup>\*\*\*\*</sup> 정회원 : 경북대학교 컴퓨터과학과

kimwn@woorisol.knu.ac.kr

kimjm@nca.or.kr

논문접수 : 1998년 10월 8일

심사완료 : 2000년 2월 17일

위한 기능을 추가해야 한다. 원격 공동 실험 시스템은 고가 실험 장비에 대한 공유와 연구 결과물에 대한 공유 의식 형성을 통하여, 고가 실험 장비에 대한 중복 투자 방지, 실험 장비 사용에 대한 배타성 해결, 물리적인 지역적 거리감 해소, 실험을 위한 물리적 여행 감소, 여러 분야 전문가들 사이의 신속한 정보 교환, 시간 절약 및 장비 사용의 효율성 향상 등을 목적으로 한다[5, 6, 7].

지금까지 개발된 시스템에는 InterMed, UARC, DCEE 등이 있다. InterMed 시스템은 7 계층 구조의 공동 의학 연구를 위한 시스템으로써, 생의학 분야를 위한 도구와 서비스를 제공한다[8, 9]. UARC(Upper Atmospheric Research Collaboratory)는 여섯 개의 기관에서 그린랜드의 태양풍 관측 실험 장비를 이용할 수 있는 환경을 제공한다. 실험 장비의 멀티미디어 데이터를 교환하고, 분석 및 저장하는 기능을 제공한다[7]. DCEE(Distributed Collaboratory Experiment Environments)는 LabSpace, REE(Remote Experimental Environment), EMSL(Environmental and Molecular Science Laboratory), Spectro-Microscopy 공동 실험의 네가지 분야로 구성된다. LabSpace는 가상 공유 작업 공간을 제공하고, 실제 연구실을 메타포로 하여 가상 환경에서 연구를 진행 할 수 있는 실험 하부구조를 생성한다. REE는 D-IIID 토카막(tokamak) 실험 장비를 원격지에서 운영할 수 있는 환경을 제공한다. EMSL은 분자 과학 분야의 공동 연구를 지원하는 시스템으로써 자료 공유, 원격지 통신, 원격 기기 제어, 소규모 및 대규모 공동 실험을 제공한다[5, 6, 10, 11]. Spectro-Microscopy 공동 실험은 원격지에서 빔라인(Beamline) 7 장비를 원격지에서 접근하고, 장비에서 생성된 데이터를 분석하기 위하여 장비에서 제공하는 프로그램을 수정하여 여러 사용자가 공유할 수 있는 기능을 제공한다.

기존의 공동 실험 시스템은 특정 응용 분야를 목적으로 개발되었다. 이것은 해당 분야의 실험이나 장비를 지원하는 소프트웨어만을 제공한다. 또한, 제공되는 소프트웨어가 단일 사용자 인터페이스이므로 다중 사용자 인터페이스를 제공할 수 없다. 따라서, 다중 사용자 인터페이스를 지원하기 위하여 기존 어플리케이션을 수정하거나 새로이 개발해야 한다[5]. 그러므로, 특정 분야의 공동 실험 시스템에 국한되지 않고 모든 실험 시스템을 개발할 수 있는 범용 플랫폼의 개발이 필요하며, 이러한 범용 플랫폼은 단일 사용자 소프트웨어를 수정 없이 사용할 수 있어야 한다. 또한, 범용 플랫폼은 원격 모니터링, 원격 제어와 같은 실험 관련 기능을 제공해야

한다.

본 논문은 모든 응용 분야의 실험 시스템을 개발할 수 있는 환경을 제공하고, 기존의 단일 사용자 어플리케이션을 수정하지 않고 다중 사용자가 공유할 수 있는 어플리케이션 공유 방법을 제시한다. 어플리케이션 공유 방법은 프로토콜 분배를 통하여 원격 실험 기기를 제어하는 단일 사용자 어플리케이션 수정 없이 다중 사용자가 공동으로 사용할 수 있는 환경을 제공한다. 이것은 실험 기기가 존재하는 원격 실험실에만 실제 어플리케이션이 존재하고, 공동 실험에 참가 중인 다른 지역에는 그 어플리케이션의 새도우를 제공하는 방법으로 실현된다. 새도우는 실제 어플리케이션이 없는 사이트에서 표시되는 사용자 인터페이스를 의미한다. 새도우를 통하여 실험에 참가한 사용자는 동일한 실험 상황을 파악할 수 있고, 공유 인터페이스를 통하여 원격 제어를 할 수 있다. 본 논문에서 제시하는 플랫폼 PCS는 모든 분야의 원격 공동 실험 시스템 개발을 위한 환경을 제공함으로써 특정 분야에 국한되지 않는 공동 실험 시스템을 구축할 수 있다.

논문은 제 2 절 PCS의 구조, 제 3 절은 공동 실험 플랫폼 PCS의 구성요소, 제 4 절은 공동 연구 및 실험 환경, 제 5 절은 구현, 제 6 절 결론으로 구성된다.

## 2. PCS의 구조

공동 실험 시스템은 공동 작업 시스템의 기술을 기반으로 한다. 본 절은 공동 실험 시스템을 위한 플랫폼인 PCS의 구조를 설명하고, PCS에서 채택하고 있는 공동 작업 시스템의 환경과 실험에 필요한 환경을 기술한다.

### 2.1 구조

PCS는 원격 공동 실험을 위한 플랫폼으로써 공동 작업을 위한 기능과 실험을 위한 기능을 제공해야 한다. 공동 작업은 텔레포인팅, 공유 작업 공간, 멀티미디어 환경 등을 통하여 공유 작업 공간을 제공한다. 이를 위하여 공동 작업 시스템은 사용자 관리, 세션 관리, 미디어 관리, 멀티미디어 입출력 관리 등이 필요하다. 실험 시스템은 원격 실험 기기 제어, 어플리케이션 공유, 전자 실험 노트, 실험실 모니터링 등의 기능을 제공함으로써 원격 실험을 위한 환경을 제공한다. 이러한 공동 실험을 위한 환경을 제공하는 PCS 플랫폼은 그림 1의 구조를 가진다.

PCS 플랫폼은 통신 관리, 메시지 관리, 미디어 관리, 사용자 관리, 세션 관리, 멀티미디어 입출력 관리, 어플리케이션 공유 관리 요소를 가지며, 이들은 공동 작업 환경과 원격 실험 환경을 제공한다.

