

WebDAV 기반의 협업시스템을 위한 Jabber 메신저

이 흥 창[†] · 박 진 호[†] · 김 성 훈[†] · 이 명 준^{**}

요 약

Jakarta Slide는 아파치 프로젝트 중의 하나로 개발된 WebDAV 서버로서 인터넷 상에서 다양한 콘텐츠의 비동기적인 협업 활동을 지원한다. 하지만 WebDAV 프로토콜은 사용자와 그룹을 위한 가상 작업공간을 명시적으로 지원하지 않기 때문에, Jakarta Slide를 통하여 복잡한 협업을 지원하는 것은 매우 어려운 작업이다. CoSlide 협업시스템은 Jakarta Slide의 이러한 문제점을 개선하기 위하여 확장된 시스템으로서 그룹 작업을 위한 다양한 가상 작업공간을 지원함으로써 보다 효과적인 협업 환경을 제공한다. 본 논문은 CoSlide 협업시스템을 위한 CoJBother 메신저의 개발에 대하여 기술한다. CoJBother는 표준 Jabber 프로토콜을 기반으로 하는 JBother 메신저를 확장하여 개발되었으며, CoSlide 협업시스템의 사용자와 그룹을 지원하기 위하여 Jabber 프로토콜이 확장되었다. CoJBother는 일반 Jabber 사용자와 그룹에 대한 정보와 더불어 CoSlide 협업시스템의 사용자와 그룹의 정보를 표시한다. CoSlide를 통하여 협업을 수행하는 사용자는 CoJBother를 통하여 별도의 관리 작업 없이 그룹 구성원들과 그룹 채팅을 수행할 수 있다.

키워드 : WebDAV, CoSlide, JBother, Jabber

Jabber Messenger for a WebDAV-based Collaborative System

Hong-Chang Lee[†] · Jinho Park[†] · Seong-Hune Kim[†] · Myung-Joon Lee^{**}

ABSTRACT

Jakarta Slide is a WebDAV server developed as one of Apache projects, which supports asynchronous authoring for various contents on the server. Unfortunately, since the WebDAV protocol does not explicitly provide the workspaces for user groups, it is difficult to support complicated collaboration using Jakarta Slide. The CoSlide system is an extension of Jakarta Slide to address this problem, presenting more effective collaborative environment by providing various workspaces for group collaboration. In this paper, we describe the CoJBother Messenger for the CoSlide collaborative system. We developed CoJBother by extending the JBother messenger based on the standard Jabber protocol. To support the users and groups defined on the CoSlide system, we also extended the Jabber protocol. CoJBother shows the information on the CoSlide users and groups in addition to the standard Jabber users and groups. In particular, through CoJBother, the members of a group on CoSlide can initiate group chatting without any administrative overhead.

Key Words : WebDAV, CoSlide, JBother, Jabber

1. 서 론

Jakarta Slide[1]는 아파치 웹서버에서 동작하는 모듈로서 WebDAV 프로토콜을 통하여 다양한 콘텐츠의 비동기적인 협업 환경을 제공한다. Jakarta Slide에서 지원하는 WebDAV[2,3,4]는 HTTP/1.1을 확장한 웹 통신 프로토콜로서 다양한 콘텐츠의 비동기적인 협업을 인터넷을 통하여 지원하기 위한 표준 하부구조를 제공한다. 사용자는 Jakarta Slide 서버를 이용하여 여러 사용자와 원격지 자원을 공유하면서 비동기적으로 협업을 수행할 수 있다.

CoSlide 협업시스템[5]은 원격지의 자원과 가상공간을

활용하여 효과적인 협업을 지원하는 시스템으로서 Jakarta Slide를 사용하여 구축되었다. 기존의 Jakarta Slide는 개별 사용자와 사용자 그룹에 대한 가상공간에 대한 체계적인 지원이 부족하여 다양한 그룹을 통하여 다수의 사용자가 협업을 수행하기 위한 효과적인 환경을 형성하는 것이 매우 어려운 작업이었다. CoSlide는 Jakarta Slide의 기능을 바탕으로 개별 사용자와 사용자 그룹에 대하여 체계적으로 가상공간을 제공함으로써 보다 효과적인 협업 환경을 제공한다. CoSlide에서 제공하는 협업 환경을 사용하기 위해서는 CoSlide의 사용자로 등록되어야 하며, 개별 사용자는 여러 그룹에 소속되어 다양한 종류의 협업을 수행할 수 있다. 사용자는 CoDAView[6]와 같은 CoSlide 전용 클라이언트를 통하여 자신의 가상공간과 자신이 속한 그룹의 가상공간에 접근하여 다양한 협업 활동을 수행할 수 있다. 이렇게 여러 사용자가 다수의 그룹에 속하여 다른 사용자들

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성지원사업의 연구결과로 수행되었습니다(IIITA-2007-C1090-0701-0039).

† 준회원: 울산대학교 컴퓨터정보통신공학부 석사과정

** 정회원: 울산대학교 컴퓨터정보통신공학부 교수(교신저자)
논문접수: 2007년 4월 11일, 심사완료: 2007년 9월 22일

과 협업을 수행할 때 사용자들의 원활한 통신을 위해서는 주어진 협업 환경에 부합하는 동기적인 협업을 위한 도구가 필요하다.

본 논문에서는 여러 사용자가 CoSlide 협업시스템을 이용하여 협업을 수행할 때 그룹 구성원들 간의 원활한 통신을 지원하는 메신저인 CoJBother의 개발에 대하여 기술한다. CoJBother는 메신저를 위한 표준 프로토콜 중의 하나인 Jabber[8] 프로토콜을 사용하는 JBother[7] 메신저를 확장하여 구현되었으며 기존의 Jabber 사용자와 그룹 간의 통신과 더불어 CoSlide 협업시스템의 사용자와 그룹 간의 통신을 또한 지원하여 보다 효과적인 협업 환경을 제공할 수 있도록 도와준다. CoJBother에서 사용한 프로토콜인 Jabber 프로토콜은 XML로 구성된 XMPP 기반으로 플랫폼이나 서버, 클라이언트에 상관없이 사용될 수 있으며, Jabber를 사용하는 모든 사용자들과 연결이 될 수 있는 오픈 메신저 프로토콜이다. CoJBother는 JBother에서 사용한 Jabber 프로토콜을 확장하여 CoSlide의 사용자들 및 그룹 간의 통신을 지원할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 관련 연구에 대하여 살펴보고 3장에서는 CoJBother에서 CoSlide와의 연동을 위한 Jabber 프로토콜의 확장에 대하여 살펴본다. 그리고 4장에서는 확장된 Jabber 프로토콜을 사용하여 CoSlide 사용자와 그룹 간의 원활한 통신을 지원할 수 있도록 JBother를 확장한 CoJBother의 구조와 기능에 대하여 살펴본다. 마지막 5장에서는 결론 및 향후 연구 과제에 대하여 살펴본다.

