원격 공동 실험 시스템 개발

김상욱, 김정미, 김우년 *
경북대학교 컴퓨터과학과 컴퓨터언어연구실

Development of Remote Collaborative Experimental System

Sangwook Kim. Jeongmi Kim, Woonyon Kim *
Department of Computer Science Kyungpook National University

요 약

본 논문은 원격 공동 실험 시스템인 PCS(Platform for Collaborative System) 개발에 관한 연구이다 PCS는 특정 응용 분야를 곡격으로 하지 않고, 모든 분야의 공동 실험 시스템을 개발할 수 있는 공동 실험 환경이다 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

1. 서론

본 논문은 일반적인 공동 작업 시스템에 필요한 구성요소를 정립하고, 특히 원격 공동 실험 시스템에 필요한 구성요소를 제시함으로써 원격 공동 실험을 위한 시스템 소프트웨어인 PCS(Platform for Collaborative System) 개발에 관한 연구이다

지금까지 개발된 시스템은 InterMed, UARC, DCEE 등이 있다 InterMed 시스템은 의학 연구를 위한 공동 시스템으로 7계층 구조를 가지며, 생의학 분야를 지원하는 툴과 서비스를 제공한다[2,6] UARC는 그린랜드의 태양풍 관측 실험 장비를 여섯 개의 기관에서 이용할 수 있도록 하고, 실험 장비로부터의 멀티미디어 데이터 교환, 분석, 저장 기능을 제공한다[7] DCEE(Distributed Collaboratory Experiment Environments)는 LabSpace, REE(Remote Experimental Environment), EMSL(Environmental and Molecular Science Laboratory), Spectro-Microscopy의 네가지 공동 실험으로 구성된다[3]. LapSpace는 실제 연구실을 메타포로 구성한 가상 공유 작업 환경에서 연구할 수 있는 실험 하부구조를 생성하고, REE는 D-IIID 토카막(tokamak)이라는 실험장비의 원격지 운영 환경을 제공한다. EMSL은 분자 과학 분야의 공동 연구 시스템으로 자료 공유, 원격지와의 통신, 원격 기기 제어, 대소규모의 공동 실험을 제공한다[1,3,4,5]. Spectro-Microscopy 공동 실험은 원격지의 Beamline 7 이라는 장비의 접근과 제어 프로그램의 공유 기능을 통하여 데이터를 분석한다[3]

이러한 기존 시스템은 특정 응용 분야를 목적으로 하기 때문에 해당 분야의 실험이나 장비를 지원하는 소프트웨어만 제공한다 또한, 제공되는 소프트웨어가 단일 사용자용으로 다른 사용자에 의하여 공동으로 사용될 수 없기 때문에 다중 사용자용으로 변경하는 부가적인 작업을 거쳐야한다[1]. 그러므로, 특정 응용 분야에 국한되지 않는 모든 분야의 실험 시스템을 개발할 수 있는 범용 시스템의 개발이 필요하고, 이러한 범용 시스템은 단일 사용자 소프트웨어의 수정없이 모든 실험이나 장비를 공동으로 제어할 수 있어야 효율적이다.

본 논문은 실험 시스템의 요소로 어플리케이션 공유 기법을 제시하며, 이 기법은 기존 실험 제어 어플리케이션을 프로토콜 분배와 공유를 통하여 수정없이 다중 사용자가 공유할 수 있기 때문에 모든 분야의 공동 실험 시스템을 개발할 수 있다 그러므로, PCS는 공동 실험 시스템 구축을 위한 범용 환경이다. 논문은 제 2절은 PCS 시스템의 구조, 제 3 절은 공동 실험 환경, 제 4절은 공동 실험 시스템의 구현, 제 5절은 결론이다.

2. PCS 시스템의 구조

PCS는 화이트보드, 화상회의, 채팅, 멀티미디어 메일과 같은 기능을 통하여 사용자 사이의 기본적인 통신 수단을 제공하는 공동작업 기본 환경과, 원격 실험 기기 제어, 실험 노트, 실험실 모니터링, 실험 장면 프리젠테이션 등의 기능을 통하여 실험을 할 수 있는 원격 실험 환경이 있다 그림 1은 PCS의 구조이다.