<b>공동 작업 기본 환경</b> 화이트보드 화상회의 채팅 멀티미디어 메일			<b>원격 연구 실험 환경</b> 전자실험노트 원격 실험 기기 제어 어플리케이션 공유 원격 실험실 환경 모니터링		
미디어 관리	어플리케이션 공유 관리	사용자 관리	세션 관리	멀티미디어 입출력 관리	메시지 관리
통신 관리(TCP/IP)					

그림 1 PCS의 구조

PCS 플랫폼은 맨 하위 계층에 통신 관리 요소가 있다. 통신 관리 요소는 데이터 전송 메커니즘을 통하여 신뢰성을 보장하는 데이터 전송, 멀티캐스트, 파일 전송 등의 기능을 제공한다. 신뢰성 보장 데이터 전송은 실험 데이터의 무결성을 보장하며, 멀티캐스트는 공동 실험 참가자들에게 데이터를 분배한다. 통신 관리 계층위에는 공유 관리 계층이 있으며, 이는 여러 실험 참가자들에 의해 공동으로 필요한 정보를 관리한다. 공유 관리 계층에는 메시지 관리, 미디어 관리, 사용자 관리, 세션 관리, 멀티미디어 입출력 관리, 어플리케이션 공유 관리 요소가 있다. 각 요소에 대한 설명은 3.1절에서 기술한다.

가장 상위 계층은 상호 작용 계층으로써 사용자 사이의 상호작용과 원격 실험 기기와의 상호작용을 지원한다. PCS의 작업 환경은 사용자 사이의 기본적인 통신 수단을 제공하는 공동작업 기본 환경과 실험 장비를 이용하여 실험을 할 수 있는 원격 실험 환경이 있다.

**2.2 공동작업 기본 환경**

공동 작업 기본 환경은 화상 회의, 화이트 보드, 채팅 등과 같은 기능으로서 공동 작업 시스템에 필요한 기능이다. 이러한 기능은 주로 원격지 사용자와의 의사 교환 및 통신을 위한 것이며, 공동 실험에서도 필요하다. 화상회의는 원거리에 있는 참가자와 얼굴을 보면서 대화를 할 수 있는 기능을 제공하며, 공동 작업을 수행하고 있는 상대방을 확인 할 수 있다. 화상회의는 실시간으로 오디오와 비디오를 전송하며, 이들 사이의 동기화가 필요하다. 동기화는 참조 지점 방법과 큐 방법[12]을 이용한다. 화이트보드는 공유 작업 공간에서 의사 교환 기능을 하며, 화상회의의 기능을 보조한다. 화이트 보드는 작업 공간의 공유를 지원하며, 텍스트, 그래픽, 이미지 등의 단순 미디어를 이용한 그리기 및 단순 편집 기능을 제공한다. 채팅은 토의 환경을 제공하며, 개인적인 일이나 비밀 대화를 진행 할 때 사용된다.

**2.3 원격 실험 환경**

원격 실험 환경은 인간과 실험 기기 사이의 상호작용을 지원하고 가상 실험실에서 원격 실험이 가능한 환경을 제공한다. 가상 실험실은 공동 실험에 참가한 참가자의 위치를 의미하고, 원격 실험실은 실험기기가 존재하는 실제 실험실의 위치를 의미한다. PCS는 실험 환경을 위하여 전자 실험 노트, 어플리케이션 공유, 원격 기기 제어, 원격 실험실 모니터링 기능을 제공한다. 이러한 환경은 원격 공동 실험에서 원격 제어 환경을 지원하며, 실제 실험실에 존재하는 것과 유사한 환경을 제공한다. 그림 2는 공동 실험 환경이다.

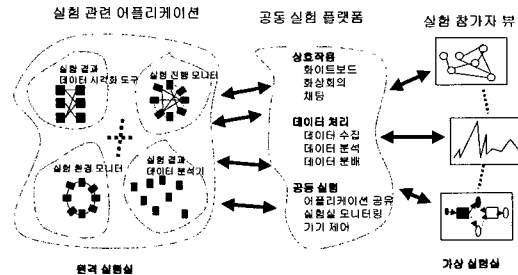


그림 2 공동 실험 환경

원격 실험실에는 실험 기기와 실험 관련 어플리케이션들이 있고 가상 실험실에는 지역적으로 분산된 실험 참가자들이 있다. 공동 실험 플랫폼은 가상 실험실과 실제 실험실의 상호 작용을 지원한다. 공동 실험 플랫폼은 인간과 인간의 상호작용을 지원하기 위한 기능, 데이터 처리를 위한 기능, 공동 실험을 위한 기능을 포함한다. 특히, 어플리케이션 공유, 실험실 모니터링, 원격 기기 제어와 같은 실험을 위한 기능은 분산된 원격지 참가자들이 공동 실험에 참가할 수 있는 환경을 제공한다.

**3. 공동 실험 플랫폼 PCS의 구성요소**

**3.1 구성요소**

공동 실험 플랫폼 PCS는 공동 작업 기본 요소와 원격 연구 실험 요소를 이용하여 원격 공동 실험을 지원한다. PCS는 관리 객체로 구성되며, 관리 객체는 여러 객체와 이들을 관리하기 위한 오퍼레이션으로 구성된다.

관리 객체에는 통신 관리, 메시지 관리, 세션 관리, 사용자 관리, 미디어 관리, 어플리케이션 관리 객체가 있다. 공동 실험 객체는 원격 공동 실험을 수행하기 위해서 필요한 객체로서 통신 객체, 메시지 객체, 세션 객체, 사용자 객체, 미디어 객체, 어플리케이션 객체가 있

다. 공동 실험 객체는 원격 공동 작업을 지원하기 위해 필요한 공동 작업 객체와 실험을 지원하기 위한 실험 객체로 구분한다. 공동 실험 객체는 실험과 관련된 어플리케이션 객체가 속하고, 나머지 객체는 공동 작업 객체로 분류한다. 그림 3은 PCS를 구성하는 객체들을 UML[13]의 클래스 다이어그램으로 나타낸 것이다. 그림3은 전체에서 중요한 부분만 표현하였으며, 각각의 세부 사항은 아래에 설명되어 있다.