2. 관련 연구

본 장에서는 본 논문에서 다루고 있는 Jabber 프로토콜과 JBother 그리고 CoSlide/CoDAView 협업시스템에 대하여 기술한다. Jabber는 오픈 메신저 프로토콜로서 XML로 구성된 XMPP 기반으로 정형화된 형식의 메시지를 주고받음으로써 플랫폼이나 서버, 클라이언트와 무관하게 여러 사용자가 다양한 형식을 통하여 서로 통신을 할 수 있도록 한다. JBother는 Jabber 프로토콜을 사용한 메신저로서 자바로 구현된 오픈소스 프로그램이다. CoSlide 협업시스템은 WebDAV 프로토콜을 사용한 Slide를 확장한 시스템으로 개인과 그룹에 대한 가상공간을 지원하여 보다 효과적인 협업 환경을 제공할 수 있다. CoDAView 는 CoSlide에서 제공하는 협업 환경을 효과적으로 사용할 수 있도록 도와주는 클라이언트 도구로서 GUI 인터페이스를 통하여 가상공간의 자원을 쉽게 관리할 수 있다.

2.1 Jabber

Jabber는 “실시간 메신저의 리눅스”라고 잘 알려진 오픈 소스 실시간 메신저 서비스이다. Jabber는 XML 프로토콜로서 인터넷 상의 두 요소들 간에 메시지나 프레젠테이션

〈표 1〉 Jabber 기술의 특징

Open	Jabber 프로토콜은 무료이며 공개 되어있고 이해하기 쉬운 구조로 되어있음
Standard	XMPP라는 인스턴트 메시징과 프레젠테이션 기술을 바탕으로 IETF에 의해 표준화됨
Proven	1988년 Jeremie Miller에 의해 개발된 이후 Jabber 기술은 현재 보다 안정화 되었고 수천 명 이상의 개발자들이 참여하고 있음
Decentralized	Jabber 네트워크의 구조는 이메일과 비슷하게 Jabber 사용자는 서버의 위치나 존재 유무를 알 필요 없이 이용할 수 있음
Secure	SASL과 TLS 등의 강력한 보안 시스템과 독립된 서버로 보안성 향상
Extensible	누구나 응용할 수 있는 XML 네임스페이스로 쉽게 확장할 수 있음
Flexible	Jabber 응용프로그램은 다양한 도구와 함께 연동될 수 있음
Diverse	다양한 기업과 오픈 소스 프로젝트에서 실시간 응용프로그램과 서비스 개발을 위해 Jabber 프로토콜을 사용함

```
<message from='juliet@capulet.com' to='romeo@montague.net'>
  <body>Wherefore art thou, Romeo?</body>
</message>
```

(그림 1) Jabber 인스턴트 메시지의 간단한 구조

(presence), 또 다른 구조의 정보를 실시간으로 교환할 수 있도록 한다. 표<1>은 Jabber의 특징을 나타낸다.

Jabber는 XML로 구성된 프로토콜로서 간단한 구조로 정의되어 있어 응용이 용이하고 다양한 플랫폼에서 사용이 가능하다. (그림 1)은 Jabber 프로토콜에서 간단한 인스턴트 메시지 전송의 구조다.

2.2 Jabberd

Jabberd[10]는 실시간 메시징 프로토콜인 표준 Jabber 프로토콜과 XMPP 프로토콜을 지원하는 서버이다. Jabberd는 C/C++로 구현되어있고 컴포넌트의 추가, 삭제가 쉬운 모듈 구조로 구성되어있으며 공개 소프트웨어로서 누구나 쉽게 사용할 수 있다. 본 연구에서는 표준 Jabber 프로토콜뿐만 아니라 협업시스템의 정보를 제공할 수 있도록 확장된 Jabber 프로토콜을 지원하기 위하여 Jabberd 서버를 확장하여 구현하였다.

2.3 JBother

Jabber 메신저는 Jabber 프로토콜을 이용하여 다양한 메신저의 기능을 확장하고 제공하는 메신저로서 JBother는 자바언어로 구현된 대표적인 표준 Jabber 메신저이다. JBother는 윈도우 스타일 형식으로 그룹 채팅을 지원하며 다양한 리소스를 제공하고 트랜스포트 기능으로 다른 메신저와의 연동도 가능하게 한다. 그리고 여러 스킨을 제공해 각 사용자의 메신저 환경을 다양하게 변경할 수 있으며 JRE 환경이 갖춰진 플랫폼이라면 어디서든 쉽게 사용할

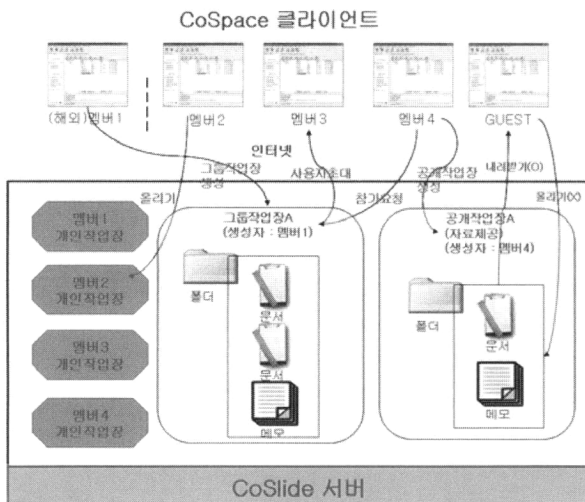


(그림 2) JBother 메신저의 기본 인터페이스

수 있다. 또한 다양한 플러그인을 통해서 여러 플랫폼에 맞게 최적화할 수 있다.

2.4 CoSlide 협업시스템

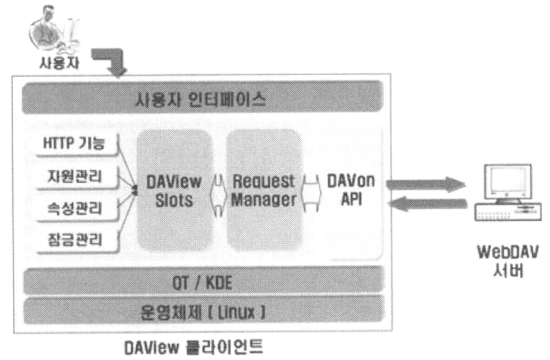
CoSlide 협업시스템은 원격지의 자원과 가상공간을 활용하여 효과적인 협업을 지원하는 시스템으로서 Jakarta Slide를 확장하여 구축되었다. 서버와 클라이언트로 구성된 CoSlide 협업시스템은 WebDAV 프로토콜을 이용하여 협업을 수행할 수 있는 효과적인 기능들을 제공한다. 다음의 (그림 3)은 CoSlide 협업시스템의 전체적인 동작 과정을 보여준다.