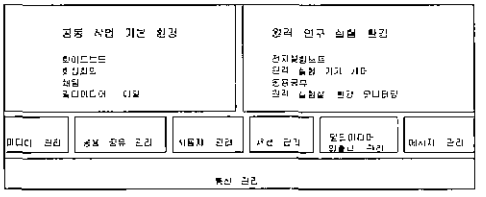

그림 1 PCS의 구조

이러한 환경은 통신 관리, 메시지 관리, 미디어 관리, 사용자 관리, 세션 관리, 멀티미디어 입출력 관리와 같은 기본적인 공동 작업 요소와 어플리케이션 공유 관리의 공동 실험 요소로 구현된다.

2.1 구성요소

PCS의 구성요소는 공동 실험 시스템에서 생성되는 객체에 대한 관리 메커니즘이며 관리 객체라 한다 공동 실험 플랫폼은 이러한 관리 객체의 결합이며 $E(\varepsilon)$로 표기한다 정의 1은 공동 실험 시스템의 정의이다.

정의 1 공동 실험 시스템 $E(\varepsilon)$

$E(\varepsilon) = \{ T(\tau, o^{\tau}), M(\mu, o^{\mu}), \Sigma(\sigma, o^{\sigma}), \Theta(\theta, o^{\theta}), \Delta(\delta, o^{\delta}), A(a, o^{a}) \}$,

$T(\tau, o^{\tau})$: 통신 관리 객체, $M(\mu, o^{\mu})$: 메시지 관리 객체,

$\Sigma(\sigma, o^{\sigma})$: 세션 관리 객체, $\Theta(\theta, o^{\theta})$: 사용자 관리 객체,

$\Delta(\delta, o^{\delta})$: 미디어 관리 객체, $A(a, o^{a})$ 어플리케이션 관리 객체

$E(\varepsilon)$는 통신 객체 관리 $T(\tau, o^{\tau})$, 메시지 객체 관리 $M(\mu, o^{\mu})$, 세션 객체 관리 $\Sigma(\sigma, o^{\sigma})$, 사용자 객체 관리 $\Theta(\theta, o^{\theta})$, 미디어 객체 관리 $\Delta(\delta, o^{\delta})$, 어플리케이션 객체 관리 $A(a, o^{a})$의 관리 객체를 지원하며, 관리 객체에 의하여 관리되는 객체를 공동 실험 객체라 한다 정의 2는 공동 실험 객체의 정의이다

정의 2 공동 실험 객체 ε

$\varepsilon = \Psi \cup \rho$,

$\Psi = \{ \tau, \mu, \sigma, \theta, \delta \}$,

$\rho = \{ a \}$

공동 실험 객체는 ε로 표기하고, 공동 객체 Ψ와 실험 객체 ρ로 구분한다. Ψ는 공동 작업시 기본적으로 관리되어야 하는 객체이고, ρ는 공동 실험시 필수적으로 관리되어야 하는 객체이다 Ψ는 통신 객체 τ, 메시지 객체 μ, 세션 객체 σ, 사용자 객체 θ, 미디어 객체 δ가 존재하며, 이들 객체는 각각의 관리 객체에 의하여 처리된다. ρ는 어플리케이션 객체 a가 존재하며, 이플리케이션 관리 객체에 의하여 처리된다. 정의 3은 각각의 관리 객체에 대한 정의이다.

정의 3. 관리 객체

$T(\tau, o^{\tau})$ $\tau = \{ tcp, udp, multcst \}$,

$o^{\tau} = \{ open, close, read, write \}$

$\Delta(\delta, o^{\delta})$ $\delta = \{ t, g, i, a, v \}$,

$o^{\delta} = \{ create, update, delete \}$.