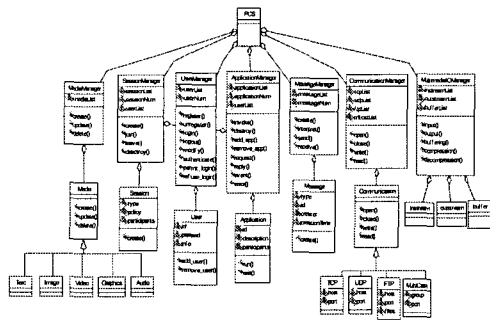


그림 3 PCS 구성 요소

3.1.1 통신 관리 객체

통신 관리 객체는 공동 실험 플랫폼의 가장 하부에서 공동 작업의 기본 조건인 통신 수단을 제공하기 위하여 통신 채널을 개방하거나 폐쇄한다. 통신 관리 객체는 TCP(tcp), UDP(udp), 멀티캐스트(multicast), FTP(ftp) 등의 통신 객체를 인스턴스로 가진다. 통신을 관리하기 위한 오퍼레이션은 채널 생성(open)과 채널 소멸(close), 데이터의 전송(write) 및 수신(read)이 있다.

3.1.2 미디어 관리 객체

미디어 관리 객체는 공동 실험 과정에서 생성되는 모든 미디어 객체를 관리한다. 화이트 보드를 통하여 원격 사용자와 의사를 교환할 경우, 사용자는 화이트보드에 텍스트, 그래픽, 이미지 객체 등을 생성, 수정, 삭제하는 행위를 한다. 이러한 객체들은 사용자의 행위에 따라 관리된다. PCS가 제공하는 미디어 객체에는 텍스트(t), 그래픽(g), 이미지(i), 오디오(au), 비디오(v)가 있다. 미디어 객체 관리 오퍼레이션은 생성(create), 변경(update), 삭제(delete)이 있고, 변경 오퍼레이션(update)은 미디어 객체의 위치, 모양, 색, 스타일 등의 속성을 수정한다.

미디어 객체 중에서 오디오 및 비디오와 같은 대용량의 미디어 데이터를 처리하기 위하여 멀티미디어 입출력 관리가 필요하다. 실시간 오디오 및 비디오 전송은

원격 공동 실험에 필수적인 사항이다. 멀티미디어 입출력시 관리되어야 하는 객체는 입력 스트림(instream), 출력 스트림(outstream), 버퍼(buffer)가 있으며, 오디오와 비디오 객체에 대하여 생성한다. 멀티미디어 입출력 객체의 관리 오퍼레이션은 입력(input), 출력(output), 버퍼링(buffering), 압축(compression), 압축풀기(decompression)이 있다.

3.1.3 사용자 관리 객체

공동 실험 플랫폼은 여러 사용자가 실시간 동기 공동 작업을 지원한다. 공동 작업 중인 사용자는 필요에 따라 참가하거나 탈퇴하는 등의 동적인 특성이 있다. 그리고, 이러한 사용자는 시스템의 정당한 사용자임을 보장하기 위하여 등록, 관리한다. 사용자의 등록, 해지 등의 관리는 시간에 따라 변화하는 동적인 특성이 있다.

사용자 객체는 사용자 식별자(id), 비밀번호(password), 기타 사용자 관련 정보(othinfo)를 튜플로 가진다. 사용자 객체는 임의의 한 사용자와 연관되며 그 사용자를 독립적인 형태로 표현하는 수단이다. 사용자 객체에 대한 관리 오퍼레이션은 등록(register), 해지(unregister), 로그인(login), 로그아웃(logout), 정보 변경(modify) 등이 있다.

3.1.4 세션 관리 객체

세션 관리 객체는 사용자가 생성하는 세션 관련 이벤트에 대하여 세션 객체를 관리하는 것으로, 세션 관리 오퍼레이션에 의하여 수행된다[14, 15]. 세션 객체 관리는 세션 객체와 관리 오퍼레이션으로 구성된다.

세션 객체는 공동 작업 방식에 따라 실시간형, 비실시간형, 개방형, 폐쇄형, 예약형, 즉시형이 있다. PCS 시스템은 실시간형(rt), 비실시간형(nrt), 개방형, 폐쇄형을 제공하며, 개방형과 폐쇄형의 경우, 작업 그룹 개념을 이용하여 세션의 개방 여부를 결정한다. 개방 여부에 의하여 개인 세션(private), 공유 세션(shared), 공개 세션(public)을 정의한다. 이는 작업의 공유 정도를 다양하게 함으로써 공동 작업의 다양한 형태를 지원한다.

세션 객체에 대한 관리 오퍼레이션은 세션 객체 생성, 세션 참가, 세션 탈퇴, 세션 객체의 소멸을 관리하기 위하여 각각 생성(create), 참가(join), 탈퇴(leave), 소멸(destroy) 오퍼레이션이 존재한다[15]. 세션 관리 객체는 이들 오퍼레이션을 통하여 세션 객체를 관리한다.

3.1.5 메시지 관리 객체

메시지는 사용자의 이벤트를 공동 실험에 참가한 다른 사용자에게 전송하기 위하여 형상화된 형태로서 통신 객체를 통하여 전송된다.

메시지 객체는 메시지형(*type*), 메시지를 발생시킨 사용자 ID(*uid*), 기타 메시지로 전송되어야 하는 정보(*otherinfo*)를 가진다. 메시지 객체에 대한 관리 오퍼레이션은 메시지 생성(*create*), 번역(*interpret*), 전송(*send*), 접수(*receive*) 등이 있다.

3.1.6 어플리케이션 관리 객체

어플리케이션 객체는 실행 프로그램으로써 어플리케이션 공유기에 의하여 접근되는 독립된 단위이다. 어플리케이션 객체에는 어플리케이션 관리 모듈이 제공하는 관리 오퍼레이션에 의하여 관리된다.

어플리케이션 객체는 어플리케이션 ID(*id*)와 기타 어플리케이션에 관련된 정보(*otherinfo*)를 튜플로 가진다. 어플리케이션 객체에 대한 관리 오퍼레이션은 어플리케이션 구동(*invoke*), 종료(*destroy*), 어플리케이션 추가(*add\_app*) 및 삭제(*remove\_app*) 등이 있다.

어플리케이션 객체는 실험을 위해 이용하는 윈도우 어플리케이션이며, 실험의 종류와 기기에 따라서 여러 어플리케이션이 될 수 있다.