(그림 3) CoSlide 협업시스템의 동작

2.4.1 CoSlide 서버

CoSlide 협업시스템은 원격지의 자원과 가상공간을 활용하여 보다 효과적인 협업 환경을 제공하는 시스템으로서 Jakarta Slide를 확장하여 구현되었다. Jakarta Slide는 협



(그림 4) DAVView 클라이언트 구조

업을 지원하기 위한 표준 하부구조를 제공하는 웹 통신 프로토콜인 WebDAV를 사용하여 다양한 콘텐츠의 비동기적인 협업을 지원하는 협업 모듈이다. 하지만 기존의 Jakarta Slide는 협업을 하는 사용자와 그룹별 가상공간을 지원하지 않아 다수의 사용자와 그룹이 참여하는 협업 시 효과적인 협업 환경을 제공하기 어려웠다. Jakarta Slide의 문제점을 개선한 CoSlide는 협업에 참여하는 사용자와 그룹별로 가상공간을 제공하여 보다 효과적인 협업 환경을 제공한다. CoSlide 협업시스템을 사용하는 사용자는 자신만의 가상공간을 제공받게 되며 참여하는 그룹의 가상공간에 접근하여 그룹의 구성원들과 용이하게 협업을 수행할 수 있다.

2.4.2 CoDAView 클라이언트

CoDAView 클라이언트는 CoSlide 서버에 접근하여 협업을 수행할 수 있는 인터페이스와 자원 관리를 위한 WebDAV 명령을 수행하는 기능을 제공하는 도구이다. CoDAView 클라이언트의 인터페이스는 QT의 클래스들로 구성되어있고 시그널/슬롯의 이벤트 핸들 방식으로 이벤트를 처리한다. WebDAV 명령을 수행하기 위해서는 서버에 요청을 보내는 클래스들과 WebDAV 클라이언트 라이브러리를 통하여 사용자의 요청에 따른 자원 관리를 수행한다.

2.4.3 WebDAV 프로토콜

WebDAV 인터넷을 통하여 다양한 콘텐츠의 비동기적인 협업 저장을 지원하기 위한 프로토콜이다. WebDAV는

〈표 2〉 WebDAV에서 제공하는 메서드들

메서드	기능
HEAD, TRACE	네트워크 행동을 찾고 추적하는 기능
GET	문서를 서버에서 받음
PUT, POST	문서를 서버에 전달
DELETE	자원 삭제
MKCOL	컬렉션 생성
PROPFIND, PROPPATCH	자원의 속성을 검색하고 설정
COPY, MOVE	이름 공간 문맥 내에 있는 자원 관리
LOCK, UNLOCK	읽어 쓰기 방지 기능
OPTIONS	서버가 지원하는 메서드 출력

HTTP/1.1을 확장한 프로토콜로서 사용자들에게 원거리 서버들의 파일을 수정하고 관리할 수 있도록 한다. WebDAV의 기능은 속성을 이용한 자원관리, 덮어쓰기 방지, 이름공간 관리 등이 있다. WebDAV는 각각의 기능들을 위해 <표 2>에서 보는 것과 같이 HTTP1.1의 메서드도 사용하고 있으며 일부 메서드는 확장하였고 몇몇의 메서드들은 추가 되었다.

3. CoSlide를 위한 Jabber 프로토콜의 확장

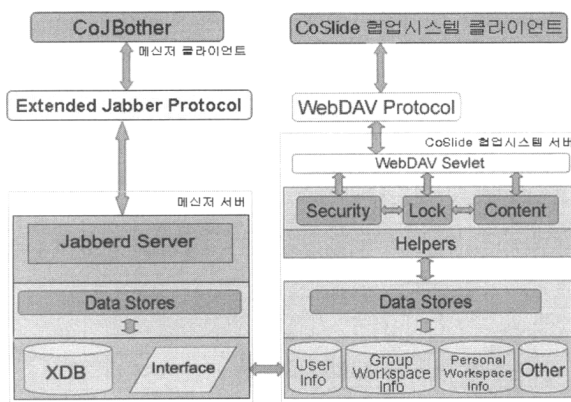
3.1 협업시스템과 연동되는 메신저의 구현

본 연구에서는 CoSlide 협업시스템을 통하여 협업을 수행할 때 참여자들 간에 원활하게 의사를 소통할 수 있도록 메신저를 개발하여 협업시스템과 연동하는데 그 목적이 있다. 메신징 시스템은 메신징 프로토콜, 메신저 서버 그리고 메신저 클라이언트로 구성된다. 본 연구에서 메신징 프로토콜은 메신징 프로토콜로 널리 사용되는 Jabber 프로토콜을 확장하여 협업시스템의 정보를 표현할 수 있도록 하였다. 또한, 메신저 서버는 표준 Jabber 프로토콜을 지원하는 Jabberd 서버를 확장하여 확장된 Jabber 프로토콜을 지원할 수 있도록 하였고 CoJBother 메신저 클라이언트는 JBother 메신저를 확장하여 구현하였다.

(그림 5)는 협업시스템의 구조와 함께 연동된 메신징 시스템의 구조를 보여준다. 개발된 Jabber 메신저는 협업시스템의 데이터베이스에 접근하여 협업시스템에서 협업그룹에 참여한 사용자와 그룹에 관한 정보를 가져온다. 그리고 확장된 Jabber 프로토콜을 통하여 클라이언트에게 제공하여줌으로써 사용자는 자신이 속한 협업 그룹의 사용자들과 편리하게 의사전달을 할 수 있다.

3.2 표준 Jabber 프로토콜

표준 Jabber 서버에서는 메신저 사용자의 정보를 저장하고 불러오기 위해 XML로 정의된 xdb 파일을 사용한다. xdb 파일은 사용자의 기본 정보 및 로그인 정보를 표현한다.



(그림 5) 협업시스템의 구조와 함께 연동된 메신징 시스템의 구조

```
<xdb>
<password xmlns='jabber:iq:auth' xdbns='jabber:iq:auth'>1</password>
<query xmlns='jabber:iq:register' xdbns='jabber:iq:register'>
<username>jinho</username>
<password xmlns='jabber:iq:auth'>1</password></query>
<query xmlns='jabber:iq:roster' xdbns='jabber:iq:roster'>
<item jid='john@203.250.77.124' name='john' subscription='both' /></query>
<foo xmlns='jabber:x:offline' xdbns='jabber:x:offline' />
<query xmlns='jabber:iq:last' last='1165563334' xdbns='jabber:iq:last' />
</xdb>
```

(그림 6) 사용자의 정보를 표현하는 xdb 파일

```
<query xmlns='jabber:iq:roster' xdbns='jabber:iq:roster'>
<item jid='jinop@203.250.77.124' name='jinop' subscription='to'>
<group>my friends</group>
</item>
<item jid='heinz@203.250.77.124' name='heinz' subscription='to'>
<group>my friends</group>
</item>
</query>
```

(그림 7) 사용자의 이웃 목록 정보

Jabber 서버는 이 정보를 이용하여 사용자에게 이웃 목록을 전송한다.