$I(\iota, o^{\iota})$ $\iota = \{ instream, outstream, buffer \}$,

$o^{\iota} = \{ input,output,buffering,compression,decompression \}$

$\Theta(\theta, o^{\theta})$. $\theta = \{ u_i \mid u_i = <id, password, otherinfo>, i=0, \ldots n \}$,

$o^{\theta} = \{ register, unregister, login, logout, modify \}$

$\Sigma(\sigma, o^{\sigma})$ $\sigma = \{ rt, nrt, private, shared, public \}$,

$o^{\sigma} = \{ create, join, leave, destroy \}$

$M(\mu, o^{\mu})$ $\mu = \{ m_i \mid m_i = < type, uid, otherinfo >, i=0, \ldots n \}$,

$o^{\mu} = \{ create, interpret, send, receive \}$

$A(a, o^{a})$ $a = \{ a_i \mid a_i = < id, otherinfo >, i=0, \ldots n \}$

$o^{a} = \{invoke, destroy, request, reply, event, error \ldots\}$.

관리 객체는 객체와 적용되는 관리 오퍼레이션으로 정의된다 정의된 관리 객체는 관리 오퍼레이션을 통하여 서로 상호작용함으로써 공동 실험 시스템을 구성한다.

2.2 구성요소의 상호작용

구성요소의 상호작용은 관리 객체와 상호작용 오퍼레이션으로 표현된다 정의 4는 관리 객체의 상호작용을 정의한다.

정의 4 관리 객체의 상호작용 $X(E(\varepsilon), o^{\chi})$

상호작용 오퍼레이션 $o^{\chi} = \{ transmit \}$

관리 객체는 개별 객체를 독립적으로 관리하며, 관리 객체의 상호작용은 제어와 데이터 정보를 전달하는 것이다 상호작용 오퍼레이션은 제어와 데이터 흐름을 의미하는 *transmit* 이 존재한다. 관리 객체는 상호 작용 오퍼레이션에 의하여 서로 연관되어 공동 실험 시스템을 구성한다. 그림 2는 PCS의 구성요소의 상호작용을 나타낸다.

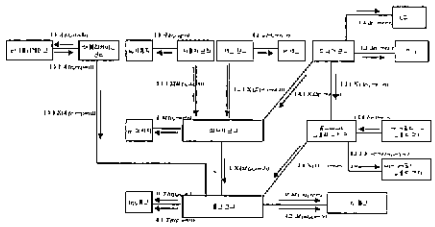

그림 2 PCS 구성요소의 상호작용

그림 2의 사각형은 공동 실험 객체와 관리 객체를, 연결선은 이들 사이의 연관성을, 화살표는 제어와 데이터의 이동 방향을, 화살표의 텍스트는 오퍼레이션을 나타낸다.

사용자 관리 객체는 사용자가 로그인하면 사용자 u_i에 대한 로그인을 처리하기 위하여 $\Theta(u_i, login)$의 관리 오퍼레이션을 수행하고, 이를 상호작용 오퍼레이션 $X(\Theta(u_i, login))$에 의하여 메시지 관리 객체로 전달한다. 메시지 관리 객체는 진달받은 로그인 내용을 $M(\mu_i, create)$을 통하여 메시지 μ_i를 생성하여 통신 객체로 전달한다. 통신 객체 관리는 $0\ T(tcp_i, open)$에 의하여 미리 오픈된 채널 tcp_i로 메시지를 진송한다 세션 관리 객체는 실시간 세션 rt_i를 생성하기 위하여 $M(rt_i, create)$의 관리 오퍼레이션을 수행하고, 이를 역시 메시지 관리 객체를 거쳐 통신 관리 객체로 전달되어 처리된다. 미디어 관리 객체는 $1.4{\cdot}\Delta(t_i, create)$의 텍스트 같은 정적 미디어가 생성될 경우, 역시 메시지 관리 객체를 거쳐 통신 관리 객체로 전달되어 처리된다 그러나, $1.3{\cdot}\Delta(v_i, create)$와 같은 오디오/비디오 스트림을 생성할 경우, 멀티미디어 입출력 관리 객체를 거쳐 통신 객체의 udp_i채널을 통하여 전달된다 어플리케이션 관리 객체는 임의의 어플리케이션 a_i를 공유하기 위하여 먼저 $A(a_i, invoke)$관리 오퍼레이션을 수행하고, 생성된 어플리케이션 a_i는 $A(a_i, request)$와 같은 요청 프로토콜을 통신 관리 객체로 전달함으로써 다른 사이트의 어플리케이션 객체 관리와 프로코톨을 공유한다. 다른 사이트의 어플리케이션 객체 관리는 전송 받은 이벤트 $A(a_i, event)$를 새도우 어플리케이션 a_i로 전달하여 처리되도록 한다.