3.2 구성요소의 상호작용

그림 4는 공동 실험 플랫폼 PCS 구성 요소들을 UML의 상호작용 다이어그램을 이용하여 표현한 것이다. 각각의 관리 객체는 개별 객체를 독립적으로 관리하며, PCS는 관리 객체 사이의 상호 작용을 통하여 원격 공동 실험을 지원한다.

관리 객체는 다른 관리 객체와 데이터 및 제어 정보를 주고 받는 것에 의해 필요한 처리를 수행한다.

그림 4는 PCS의 구성요소인 객체와 관리 객체들 사이의 상호작용이다. 그림 4에서 사각형은 공동 실험 객

체와 관리 객체를 표시하고, 연결선은 데이터와 제어의 이동 경로이다. 또한, 화살표는 제어와 데이터의 이동 방향을 표시하며, 화살표와 함께 표시된 텍스트는 관리 객체의 오퍼레이션이다. 텍스트 중에서 번호는 오퍼레이션의 순서를 나타내며, 그림에서는 중복된 오퍼레이션에 대해서는 표시하지 않았다. 그림은 몇 가지 오퍼레이션을 동시에 표시한 것이다.

통신 관리 객체는 TCP와 UDP를 서비스하기 위하여 tcp, udp의 두 서비스를 열고(open), 사용자 접속을 대기한다. 사용자가 접속하면 데이터를 읽어서 메시지 형태로 만들고(3 ~ 5) 메시지의 정보를 번역한다(6). 번역된 정보가 사용자 로그인 정보라면 정당한 사용자인지 검증을 하고(7), 사용자를 리스트에 추가하고 로그인이 성공적임을 참가자에게 알린다(8,9). 사용자가 특정 세션에 참가를 하게 되면 앞과 동일한 순서로 메시지를 해석하고 메시지의 내용에 따라서 세션에 참가(14, 15)하게 된다. 참가한 세션에서 화이트보드를 이용하여 공동작업을 진행하는 경우, 현재 사이트의 참가자가 선을 그렸을 경우, 이에 대한 그래픽 데이터를 생성하고(19), 메시지 관리 객체를 통해 참가한 다른 참가자에게 전달한다(20). PCS는 이와 같은 순서로 참가자들 사이에 공동 작업 또는 공동 실험을 지원한다.

4. 원격 공동 연구 및 실험

원격 공동 실험은 다중 사용자와 원격 실험 기기의 상호작용을 지원함으로써 분산 공동 실험실을 구축한다. 원격 실험 기기와의 상호작용은 원격 실험 기기 제어와 원격 실험실 모니터링의 형태로 나타난다. 기존의 실험

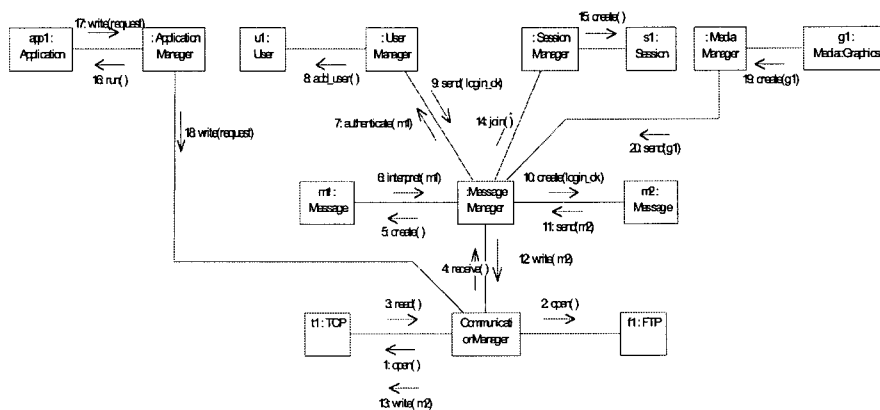


그림 4 PCS 구성요소의 상호작용

기기 제어 어플리케이션은 단일 사용자용이기 때문에 여러 사용자가 공유할 수 없다. 이러한 점은 기존의 실험 기기 어플리케이션이 공동 실험 시스템을 구축하는데 이용될 수 없고, 복수 사용자 인터페이스를 지원하기 위하여 새로이 개발되어야 함을 의미한다. 본 절은 실험 기기에 독립적이며, 기존 제어 어플리케이션의 수정 없이 공동 실험을 지원할 수 있는 어플리케이션 공유를 제시한다. 어플리케이션 공유는 실험 기기 제어 어플리케이션을 객체 단위로 공유하며 프로토콜 분배 방법을 사용한다.

4.1 어플리케이션 공유

어플리케이션 공유는 어플리케이션과 윈도우 시스템 사이의 직접 연결을 어플리케이션 공유기를 통과하도록 함으로써, 어플리케이션과 윈도우 시스템 사이의 송수신 정보를 얻을 수 있다. 어플리케이션 공유기는 수신된 정보를 공동 실험에 참가한 새도우 사이트에 전달하고, 새도우 사이트는 윈도우 시스템에 처리를 요청함으로써 호스트 사이트와 새도우 사이트에 동일한 인터페이스를 유지한다. 새도우 사이트의 이벤트는 동일한 방법으로 호스트 사이트의 어플리케이션에 전달되어 처리된다. 여기서, 어플리케이션이 존재하는 사이트는 호스트 사이트이며, 어플리케이션의 인터페이스만 표현되는 곳은 새도우 사이트라 한다.

그림 5는 X 윈도우 시스템에서 어플리케이션 공유 시스템의 모델이다.