(그림 6)에서 <query> 태그는 Jabber 프로토콜에서 표현하는 Infor/Query 정보(iq) 중에서 특정 정보를 얻거나 전송하기 위한 쿼리 정보를 나타낸다. <xmlns>는 Jabber에서 일어나는 수많은 통신 이벤트에 대한 유형을 정의한 네임스페이스(NameSpace)를 나타낸다. 'jabber:iq:register' 네임스페이스는 접속한 사용자의 아이디와 패스워드 정보를 표현한다. 'jabber:iq:roster' 네임스페이스는 접속한 사용자의 이웃 목록을 표현하며 하위에 <item> 태그로 이웃들을 개별적으로 저장한다.

Jabber 서버는 xdb 파일로부터 사용자의 이웃 목록 정보를 얻는다. 그리고 클라이언트에게 다음과 같은 표준 Jabber 프로토콜을 통하여 사용자의 이웃 목록과 그룹 정보를 전달한다.

(그림 7)에서 각각의 <item> 태그는 등록된 이웃을 나타내는데 jid는 등록된 이웃의 Jabber ID를 나타낸다. <item> 태그의 하위 태그로 <group> 태그가 존재하는데 이 태그를 사용하는 경우, 클라이언트 인터페이스에서는 해당 사용자가 특정 그룹 안에 포함되어 있음을 보여준다. 위의 그림에서 사용자 jinop와 heinz는 my friends라는 그룹에 소속되어 있음을 나타낸다.

하지만 Jabber 프로토콜은 사용자의 이웃을 정의하는 <item> 태그에 <group> 태그를 하나 밖에 기술할 수 없기 때문에 한 사용자가 여러 그룹에 소속되는 것을 표현할 수 없다. 그래서 다수의 사용자가 여러 그룹에 속해 협업을 하는 환경을 지원하기 위해서는 표준 Jabber 프로토콜의 확장이 필요하다.

3.3 JBother에서 사용하는 Jabber 프로토콜

JBother는 표준화된 Jabber 프로토콜의 사용을 위해 Smack[9] 라이브러리를 사용한다. Smack은 인스턴트 메시

```
// Jabber 서버에 접속
XMPPConnection connection = new XMPPConnection("jabber.org");
// Jabber 서버에 사용자로서 인증
connection.login("mtucker", "password");
// 지정한 사용자에게 메시지 전송
connection.createChat("jsmith@jivesoftware.com").sendMessage("Howdy!");
```

(그림 8) Smack 라이브러리의 사용

```
// Jabber 서버에 사용자로서 인증
<query xmlns="jabber:iq:auth">
  <username>mtucker</username>
  <password>password</password>
</query>

// 지정한 사용자에게 메시지 전송
<message to="jsmith@jivesoftware.com" id="message22">
  <body>Howdy!</body>
</message>
```

(그림 9) Smack 라이브러리를 통한 Jabber 프로토콜 메시지의 내용

지와 상태 표시(Presence)기능을 제공하는 XMPP 기반의 오픈 소스 Jabber 클라이언트 라이브러리이다. Smack 라이브러리는 자바로 구현되어 있으며 XML 포맷으로 데이터를 주고받는다. Smack 라이브러리는 Jabber 서버에 접근하는 방법과 접근하고 나서 Jabber 프로토콜에 맞는 형식으로 데이터를 주고받을 수 있는 방법을 제공한다. (그림 8)은 Smack 라이브러리를 통하여 간단하게 Jabber 프로토콜을 사용하는 방법을 보여준다.

(그림 9)는 (그림 8)과 연관되어 구성된 Jabber 프로토콜 메시지의 내용을 보여준다.

3.4 협업시스템을 위한 Jabber 프로토콜의 확장

기존의 Jabber 프로토콜은 한 사용자가 한 그룹에만 소속될 수 있기 때문에 다수의 사용자가 여러 그룹에 소속되어 작업을 하게 되는 협업시스템의 정보를 표현하기는 어렵다. 이러한 문제점을 해결하고 협업시스템에 맞는 메신저 정보를 표현하기 위해서는 Jabber 프로토콜의 XML 스키마를 확장해야 한다.

확장된 XML 스키마는 여러 그룹에 속할 수 있는 사용자를 표현하기 위해 각 그룹별로 사용자를 복수로 표현한다.

```
<query xmlns="jabber:iq:CoSlide_roster" xdbns="jabber:iq:CoSlide_roster">
  <item jid="jinop@203.250.77.124" name="jinop" subscription="to">
    <group>my friends</group>
    <type>usergroup</type>
  </item>
  <item jid="heinz@203.250.77.124" name="heinz" subscription="to">
    <group>my friends</group>
    <type>usergroup</type>
  </item>
  <item jid="jinop@203.250.77.124" name="jinop" subscription="from">
    <group>CoSlide</group>
    <type>workgroup</type>
  </item>
</query>
```

(그림 10) 작업팀 정보의 표현을 위한 XML 문서의 예

그리고 <item> 태그 하위에 <type> 태그를 두어 사용자 그룹과 협업 작업을 하는 그룹의 표현한다. (그림 10)은 협업시스템의 정보를 표현하기 위한 XML 문서의 예이다.

(그림 10)의 XML 문서에서는 협업시스템의 정보를 표현하기 위해 jabber:iq:CoSlide_roster라는 XML 네임스페이스를 추가적으로 사용하였다. 새로운 네임스페이스에서 사용자의 그룹을 분류하기 위한 <type>를 추가하여 Jabber 사용자와 협업시스템 사용자를 구별한다. <type> 태그의 값이 'usergroup'이면 일반 Jabber 사용자를 나타내고 <type> 태그의 값이 'workgroup'이면 협업시스템에 참여하고 있는 사용자를 나타낸다.

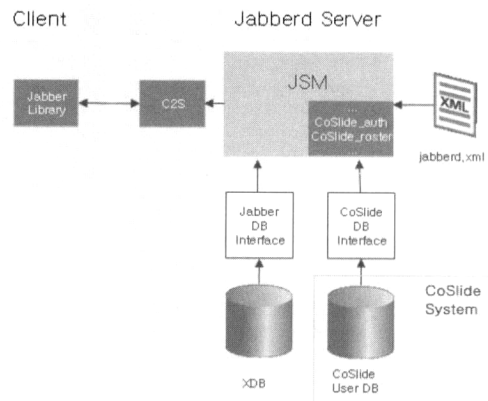
확장된 XML 스키마를 통하여 서버는 한 사용자가 여러 그룹에 소속되는 협업시스템의 정보를 표현할 수 있는 확장된 Jabber 프로토콜을 제공할 수 있다. 따라서 클라이언트는 이 프로토콜을 통하여 협업시스템에 소속된 사용자들 간에 원활한 통신을 지원할 수 있다.