3. 공동 실험 환경

본 절은 공유할 실험기기의 재한과 기존 제어 어플리케이션의 수정없이, 실험 기기를 공유, 제어할 수 있는 PCS의 근본 기술인 어플리케이션 공유 기술을 제안한다.

3.1 어플리케이션 공유 기술

어플리케이션 공유는 자신에게 없는 원격지의 어플리케이션을 사용하는 기능과 단일 사용자용 어플리케이션을 참가자들이 공동 작업할 수 있는 환경을 제공한다. 어플리케이션 공유기는 어플리케이션과 윈도우 시스템의 연결을 가로챔으로써 그 사이의 정보를 이용하여 어플리케이션을 공유한다. 그림 3은 X윈도우 시스템의 어플리케이션 공유 모델이다.

이플리케이션 공유기 ASE_i는 X서버 X_i 및 다른 사이트의 어플리케이션 공유기 ASE_j. ASE_k와 연결을 설정하고 클라이언트 프로그램의 수행을 대기한다. 사이트 i에서 임의의 X 어플리케이션 App_i가 수행되면 App_i에 대해서 사이트 i는 호스트 사이트가 되고, 연결이 설정된 다른 사이트 j, k는 새도우 사이트가 된다. 이때, App_i는 ASE_i와 연결되고, ASE_i는 X_i와 연결을 추가로 설정한다. 호스트의 ASE_i는 X_i와 App_i 사이의 정보인 X프로토콜을 수신하여 새도우 사이트의

ASE_j, ASE_k로 전송한다 수신된 프로토콜은 X_j, X_k에 의해 처리되어 App_i와 동일한 어플리케이션 새도우 $SApp_i$를 생성, 유지한다.

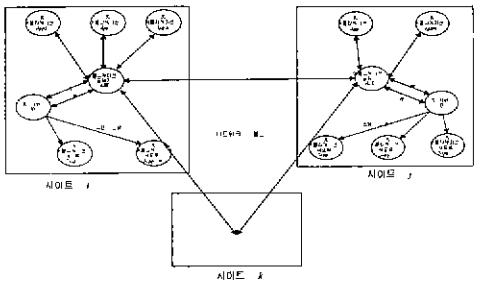

그림 3 어플리케이션 공유

새도우 사이트에서 발생한 이벤트는 ASE_j, ASE_k로부터 ASE_i로 전송되고, ASE_i는 수신된 이벤트를 App_i에 전송하여 이벤트를 처리한다. 어플리케이션 공유 기술은 기존의 단일 사용자 실험 어플리케이션을 수정, 재개발 하지않고, 다중 사용자가 공동 실험할 수 있는 공유 인터페이스 환경을 제공한다

3.2 실험 어플리케이션

실험 어플리케이션은 원격 실험 기기 제어와 실험실 모니터링 등이 있다.

3.2.1 원격 실험 기기 제어

원격 실험 기기 제어는 어플리케이션 공유를 통하여 원격지의 실험 기기 조작, 실험 파라메터 변경, 실험 과정 모니터링, 실험 데이터의 분석, 실험 시뮬레이션 등의 기능을 제공한다 그림 4는 원격 실험 기기 제어의 동작 모델이다.

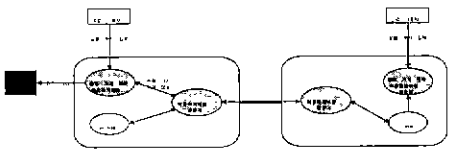

그림 4 원격 기기 제어

가상 실험실의 참가자는 공유 인터페이스를 통해서 실험 파라메터 변경, 기기 동작 등을 지시한다. 제어 정보는 어플리케이션 공유기를 통하여 원격 실험실의 실제 어플리케이션으로 전송된다 실제 어플리케이션은 전송받은 파라메터에 따라 실험 기기를 동작시키고, 이에 의한 인터페이스의 변경은 가상 실험실로 전달되어 동일한 인터페이스를 공유한다. 실험기기의 동작과정은 모니터링 카메라를 통하여 전달된다