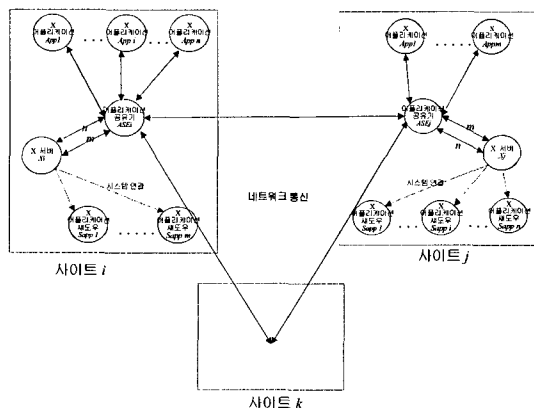


그림 5 어플리케이션 공유

어플리케이션 공유기  $ASE_i$ 는 X 서버  $X_i$  및 다른 사이트의 어플리케이션 공유기  $ASE_j$ ,  $ASE_k$  와 연결을 설정하고 클라이언트 프로그램의 수행을 대기한다. ASE와 X서버의 최초 연결 설정은 텔레포팅에 이용

된다. 사이트  $i$ 에서 입의의 X 어플리케이션  $App_i$ 가 수행되면  $App_i$ 에 대해서 사이트  $i$ 는 호스트 사이트가 되고, 연결이 설정된 다른 사이트  $j$ ,  $k$ 는 새도우 사이트가 된다.  $ASE_i$ 는  $App_i$ 와  $X_i$ 를 연결하기 위하여 X 서버와의 연결을 추가로 설정한다. 호스트의  $ASE_i$ 는  $X_i$ 와  $App_i$  사이의 X 프로토콜을 수신하여  $ASE_j$ ,  $ASE_k$ 로 전송하고,  $ASE_j$ ,  $ASE_k$ 는  $X_j$ ,  $X_k$ 에 전송하여 호스트 사이트의  $App_i$ 와 동일한 어플리케이션 새도우인  $SApp_i$ 를 생성하고 유지한다. 새도우 사이트인  $j$ 나  $k$ 의  $SApp_i$ 에서 발생한 이벤트는  $X_j$ ,  $X_k$ 로부터  $ASE_j$ 로 이벤트 정보를 전송한다.  $ASE_i$ 는 수신된 이벤트를  $App_i$ 에 전송하여 이벤트를 처리한다.

어플리케이션 공유는 X 어플리케이션과 X 서버 사이에 가상 서버를 두어서 어플리케이션과 X 서버 사이의 X 프로토콜을 여러 사이트의 X 서버에게 분배하여 호스트의 어플리케이션과 동일한 새도우를 생성한다. 어플리케이션 공유는 기존의 단일 사용자 어플리케이션을 수정하거나 새로이 개발하지 않고, 여러 사용자가 공동 실험을 진행할 수 있는 공유 인터페이스 환경을 지원한다. 또한, 실험 어플리케이션이 지원하는 실험 기기 제어를 원격지에서 조작 가능하며, 어플리케이션이 지원하는 시뮬레이션 기능 등의 여러 가지 기능을 지원할 수 있다. 그림 6은 어플리케이션 공유 절차이다.

```
// initial state
ASEi connect to Xi's DISPLAY:0
add initial system resource to resource map in ASEi
open Xi's DISPLAY:1 port for client's connection
open port for shadow system's connection

// HOST SITE : ASEi
connection request from shadow ASEj
accept connection request from ASEj
send ASEj's initial information for comparing initial system state

// multiplex protocol
connect client Appi from Xi's DISPLAY:1
do until Appi is exit
  get protocol pk from Appi
  multiplex pk to every shadow ASEj
  if Appi has some pk+1 from ASEj
    send pk+1 to Appi
  end if
end do

// SHADOW SITE: ASEj
do until pk is CLIENT_EXIT
  read pk from ASEj
  map pk resource id to ASEj's resource id
  send pk to ASEj's Xj's DISPLAY:0
  if ASEj has event
    get pk+1 from ASEj's Xj
    map pk+1's resource id to ASEj's resource id
    send pk+1 to ASEi
  end if
end do
```

그림 6 어플리케이션 공유 절차

4.2 연구 실험 어플리케이션

본 절은 원격 공동 실험 시스템에서 어플리케이션 공유를 이용한 실험 환경을 보이고, 이 실험 환경이 어플리케이션 공유를 어떻게 이용하는지 설명한다. 어플리케이션 공유 기술을 이용한 실험 환경은 원격 실험 기기 제어와 원격 실험실 모니터링이 있다.

4.2.1 원격 실험 기기 제어

원격 기기 제어는 원격지에서 실험 기기를 제어하여 원격 실험이 진행 될 수 있는 환경을 제공한다. 원격 기기 제어는 원격지에서 실험 기기를 직접 조작하거나, 실험 기기의 여러 파라미터를 변경하는 기능을 수행한다. 또한, 기기 제어 어플리케이션이 제공하는 실험 과정 모니터링, 실험 결과 데이터의 분석, 실험 시뮬레이션 등의 기능을 어플리케이션 공유를 통해서 제공한다. 실험 기기의 제어는 실험 기기 제어 어플리케이션을 통해서 수행된다. 따라서, 원격 기기 제어가 가능한 실험 기기는 실험 기기 제어 어플리케이션을 통해서 제어가 가능한 실험 기기라 제한된다. 그림 7은 원격 기기 제어의 동작 모델이다.

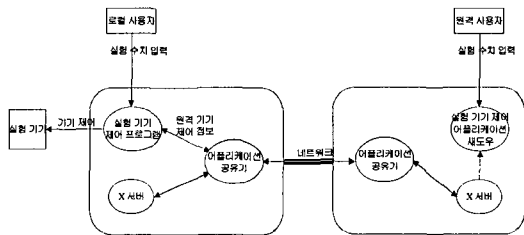


그림 7 원격 기기 제어

실험 기기가 있는 원격 실험실의 실험 기기 제어 어플리케이션을 어플리케이션 공유를 통해서 가상 실험실에서 디스플레이한다. 가상 실험실에서 참가자는 공유 인터페이스를 통해서 실험과 관련된 기기의 동작을 제어한다. 제어 정보는 원격 실험실의 실제 어플리케이션으로 전송되고, 원격 실험실의 어플리케이션은 이 정보를 처리하여 실험 기기의 정보를 변경한다. 어플리케이션의 실행에 의하여 실험 기기는 전송된 파라미터 정보에 따라서 동작을 수행한다. 어플리케이션의 동작에 의한 인터페이스의 변경은 가상 실험실로 전달되어 공유 인터페이스를 동일하게 유지하며, 동작의 수행은 실험실 모니터링 카메라를 통하여 비디오로 전달된다. 실험에 참가하고 있는 여러 가상 실험실에서도 동일한 처리가 가능하며, 모두 동일한 뷰를 공유하게 된다. 이러한 원격 기기 제어는 원격지에서 실험을 수행하기 위해서 필

수적이다.