4. 협업시스템을 위한 CoJBother 메신저 클라이언트의 구현

협업시스템을 지원하는 효과적인 메시징 시스템을 개발하기 위해서는 협업시스템 정보를 제공하는 서버와 그 정보를 바탕으로 원활한 통신기능을 제공하는 메신저 클라이언트가 필요하다. 메신저 서버는 협업시스템의 정보저장소에 접근하여 확장된 Jabber 프로토콜을 이용하여 제공하여 준다. 그리고 메신저 클라이언트는 다양한 기능과 인터페이스를 통하여 협업시스템 정보를 사용자에게 제공하여준다.

4.1 CoJBother

CoJBother는 Jabber 메신저인 JBother를 확장한 시스템으로서 표준 Jabber 프로토콜을 지원하여 일반 Jabber 사용자들 간에 통신을 지원할 뿐만 아니라 협업시스템에 맞게 확장된 Jabber 프로토콜을 지원하여 협업시스템에 참여하는 사용자들 간의 통신도 지원한다. (그림 11)은 표준



(그림 11) 협업시스템과 연동한 Jabber Server의 구조

Jabber Server에 협업시스템의 정보를 연동하는 과정을 보여준다.

4.2 협업시스템을 위한 Jabber Server의 확장.

CoJBother와 동작할 서버는 표준 Jabber 서버인 Jabberd를 CoSlide 협업시스템에 맞도록 확장하여 구현된다. Jabberd는 모듈구조로 되어있어 서비스의 확장 및 추가가 용이하다.

본 연구에서는 Jabberd에 CoSlide_auth와 CoSlide_roster 모듈을 추가하였다. CoSlide_auth는 협업시스템의 인증을 위한 컴포넌트이고, CoSlide_roster 모듈은 CoSlide 협업시스템의 사용자와 그룹의 정보를 클라이언트에게 전송하기 위한 컴포넌트이다. 확장된 Jabberd 서버는 이 두 가지 모듈을 통하여 CoSlide 협업시스템과 연동한다. 추가된 모듈은 JSM(Jabber Session Manager)의 하부 컴포넌트로 구현되는데, JSM의 하부 컴포넌트들은 Jabberd의 설정 파일인 jabberd.xml 내의 <load main="jsm"/> 엔트리에 등록된다.

CoSlide 협업시스템과 연동된 Jabberd 서버는 일반 Jabber 사용자와 협업시스템 사용자를 구별하여 인증을 한다. Jabberd는 일반 Jabber 사용자로 인증을 받으면 xdb로부터 정보를 가져와서 클라이언트로 응답을 보내게 된다. 일반 사용자가 아닌 협업시스템 사용자로서 인증을 받으면 xdb뿐만 아니라 CoSlide의 사용자와 그룹 정보를 가져오기 위한 모듈인 CoSlide_auth를 통하여 협업시스템의 사용자 정보 저장소에 접근을 하게 된다. (그림 12)는 users.c 파일의 협업시스템 사용자로 인증을 받는 부분의 코드를 보여준다.

사용자가 CoSlide 협업시스템의 사용자로서 인증을 받고 난 후에 서버는 그 사용자가 속해 있는 그룹의 정보를 클라이언트로 전송한다. 협업시스템 사용자와 그룹에 대한 정보는 각각 XML문서로 정의되어 저장된다. 저장되어 있는 협업시스템의 정보들을 클라이언트로 전송하기 위하여 추가된 CoSlide_Roster 컴포넌트를 사용하며, 협업시스템의 저장소에 접근하여 정보를 얻고 난 뒤에 확장된 Jabber 프로토콜을

```
// CoSlide 협업시스템의 사용자 패스워드를 가져오는 메소드
char *get_iplace_pass(char *userid){
    ...
    char *name = xmlnode_get_attrib(cur, "name");
    // 입력된 사용자의 패스워드가 존재하면 패스워드를 반환
    if(strcmp(name, "password") == 0){
        value = xmlnode_get_attrib(cur, "value");
        return conv_encoding("EUC KR", "UTF 8", value);
    }
    ...
}

// Jabberd에서 사용자 인증을 처리하는 메소드
udata js_user(jsmi si, jid id, HASHTABLE ht){
    ...
    newu->pass = pstrdup(p, get_iplace_pass(newu->user));
    newu->id = jid_new(p, jid_full(uid));
    return newu;
}
```

(그림 12) Jabberd에서 CoSlide 협업시스템 사용자 인증을 처리하는 코드

```
// CoSlide 협업시스템의 정보를 가져오는 메소드
xmlnode coslide_roster_get(char *user)
{
    ...
    DIR *dp;
    xmlnode ret;
    if((dp = opendir("/usr/local/tomcat-src/store/metadata/users")) == NULL.)
        return ret;
    ret = xmlnode_new_tag("query");
    xmlnode_put_attrib(ret, "xmlns", "NS_COSLIDE_ROSTER");
    while((dir_info = readdir(dp)) != NULL){
        ...
        strcat(dirtemp, dir_info->d_name);
        xmlnode temp = xmlnode_file(dirtemp);
        char *name = xmlnode_get_attrib(temp, "name");
        xmlnode tmp_item = xmlnode_new_tag("item");
        xmlnode_put_attrib(tmp_item, "jid", strtemp);
        ...
        xmlnode_insert_tag_node(ret, tmp_item);
    }
    ...
}
```

(그림 13) 협업시스템 저장소로부터 사용자 정보를 가져오는 코드 부분

통해서 클라이언트로 전송한다.

CoSlide_Roster는 기존 Jabberd 서버의 Roster 컴포넌트를 확장한 컴포넌트로 클라이언트가 서버에 접속을 하면 협업시스템의 정보를 얻어서 확장된 Jabber 프로토콜을 통하여 클라이언트로 보내도록 한다.

C로 구현된 Jabberd에서 XML 문서로부터 정보를 얻기 위해 XMLParser 라이브러리를 사용하였다. (그림 13)은 CoSlide_Roster 컴포넌트에서 협업시스템의 저장소에 접근해 정보를 가져오는 코드 부분이다.