3.2.2 실험실 환경 모니터링

실험실 환경 모니터링은 모니터링 카메라 제어 어플리케이션을 공유함으로써 실현되며, 원격 실험실 환경, 실험 장비 상태, 실험 진행 상황, 공동 실험 참가자 등의 정보를 비디오 인터페이스를 통하여 가상 실험실로 제공한다. 비디오 인터페이스는 모니터링 카메라의 초점 확대/축소, 스캔, 뷰 이동 등의 파라메터를 설정, 변경할 수 있다.

4. 구현

공동 실험 시스템은 Solaris 2.4와 X윈도우 환경에서 C++를 사용하여 개발되으며, 원격 기기 제어, 실험실 환경 모니터링, 전자 실험 노트 등의 실험 환경을 제공한다 어플리케이션 공유 기법은 X윈도우의 X프로토콜을 이용, 구현하였다 그림 5는 공동 실험 시스템의 가상 실험실을 표시한다.

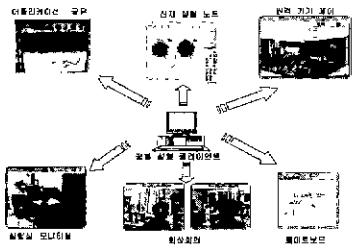

그림 5 공동 실험 시스템

공동 실험 참가자는 실험실 모니터링을 통해 원격 실험실과 기기의 상태를 관찰한다. 원격 실험실의 연구자와 화상회의를 통해 실험을 토론하며, 실험 기기 제어 어플리케이션의 새도우 인터페이스를 통하여 원격 실험 기기를 제어한다 실험 내용과 데이터는 전자 실험 노트에 기록하며, 기록된 정보는 모든 실험 참가자의 참조와 공동 편집이 가능하다.

5. 결론

본 논문은 원격 공동 실험을 위한 시스템을 설계하고 구현하였다 공동 실험 시스템은 일반적인 원격 공동 실험을 위한 하부구조로서 일반적인 공동 실험 분야에 적용이 가능하다

PCS 시스템은 어플리케이션 공유로써 단일 사용자용 어플리케이션을 다중 사용자가 공유할 수 있는 환경을 제공한다. 또한, 어플리케이션 공유는 단일 사용자 어플리케이션을 다중 사용자용으로 전환할 수 있기 때문에 모든 분야의 공동 실험을 지원할 수 있다. 실험실 환경 모니터링과 원격 기기 제어 기능을 제공함으로써, 원격지에서도 실험실의 실험 기기를 직접 제어할 수 있으며, 실험의 진행 과정을 알 수 있다.

앞으로의 연구는 웹에서 공동 실험을 수행할 수 있는 환경에 대한 연구가 필요하며, 웹에서 원격 모니터링 정보를 실시간으로 제공하는 방법에 관한 연구가 필요하다.

참고문헌

[1] D A Agarwal, et al, "The Reality of Collaboratories," Computer Physics Communications, vol. 110, pp 134-141, May, 1998

[2] Edward H Shortliffe, et al, "InterMed An Internet-Based Medical Callaboratory," April, 1996, http://smi-web.stanford.edu/pubs/SMI_Reports/SMI-96-0622.pdf

[3] William E. Johnston, et al, "Distributed, Collaboratory Experiment Environments (DCEE) Program' Overview and Final Report," February, 1997 http://www-itg.lbl.gov/DCEE/Overview.fm.html

[4] R Kouzes, "Building a Collaboratory in Environmental and Molecular Science," http://www.emsl.pnl.gov:2080/docs/collab/presentations/collaboratory WP.html.

[5] R. T. Kouzes, et al, "Collaboratories-Doing Science on the Internet," IEEE Computer, pp 40-46, August 1996

[6] E. H. Shortliffe, et al, "InterMed - An Internet-Based Medical Collaboratory," INET '96 Annual Meeting of the Internet Society, pp. 25-28, June 1996

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