여러 사용자가 동일한 기기에 대하여 제어를 수행하고자 하면, 사용자들 사이에 제어 명령의 충돌이 발생할 수 있다. 이러한 충돌 문제는 발원된 제어를 통하여 해결된다. 각 세션은 역할에 따라서 모든 참가자를 책임자와 회원으로 크게 분류하고 있으며, 발원된 제어는 각 세션의 책임자에 의하여 수동적으로 처리된다.

4.2.2 실험실 환경 모니터링

실험실 환경 모니터링은 가상 실험실에서 원격 실험실의 환경, 실험 장비의 상태, 실험의 진행 상황, 공동 실험 참가자 등의 정보를 얻을 수 있도록 한다. 특히, 원격 실험실에서 실험을 진행하고 있는 것과 유사한 환경을 제공한다.

실험실 환경 모니터링은 실험실의 카메라를 통해 입력된 정보를 가상 실험실로 전송하며, 실험실의 현재 상황을 비디오로 제공한다. 따라서, 카메라의 파라미터를 설정 또는 변경할 수 있어야 하며, 이러한 파라미터에는 초점 조절, 확대/축소, 실험실 스캔, 카메라의 뷰 이동 등이 있다. 그림 8은 실험실 환경 모니터링의 모델이다.

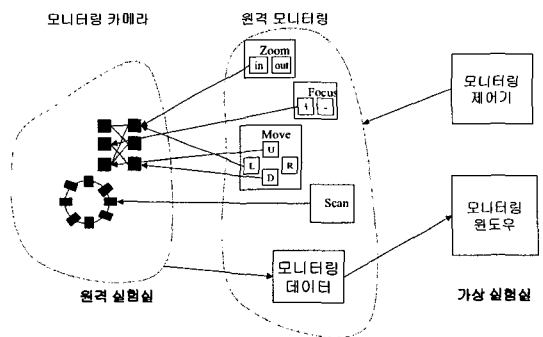


그림 8 실험실 환경 모니터링

참가자는 모니터링 제어기를 통하여 모니터링 카메라의 파라미터를 변경, 설정하고, 이러한 정보는 원격 모니터링 서비스를 통하여 원격 실험실의 카메라에 전달되어 카메라가 동작한다. 변경된 뷰는 카메라에서부터 공동 실험 참가자에게 전송된다.

5. 구현

공동 실험 시스템 CORES(Collaborative Research and Experiment System)은 Solaris 운영체제와 X 윈도우즈 환경에서 C++를 사용하여 개발되었다. 공동 실험 시스템은 공동 실험 플랫폼을 이용하여 클라이언트/

서버 구조로 개발되었다. 서버는 사용자 관리기, 정보 관리기, 세션 관리기, 메시지 관리기로 구성되고, 클라이언트는 미디어 관리, 어플리케이션 공유 관리, 메시지 관리, 멀티미디어 입출력 관리로 구성된다. 클라이언트의 실험을 위한 어플리케이션은 원격 기기 제어, 어플리케이션 공유, 실험실 환경 모니터링, 전자 실험 노트 등의 기능을 제공한다. 어플리케이션 공유기는 X 윈도우즈에서 사용하는 X 프로토콜을 이용하여 구현되었다. 어플리케이션 공유는 기기용 제어 어플리케이션을 여러 참가자가 공유할 수 있는 환경을 제공하며, 또한 제어 어플리케이션을 통한 원격 기기 제어 환경을 제공한다. 전자 실험 노트는 이미지, 오디오, 비디오 등을 공동 저장 및 편집할 수 있으며, 특정 페이지의 내용 및 마킹 정보를 다른 참가자에게 비실시간으로 전송할 수 있다. 화상회의는 실험에 참가한 참가자들이 실험과 관련된 토의를 진행할 수 있도록 하며, 화이트보드는 화상회의를 위한 부가 서비스 기능을 제공한다. 실험실 모니터링은 원격 실험실의 실험 진행 상황과 주위 환경 등을 모니터링 할 수 있도록 한다. 원격 기기 제어는 어플리케이션에 의해서 제어되는 실험 기기에 대해서 제공되며, 실험실 모니터링은 제어 카메라를 이용하여 구현되었다.

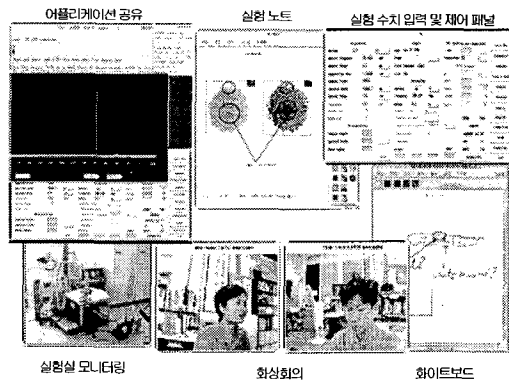


그림 9 공동 실험 시스템

그림 9는 공동 실험 시스템의 클라이언트를 표시한다. 공동 실험 참가자는 실험실 모니터링을 통해 원격지의 실험실의 내부 및 장비의 상태 등을 파악할 수 있다. 또한, 원격지 실험실의 참가자와 화상회의를 통해 실험과 관련된 정보를 교환할 수 있다. 실험을 진행하기 위하여 어플리케이션 공유를 통하여 원격 실험실의 실험 기기 제어 어플리케이션을 공유하게 된다. 어플리케이션 공유를 통하여 실험에 필요한 처리를 수행하며, 모니터링 카메라로 실험의 상태를 파악한다. 실험이 진행되는

동안 필요한 기록은 전자 실험 노트에 기록하며, 기록된 모든 정보는 실험 참가자들이 참조할 수 있으며 공동 편집도 가능하다. 원격 기기 제어는 어플리케이션 공유에 의해 제공된다.

PCS는 공동 실험을 개발할 수 있는 환경을 제공하며, 어플리케이션 공유를 통하여 기존의 실험 어플리케이션을 재 사용할 수 있다. 또한, 실험에 필요한 원격 제어 및 모니터링 기능을 제공한다. 다음 표 1은 기존 실험 시스템과의 비교이다.

6. 적용

본 절에서는 이상에서 개발한 플랫폼을 이용하여 실제 실험에 적용한 예를 설명한다. 실험은 유기물 구조 분석에 사용되는 핵자기 공명 분광계(Nuclear Magnetic Resonance Spectrometer: NMR)에 대한 공동 실험이다. 이 장치는 고체 또는 액체 상태의 유기 및 무기 화합물 시료 중에 들어 있는 양성자의 정성·정량 분석 및 그 화합물의 구조를 해석하는 데에 사용된다.