4.3 협업시스템을 위한 Jabber Client의 확장

‘Jabber 메신저’는 Jabber 프로토콜을 이용하는 메시징 시스템을 가리키는 것으로 본 연구에서는 협업시스템을 지원하도록 확장된 Jabber 프로토콜을 지원할 수 있는 Jabber 메신저를 개발한다. 이 메신저를 개발하기 위하여 표준 Jabber 프로토콜을 지원하는 JBother 메신저를 확장하도록 한다. JBother 메신저는 표준 Jabber 프로토콜을 지원할 뿐만 아니라 화려한 인터페이스와 그룹채팅 등의 다양한 기능을 지원한다.

하지만 JBother는 협업시스템의 작업그룹정보를 자동적으로 표현하지 못하기 때문에 협업시스템과 능동적으로 연동되려면 확장을 하여야 한다. JBother를 확장하여 개발된 CoJBother는 확장된 Jabber 프로토콜을 지원하여 협업시스템에서 작업 중인 그룹 참여자정보를 자동적으로 표현하고 효과적인 통신을 위한 작업그룹별 그룹채팅기능을 제공한다.

CoJBother는 표준 Jabber 프로토콜을 지원하는 JBother 메신저를 확장한 클라이언트서 확장된 Jabber 프로토콜을 지원하고 그에 맞는 인터페이스와 기능을 제공한다.

CoJBother가 협업시스템을 지원하기 위하여 기존의 JBother 클라이언트에서 수정된 부분은 크게 두 가지이다. 한 부분은 확장된 Jabber 서버와의 통신을 위하여 확장된 Jabber 프로토콜을 사용할 수 있도록 Smack 라이브러리를

수정하는 것이고 또 다른 한 부분은 서버로부터 받은 협업 시스템의 정보를 사용자에게 맞추도록 제공하는 인터페이스를 구현하는 것이다.

4.3.1 확장된 Jabber 프로토콜을 위한 Smack 라이브러리 수정

클라이언트는 Jabber 서버로 접속할 때 협업시스템 사용자로서 접속을 하기 위하여 Smack 라이브러리에서 'jabber:iq:CoSlide_auth' 네임스페이스를 사용하여 인증에 관련된 XML 문서를 전송하고 서버로부터 인증을 받는다. 그리고 서버로부터 받은 XML 문서에서 협업시스템의 사용자의 이웃정보를 구별해내기 위하여 'jabber:iq:CoSlide_

```
// 추가된 네임스페이스를 이용하여 쿼리문을 전송
public String getChildElementXML(){
    StringBuffer buf = new StringBuffer();
    buf.append("<query xmlns='jabber:iq:coslide_auth'>");
    ...
}
// 서버로부터 받은 쿼리문에서 협업시스템 관련 이웃정보를 분류
private IQ parseIQ(XmlPullParser parser) throws Exception{
    ...
    String name = parser.getName();
    String namespace = parser.getNamespace();
    ...
    else if(name.equals("query") && namespace.equals("jabber:iq:coslide_roster"))
        iqPacket = parseCoSlideRoster(parser);
    ...
}
// 서버로부터 받은 Jabber 프로토콜 패킷으로부터 사용자의 정보를 추출
private RosterPacket parseCoSlideRoster(XmlPullParser parser) throws Exception{
    ...
    String jid = parser.getAttributeValue("", "jid");
    ...
}
```

(그림 14) Smack 라이브러리의 수정

```
// 협업시스템 사용자를 메신저 이웃 목록에 보여주도록 하는 코드
// Smack 라이브러리를 통해 협업시스템 사용자에게 대한 정보를 가져온 뒤 처리함
Set set = CoSlideGroup.getObject().getData();
Iterator i = set.iterator();
if(group.equals("CoSlide") || group.equals("Contacts"))
    while(i.hasNext()){
        Map.Entry e = (Map.Entry)i.next();
        String str = (String)e.getKey();
        StringTokenizer st = new StringTokenizer((String)connection.getUser(), "");
        String name = st.nextToken();
        Vector v = (Vector)e.getValue();
        String t = (String)buddy.getUser().trim();
        for(int j = 0; j < v.size(); j++){
            String s = (String)v.elementAt(j);
            if(s.equals(t))
                CoSlideAddGroup(str.buddy);
        }
}
// 자동적으로 그룹 채팅을 지원하기 위한 기능의 소스 코드
// 그룹 채팅 기능을 위한 메뉴를 추가하고 이벤트 등록
private JMenuItem groupchat = new JMenuItem("GroupChat");
add(groupchat);
groupchat.addActionListener(listener);
...
// 기능을 활성화 하면 자동으로 그룹 채팅을 하기 위한 정보를 담은 북마크 실행
GroupChatBookmarks gc =
    new GroupChatBookmarks(BuddyList.getInstance().getTabFrame());
gc.setCoSlideGroupChat(groupname, "conference.203.250.77.124");
```

(그림 15) 협업시스템 사용자 목록과 그룹 채팅 기능을 구현하는 소스 코드

roster' 네임스페이스를 사용한다. (그림 14)는 Smack 라이브러리에 협업시스템 사용자로 접근할 수 있도록 코드를 수정한 부분과 서버로부터 받은 XML 문서로부터 협업시스템 사용자 정보를 분류하도록 코드를 수정한 부분이다.

4.3.2 확장된 Jabber 프로토콜을 위한 클라이언트 인터페이스 수정

수정된 Smack 라이브러리를 통해서 협업시스템의 정보를 가져오면 그 정보를 표현하기 위해 기존의 JBother의 소스코드를 수정할 필요가 있다. CoJBother는 JBother의 인터페이스 부분을 수정해서 협업시스템 정보를 나타낸다. 또한 편리한 그룹 채팅을 지원하기 위하여 새로운 팝업 메뉴를 추가하고 사용자가 그룹 채팅을 요청하면 그룹 정보와 사용자 정보를 이용하여 자동으로 그룹 채팅방을 생성한다. (그림 15)는 협업시스템의 정보를 나타내는 인터페이스 코드 부분과 그룹 채팅을 지원하도록 수정한 코드 부분이다.

4.4 CoJBother 사용의 예

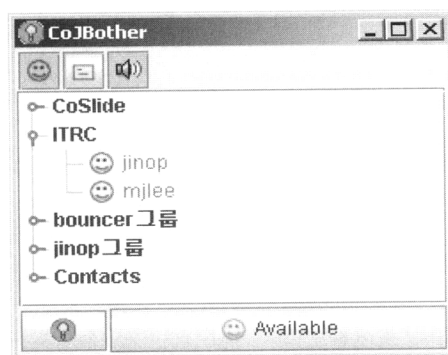
CoSlide 협업시스템은 XML 파일로 정의된 문서를 통하여 시스템 정보를 표현한다. 다음 (그림 16)은 CoSlide 협업시스템의 그룹 정보를 정의하는 XML 문서의 일부이다. 이 문서는 ITRC이라는 그룹의 속성을 정의하고 참여하는 사용자들의 정보를 표현한다.