표 1 기존 시스템과의 기능 비교

	화상회의	전자노트	화이트보드	어플리케이션공유	분석도구	웹	시뮬레이션	원격제어	모니터링
PCS	○	○	○	○	※ <sup>1)</sup>	×	※ <sup>1)</sup>	○	○
D C E E	LabSpace	○	×	○	×	×	×	○	○
	REE	○	×	○	×	×	×	○	○
	EMSL	×	○	×	×	○	○	×	○
	Spectro Microscopy	○	○	○	×	○	×	○	○
UARC	×	×	○	×	×	×	×	○	
InterMed	×	×	×	×	×	○	×	○	
ScienceSpace	×	×	×	×	×	×	○	×	
CMDA	×	×	×	×	○	×	×	○	

○ : 지원, × : 지원하지 않음  
 ※<sup>1)</sup> : 실험용 어플리케이션이 지원하는 경우 어플리케이션 공유를 통하여 이용 가능

이 장치는 마그네트, 분석 소프트웨어, 콘솔로 구분되고 모든 처리는 분석 소프트웨어인 VNMR에 의해 제어된다. VNMR은 Solaris 운영체제에서 동작하는 NMR 장치 제어 소프트웨어이다. 따라서, Solaris 운영체제에서 개발된 어플리케이션 공유 모듈을 이용할 수 있다. 아래 그림 10은 공동 실험 구성에 관한 것이다.



NMR 장비가 있는 실험실이 있고, 두 곳의 공동 실험 참가 사이트가 있다.

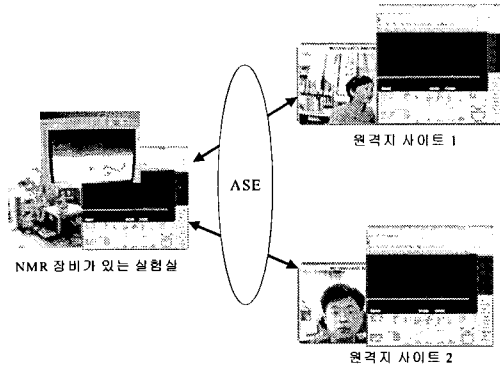


그림 10 NMR 공동 실험 구성

현재 구현된 시스템은 X 윈도우 프로토콜 공유를 통해서 공동 실험을 수행한다. 현재 X 윈도우 시스템에서 구현된 공유 시스템은 지각 참가자의 처리에 대한 문제점이 있다. 즉, 실험을 시작할 때 동시에 시작한 사람에 대한 처리는 지원되지만 지각 참가자에 대한 처리를 할 수 없다. 지각 참가자의 처리를 위해서는 발생한 프로토콜을 전체 또는 필요한 프로토콜을 선택하여 로깅한 후 이를 다시 반복하는 것에 의해 지각 참가자에 대한 처리를 지원해야 한다. 그리고, 참가자가 증가할 경우 어플리케이션 공유 모듈에서 1:1로 프로토콜을 전달하기 때문에 참가자의 수에 비례하여 VNMR의 수행 속도가 느려진다. 또한, 화상 회의 모듈 역시 속도가 느려지는 문제점이 있다. 어플리케이션의 속도가 느려지는 부분은 멀티캐스트 프로토콜을 채택하면 다소 완화될 수 있을 것으로 기대된다.

## 7. 결론

본 논문은 원격 공동 실험을 위한 플랫폼을 설계하고 구현하였으며, 플랫폼의 적용 예를 보였다. 공동 실험 플랫폼은 일반적인 원격 공동 실험을 위한 하부구조로서 일반적인 공동 실험 및 연구 도메인에 적용이 가능하다. 공동 실험 플랫폼 시스템인 PCS는 다음과 같은 장점을 가진다.

첫째, PCS 플랫폼의 구성요소는 모두 컴포넌트화 되어 있다. 구성요소의 컴포넌트화는 시스템 개발자가 간편하고 쉽게 사용할 수 있으므로, 공동 실험 시스템을 보다 빠르고 용이하게 구현한다.

둘째, PCS 플랫폼의 구성요소는 공동 시스템에 기본적인 기능으로서 범용성을 가진다. 즉, PCS 플랫폼의

구성요소는 공동 시스템에서 필수적이고 기본적이고 공통적인 기능으로 정의되어 있다. 그러므로 어떠한 목적의 공동 시스템을 구축하더라도 PCS 플랫폼의 구성요소는 사용될 수 있으며, 구성요소는 범용성을 지닌다.

셋째, PCS 플랫폼은 공동 실험 어플리케이션의 구축이 용이하다. 공동 실험 어플리케이션은 실험 기기 제어 프로그램이나 실험 데이터 처리 프로그램 등이 있으며, 이러한 프로그램은 단일 사용자 어플리케이션이다. PCS 플랫폼은 어플리케이션 공유 방법을 제공함으로써 기존의 단일 사용자 실험 어플리케이션을 다중 사용자용으로 수정하거나 다시 개발할 필요없이 다중 사용자용으로 쉽게 변경한다. 그러므로, 실험 어플리케이션의 다중 사용을 위한 공유화가 용이하다.

넷째, PCS 플랫폼은 다양한 분야의 어플리케이션을 쉽게 수용할 수 있다. 즉, PCS 플랫폼에서 개발된 공동 실험 시스템은 다른 부가적인 작업 없이 실험 대상, 방법, 장비의 변경에 적용될 수 있다. 공동 실험 시스템은 여러 종류의 장비와 방법을 지원할 필요가 있으며, 실험 장비나 방법이 변경될 때마다 필요한 어플리케이션을 수용할 수 있어야 한다. PCS 플랫폼은 어떠한 어플리케이션이라도 다중 사용자용으로 전환할 수 있으므로 실험 종류, 방법, 기기에 제한 없이 모든 분야의 실험을 수용할 수 있다.

다섯째, PCS 플랫폼은 원격지 존재감을 증대시킨다. 즉, 이미지, 오디오, 비디오 등의 멀티미디어와 원격 실험실 모니터링, 원격 실험 기기 제어 기능 등을 제공함으로써 원격지에 실제로 존재하면서 직접 조작하는 것과 유사한 실험 환경을 제공한다.