다음 (그림 17)은 'mathpf' 사용자가 CoJBother를 통하여 CoSlide 협업시스템에 접근하였을 때 자신이 소속된 그룹과 그룹원의 정보를 보여주는 기본 인터페이스를 보여준다.

CoJBother는 협업시스템을 위한 편리한 그룹 채팅 기능을 제공한다. 사용자가 팝업 메뉴를 통한 그룹 채팅방 생성을 요청하면 CoJBother는 그룹의 정보와 사용자의 정보

```
// ITRC 그룹에 참여하고 있는 사용자들의 정의
<property name="Groupmemberlist" namespace="DAV:"
    value="jinop,mathpf,mjlee" type="" protected="false">
```

(그림 16) CoSlide 협업시스템의 그룹을 정의하는 XML 문서의 일부

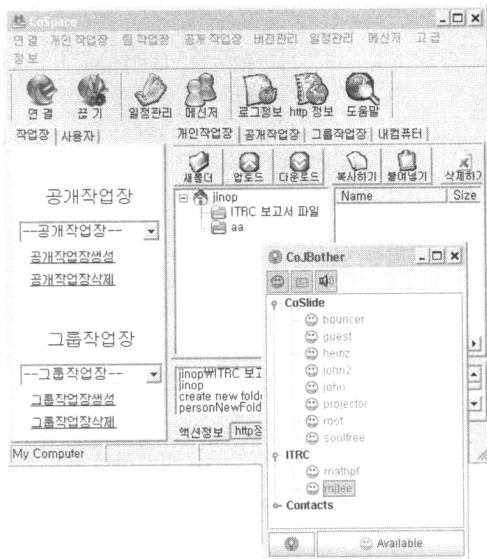


(그림 17) 협업시스템의 그룹 정보를 표시하는 CoJBother의 기본 인터페이스

를 사용하여 서버로 그룹 채팅방 개설을 요청한다. 그에 따라 Jabberd 서버는 conference 컴퍼넌트를 사용하여 그룹 채팅방을 개설하게 되어 사용자는 자동적으로 그룹 채팅방에 참여하게 된다.

4.4.1 CoJBother의 활용

다음은 CoJBother를 활용하여 협업을 수행하는 과정을 보여준다. 사용자는 원격지에 떨어진 상태로 CoSlide 협업 시스템을 통하여 협업에 참여할 수 있고 다른 참여자와 정보를 주고받기 위하여 CoJBother를 사용할 수 있다. 사용자는 CoJBother의 그룹채팅기능을 이용하여 손쉽게 협업 참여자들과 정보를 교환할 수 있다.



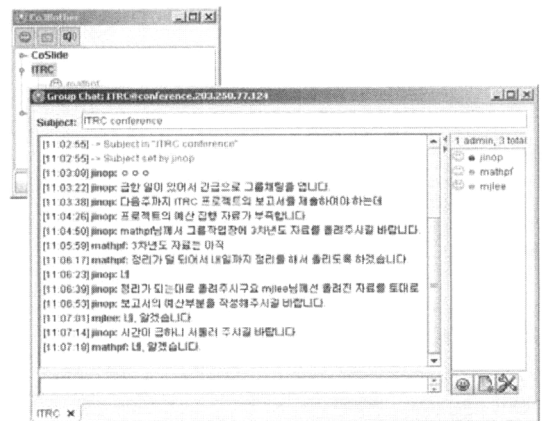
(그림 18) 협업시스템 클라이언트와 연동된 CoJBother를 사용하는 모습



(그림 19) CoJBother를 이용한 손쉬운 그룹채팅 기능

- 1) 연구소 프로젝트의 참여자인 'jinop'와 'mathpf'는 프로젝트를 수행하기 위하여 각각 미국과 프랑스로 출장을 가게 되었음.
- 2) 원격지에 떨어진 상태에서도 CoSlide 협업시스템의 클라이언트를 사용하여 출장 중 연구보고서 작성을 위한 자료를 그룹작업장에 올리는 등의 협업을 수행함.

- 3) 원격지에 떨어져있는 협업그룹 참여자들 간에 연구보고서 작성 분제로 긴급한 의사소통이 필요하게 되었음.
- 4) 이메일이나 전화와 같은 별도의 도구를 사용하지 않고 협업시스템 클라이언트에 연동된 CoJBother를 사용하여 협업그룹의 참여자들을 확인하고 원클릭으로 그룹채팅을 시작함.
- 5) 프로젝트를 진행하기 위한 회의를 시작하고 성공적으로 의사를 교환함.



(그림 20) 그룹 구성원들이 CoJBother를 통하여 통신하는 예

4.5 평가

본 연구에서 개발된 CoJBother 메신저의 성능을 평가하기 위하여 연동된 협업시스템과 다른 협업시스템을 비교하고 메신저의 연동에 따른 작업과정의 차이를 비교하였다.

4.5.1 협업시스템 비교

<표 3>은 자바 기술 기반으로 다양한 작업공간을 지원하는 iPlace[13], 대표적인 웹기반 협업시스템인 BSCW[14]와 함께 CoSlide 협업시스템의 차이점을 보여준다.

CoSlide 협업시스템은 기존의 협업시스템과는 달리 그룹작업장과 공개작업장을 완벽히 지원하고 CoJBother와 연동되어 협업을 수행하는 동시에 실시간 메시징이 가능하다.

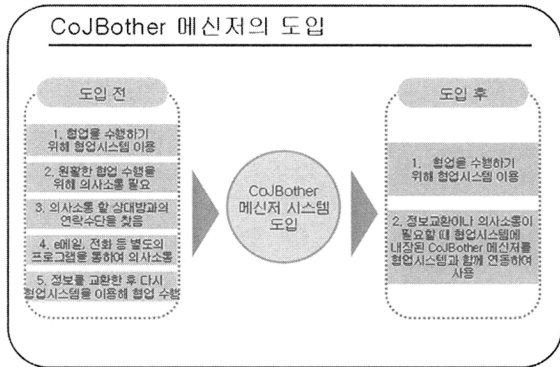
<표 3> 협업시스템 비교

	BSCW	iPlace	CoSlide
기반 프로토콜	http	http	webdav
개인작업장	○	○	○
그룹작업장	△	○	○
공개작업장	×	×	○
구현기술	Python	EJB+JSP	JAVA
사용플랫폼	Unix, Windows	Unix, Linux, Windows	Unix, Linux, Windows
그룹참여	수동적	능동적/수동적	능동적/수동적
자원처리프로그램 자동 구동	×	×	○
실시간 메시징	×	×	○

○:지원 △:부분지원 ×:미지원

4.5.2 메신저의 도입에 따른 작업 과정

다음 (그림 21)은 메신저가 없는 협업시스템과 CoJBother 메신저를 도입한 CoSlide 협업시스템에서 협업 참여자들이 의사소통을 할 때 각각의 작업과정을 보여준다.



(그림 21) 메신저의 도입 전후의 협업시스템 작업 과정

협업시스템을 이용해 협업을 수행할 때 보다 효과적인 협업을 위하여 참여자들 간에 의사소통이 필요하다. 이러한 의사소통의 도구로서 E-메일이나 전화가 오랫동안 사용되어 왔다. 하지만 이러한 방법은 불필요한 작업을 수반하여 정작 협업의 효율성을 떨어뜨린다. 본 연구에서 개발된 CoJBother 메신저를 도입하면 참여자들 간에 의사소통을 위해 별도의 도구를 사용하는 불필요한 작업이 제거된다. CoJBother는 협업시스템과 긴밀히 연동되어 협업에 참여하는 사람들과 자동적으로 연결이 되어 편리하게 의사소통할 수 있도록 한다. 특히 따라서 협업 참여자들은 협업에 필요한 의사소통을 위하여 별도의 도구를 사용하지 않고 협업시스템과 연동되는 메신저를 사용하여 보다 효과적인 협업을 수행할 수 있다.

5. 결론 및 향후 연구

본 논문에서는 여러 사용자가 CoSlide 협업시스템을 이용하여 협업을 수행할 때 그룹 구성원들 간의 원활한 통신을 지원하는 메신저인 CoJBother의 개발에 대하여 기술하였다. CoJBother는 메신저를 위한 표준 프로토콜 중의 하나인 Jabber 프로토콜을 사용하는 JBother 메신저를 확장하여 구현되었으며 기존의 Jabber 사용자와 그룹 간의 통신과 더불어 CoSlide 협업시스템의 사용자와 그룹 간의 통신을 또한 지원하도록 구현되었다.

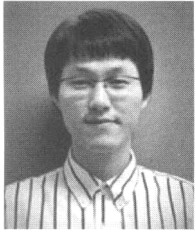
CoSlide 협업시스템 환경의 정보를 제공하기 위하여 Jabber 프로토콜을 확장하고 메시징 시스템을 위한 Jabber 서버인 Jabberd를 기반으로 메시징 시스템 서버를 개발하였다. CoJBother는 확장된 Jabberd 서버와의 효과적인 통신을 위하여 협업시스템 정보를 표현할 수 있도록 수정된 Smack 라이브러리를 사용하였으며 협업시스템 환경에 맞는 인터페이스를 제공한다. CoJBother는 CoSlide 협업시스

템에 부합하는 기능을 제공함으로써 협업시스템의 사용자와 그룹 간의 효과적인 통신을 지원한다.

현재의 CoSlide 협업시스템 사용자는 협업을 수행하기 위하여 협업 클라이언트와 더불어 다른 사용자이나 그룹과 통신을 하기 위하여 CoJBother를 별도로 사용하여야 한다. 두 개의 프로그램을 따로 사용하여 통신을 하고 협업을 수행하는 것이 효율성을 저해하는 요인이 될 수 있으므로 향후 연구는 협업시스템 클라이언트와 CoJBother 메신저를 융합하여 보다 효과적인 협업 환경을 제공하고자 한다.

참고 문헌

- [1] "http://jakarta.apache.org/slide/", Jakarta Slide.
- [2] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, "HTTP Extensions for Distributed Authoring - WEBDAV," RFC 2518, Standards Track, February, 1999.
- [3] E. James Whitehead, Jr., Meredith Wiggins, "WEBDAV: IETF Standard for Collaborative Authoring on the Web," IEEE Internet Computing, pp. 34-40, September/October 1998.
- [4] G. Clemm, E. Sedlar, J. Whitehead, "Web Distributed Authoring and Versioning.
- [5] 김동호, 박진호, 신원준, 이명준, "웹데브 기반의 효과적인 협업 작업 지원" 한국정보과학회, 한국컴퓨터종합학술대회 2006.
- [6] 박진호, 신원준, 김동호, 이명준, "WebDAV 기반의 리눅스 협업시스템 클라이언트", 한국정보과학회, 한국컴퓨터종합학술대회 2006.
- [7] "http://www.jbother.org", JBother.
- [8] "http://www.jabber.org", Jabber Software Foundation.
- [9] "http://jivesoftware.org/smack", Smack.
- [10] "http://jabberd.org", Jabberd.
- [11] 이근웅, 안건태, 김진홍, 문남두, 이명준, "효과적인 협업지원을 위한 Jabber 메신저 시스템의 확장" 한국정보과학회, 한국컴퓨터종합학술대회 2002.
- [12] D.J. Adams, "Programming Jabber", O'Reilly, 2002.
- [13] 안건태, 정명희, 이근웅, 문남두, 이명준, "iPlace: EJB 기술을 이용한 웹 기반 협업시스템", 한국정보처리학회논문지, 제8-D권, pp. 735-746, 2001년 12월.
- [14] Appelt, W., Mambrey, P. "Experiences with the BSCW Shared Workspace System as the Backbone of a Virtual Learning Environment for Students," Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 99, Seattle, June 1999.



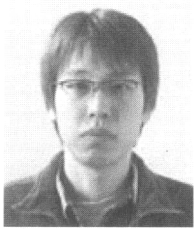
이 흥 창

e-mail : myhyunii@mail.ulsan.ac.kr
2006년 울산대학교 컴퓨터정보통신공학부
졸업(공학사)
2007년~현 재 울산대학교
컴퓨터정보통신공학부 석사 과정
관심분야: 웹기반 정보시스템, 협업시스템,
분산 프로그래밍



김 성 훈

e-mail : heinz@mail.ulsan.ac.kr
2007년 울산대학교 컴퓨터정보통신
공학부 졸업(공학사)
2007년~현 재 울산대학교
컴퓨터정보통신공학부 석사 과정
관심분야: 분산시스템, 센서네트워크
프로그래밍, 웹기반 정보시스템



박 진 호

e-mail : jinop@mail.ulsan.ac.kr
2006년 울산대학교 컴퓨터정보통신
공학부 졸업(공학사)
2007년~현 재 울산대학교
컴퓨터정보통신공학부 석사 과정
관심분야: 분산시스템, 협업시스템,
WebDAV, 웹기반 정보시스템



이 명 준

e-mail : mjlee@ulsan.ac.kr
1980년 서울대학교 수학과 졸업(학사)
1982년 한국과학기술원 전산학과 졸업
(석사)
1991년 한국과학기술원 전산학과 졸업
(박사)
1982년~현 재 울산대학교 컴퓨터정보통신 공학부(교수)
1993년~1994년 미국 버지니아대학 교환교수
2005년~2006년 미국 캘리포니아주립대학 교환교수
관심분야: 웹기반 정보시스템, 프로그래밍언어, 분산 프로그래밍,
생물정보학, 센서네트워크 프로그래밍환경