앞으로의 연구는 원격 모니터링과 같이 제어 카메라로부터 입력되는 비디오 데이터를 모든 참가자에게 품질을 보장하면서, 실시간으로 분배하는 방안에 대한 연구가 필요하다. 또한, 현재 개발된 플랫폼은 X 윈도우 시스템을 기반으로 개발되었기 때문에 MS 윈도우즈와 같이 이형의 윈도우 시스템은 지원할 수 없다. 따라서, 이형의 윈도우 시스템을 지원할 수 있는 방법에 대한 연구와 특히, 기존의 실험용 어플리케이션을 이기종 윈도우 시스템에서 공유할 수 있는 방법에 대한 연구가 필요하다.

## 참고 문헌

- [1] Ellis, Clarence, Gibbs, and Simon, "Groupware: Some issues and experiences," CACM, Vol. 34, No. 1, pp. 38-58, 1991.

[2] J. Grudin, "Computer-Supported Cooperative Work: History and Focus," IEEE Computer, pp. 19-26, May, 1994.

[3] M. Roseman and S. Greenberg, "TeamRooms: Network Places for Collaboration," Proceeding of the 1996 ACM Conference on CSCW, pp. 325-333, November, 1996.

[4] 김상욱, 김정미, 김태호, 이정훈, 배유석, "HITE: 객체 지향 멀티미디어 그룹웨어", 정보과학회논문지(B), pp. 1513-1521, Vol. 24, No. 12, 1997, 12.

[5] D. A. Agarwal, S. R. Sachs, and W. E. Johnston, "The Reality of Collaboratories," Computer Physics Communications, vol. 110, pp 134-141, May, 1998.

[6] R. T. Kouzes, J. D. Myers, and W. A. Wulf, "Collaboratories: Doing Science on the Internet," IEEE Computer, pp. 40-46, August 1996.

[7] Upper Atmospheric Research Collaboratory (UARC), <http://www.si.umich.edu/research/collaboratories.html>.

[8] Edward. H. Shortliffe, G. Octo. Barnett, James J. Cimino, Robert A. Greens, and Vimla L. Patel, "InterMed: An Internet-Based Medical Callab- oratory," April, 1996. [http://smi-web.stanford.edu/pubs/SMI\\_Reports/SMI-96-0622.pdf](http://smi-web.stanford.edu/pubs/SMI_Reports/SMI-96-0622.pdf).

[9] E. H. Shortliffe, O. Barmett, J. J. Cimino, R. A. Greenes, and V. L. Patel, "InterMed : An Internet -Based Medical Collaboratory," INET '96 Annual Meeting of the Internet Society, pp. 25-28, June 1996.

[10] William E. Johnston and Sonia R. Sachs, "Distri- buted, Collaboratory Experiment Environments (DCEE) Program: Overview and Final Report," February, 1997. <http://www-itg.lbl.gov/DCEE/Overview.fm.html>.

[11] R. Kouzes, "Building a Collaboratory in Environ- mental and Molecular Science," <http://www.emsl.pnl.gov:2080/docs/collab/presenations/collaboratory.WP.html>.

[12] R. Baker, A. Downing, K. Finn, E.Rennison, D.H. Kim, and T.H. Lim, "Multimedia Processing Model for a Distributed Multimedia I/O Systems," NOSSDAY, 1993.

[13] Martin Fowler, Kendall Scott, UML Distilled Applying The Standard Object Modeling Language, Addison Wesley Longman, Inc., 1997.

[14] W.K. Edwards, "Session Management for Collab- orative Applications," Preceedings of the 1994 ACM Conference on CSCW, pp. 323-330, October, 1994.

[15] 김정미, 김상욱, "멀티미디어 그룹웨어에서 작업 방식의 다양성에 근거한 세션 관리", 정보과학회 논문지(B), pp. 1369-1380, Vol. 25, No. 9, 1998, 9.

[16] 이용우, "온라인 원격리 협동연구와 온라인 원격리 실험기 술", 정보과학회지, pp.38-44, 1998. 7.



김 상 욱

1979년 경북대학교에서 컴퓨터공학으로 학사 학위를 취득. 1981년 서울대학교에서 컴퓨터과학으로 석사 학위를 취득. 1989년 서울대학교에서 컴퓨터과학으로 박사 학위를 취득. 1988년 ~ 현재 경북 대학교에서 컴퓨터과학과 교수로 재직 중. 관심분야는 컴퓨터 언어, 객체중심 컴퓨팅, 시각언어, 멀티미디어와 지식처리임.



진 민

1982년 서울대학교에서 컴퓨터과학으로 학사 학위를 취득. 1984년 한국과학기술 원에서 전산학으로 석사 학위를 취득. 1997년 University of Connecticut에서 컴퓨터과학으로 박사 학위를 취득. 1985년 ~ 현재 경남대학교 정보통신학부 교 수로 재직 중. 관심분야는 멀티미디어 데이터베이스, 객체 지향 데이터베이스, 제품정보 관리, 분산처리임.



손 중 경

1980년 서울대학교에서 동물학으로 학사 학위를 취득. 1982년 서울대학교에서 동 물학으로 석사 학위를 취득. 1987년 서 울대학교에서 동물학으로 박사 학위를 취득. 1984년 7월 ~ 1986년 3월 미국 코넬대학교 의과대학 생리학파 연구원. 1991년 7월 ~ 1992년 7월 미국 아이오와대학 생물학과 Post-doc. 1986년 4월 ~ 2000년 현재 경북대학교 사범대 학 생물교육과 교수. 관심분야는 동물 생태학, 원격 생태학 실험임.



김 우 년

1996년 안동대학교에서 컴퓨터공학으로 학사 학위 취득. 1998년 경북대학교에서 컴퓨터과학으로 석사 학위 취득. 2000년 현재 경북대학교 컴퓨터과학과 박사 수료. 관심분야는 컴퓨터언어, 분산 컴퓨팅, 멀티미디어 공동작업, 멀티미디어 통신, 작업흐름 관리임.



김 정 미

1995년 안동대학교에서 컴퓨터공학으로 학사 학위 취득. 1997년 경북대학교에서 컴퓨터과학으로 석사 학위 취득. 1999년 경북대학교 컴퓨터과학과 박사 수료. 2000년 현재 한국전산원 근무. 관심분야 는 휴먼인터페이스, 컴퓨터언어, CSCW, 작업흐름관리임.