

PECOLE+ : 다중그룹을 효과적으로 지원하기 위한 PECOLE 협업 시스템의 확장

김보현*, 박종문**, 이명준***, 박양수****

PECOLE+ : An Extension of PECOLE Collaborative System for Supporting Effective Multiple Groups

Bo-Hyeon Kim*, Jong-Moon Park**, Myung-Joon Lee***, Yang-Soo Park****

요 약

PECOLE은 다중채팅, 화상회의, 화면 공유 등의 다양한 협업 어플리케이션으로 구성된 협업공간을 지원하는 Peer-to-Peer 기반의 멀티미디어 분산 협업 환경이다. 하지만 PECOLE에서 제공하는 단순한 그룹 관리 기능을 통하여서는 다중 그룹에 참여하여 협업을 수행하는 것이 불가능하다. 본 논문에서는 PECOLE을 확장한 PECOLE+의 설계와 구현에 대하여 설명한다. PECOLE+는 그룹 관리 서비스와 작업장 관리 서비스를 제공하여 PECOLE의 문제점을 해결한다. 그룹 관리 서비스는 그룹 생성, 다중 그룹 참여, 그룹 검색 등의 기능을 제공하며, 작업장 관리 서비스는 그룹별로 작업 공간을 제공하여, 그룹 단위로 주어진 작업 공간에서 어플리케이션을 이용한 협업의 수행을 지원한다. 또한 제공된 플러그인 인터페이스에 부합하게 개발된 모든 협업 어플리케이션은 PECOLE+의 협업 어플리케이션으로서 작업 공간에서 실행되도록 지원한다.

▶ Keyword : 분산 협업 환경, 피어 투 피어, 그룹 관리 서비스, 작업장 관리 서비스, 적스타

Abstract

PECOLE (Peer-to-Peer Collaborative Environment) is a P2P-based multimedia distributed collaborative environment supporting a collaborative workspace which is composed of a variety of collaborative applications such as multi-chat, video conferencing, screen sharing and etc. Unfortunately, due to the PECOLE's simple group management, it is impossible to perform collaboration activities while joining multiple groups. In this paper, we present the design and implementation of PECOLE+ which is an extension of PECOLE. PECOLE+ resolves

• 제1저자 : 김보현 교신저자 : 박양수

• 투고일 : 2010. 08. 26, 심사일 : 2010. 10. 01, 게재확정일 : 2010. 10. 13.

* 울산대학교 컴퓨터정보통신공학부(School of Computer Engineering & information Technology, Ulsan University) 석사과정

** 울산대학교 컴퓨터정보통신공학부(School of Computer Engineering & information Technology, Ulsan University) 석사

*** 울산대학교 컴퓨터정보통신공학부(School of Computer Engineering & information Technology, Ulsan University) 교수

**** 울산대학교 컴퓨터정보통신공학부(School of Computer Engineering & information Technology, Ulsan University) 교수

※ 이 논문은 2010년 울산대학교 연구비에 의하여 연구되었음

the drawback of PECOLE by providing the Group Management Service and the Workspace Management Service. The Group Management Service provides functionalities such as creating groups, joining multiple groups, and searching groups, and etc. The Workspace Management Service provides each group with an associated workspace, supporting the execution of collaborative applications over the workspace. In addition, any collaborative applications with the provided plug-in interfaces can be executed over the workspace as a PECOLE+ collaborative application.

▶ Keyword : Distributed Collaborative Environment, Peer-to-Peer, Group Management Service, Workspace Management Service, JXTA

1. 서론

오늘날의 협업은 이전의 근거리에서 제한된 협업 환경에서 혁신적인 IT 기술을 접목하여 물리적인 시공간의 제약을 초월하고, 다양하고 긴밀한 협업 환경을 제공하여 복잡한 협업을 효과적으로 지원하고 있다.

협업을 제공하는 응용 시스템으로 가장 큰 비중을 차지하는 것은 웹 기반 협업 시스템으로써 온라인 협업을 지원하고 있다. BSCW[1], CoSlide[2], PhpGroupware[3], TikiWiki[4], IPlace[5]등과 같은 웹 기반 협업 시스템은 클라이언트/서버 모델로 동작한다. 이러한 구조는 협업과 관련된 비즈니스 모델을 중앙의 서버에 집중시키기 때문에 시스템의 관리와 구성이 용이하며 표준 웹 브라우저에서 동작하도록 구현함으로써 클라이언트에 대한 부담을 감소하는 장점이 있지만 클라이언트의 수가 증가할수록, 서버에 대한 부하와 대역폭 요청이 증가하기 때문에 서버에 대한 부담이 커지므로 전체적인 효율은 떨어지게 된다. 결국 클라이언트/서버 모델이 가지는 많은 장점에도 불구하고 single-point-of-failure 라는 한계는 복잡해지는 협업 환경을 효과적으로 지원하기 어렵다.

P2P 컴퓨팅[6]은 사용자들 사이에서 네트워크를 통한 파일 공유와 커뮤니케이션을 위한 분산 컴퓨팅 어플리케이션으로서 피어로 대변되는 네트워크 구성원이 서버와 클라이언트의 역할을 동시에 수행하며 인터넷과 같은 기존의 네트워크를 이용하여 수평적인 형태로 연결되고, 자율적이고 동적인 방법으로 네트워크 토폴로지를 형성하여 서로 상호 동작한다. 이러한 P2P의 특징을 협업 환경에 적용할 경우 기존의 클라이언트/서버 기반의 협업 활동을 기반으로 제공되는 협업 서비스를 P2P 네트워크에 존재하는 여러 피어에게 분산시키므로 특정 서버의 부하, 네트워크 혼잡에 대한 문제를 해결한다. 또한 P2P 컴퓨팅의 특징은 실제 인간 사회의 협업 구조와 매우 유사하며 이러한 특징은 온라인 협업에도 적용할 경우 능동적이고 효과적인 협업 환경을 구성 할 수 있는 장점이 있다. 현재

TOMSCOP[7], PECOLE[8], PHAC[9], JCLS[10]와 같은 P2P 기반의 분산 협업 환경의 연구가 활발히 진행되고 있다.

특히 멀티미디어 분산 협업 환경으로 개발된 PECOLE (Peer-to-Peer multimedia collaborative environment)은 JXTA[11] 프로토콜을 기반으로 다중채팅, 화상회의, 화면공유, 포인트 공유, 다중 언어 지원 등의 다양한 협업 어플리케이션을 제공한다. JXTA 프레임워크를 기반으로 P2P 오버레이 네트워크를 형성하고 그 위에서 다양한 협업 어플리케이션을 제공하며 피어 그룹의 웹 문서 공유뿐만 아니라 협업 사용자의 커서 움직임이나 새로운 URL링크로의 이동 같은 그룹 사용자 인식 정보를 이용해서 웹 공유 문서를 조작하는 기능을 지원한다. 특히 멀티미디어를 활용한 협업에 중점을 두고 있으며 음성이나 화상 스트림 같은 멀티미디어 자료를 전송하기 위해서 JXTA MulticastSocket 기술을 이용하고 있다. 하지만 PECOLE은 공동 작업을 수행하는 사용자들의 그룹을 세션으로 정의하여 하나의 세션에만 참가할 수 있도록 하는 단순한 그룹 관리 기능을 제공하고 있어 실제 협업 과정에서 빈번히 필요하여지는 다중 그룹의 운용이 불가능하고, 기본적으로 제공하는 협업 어플리케이션 외에도 사용자의 요구가 반영된 다양한 협업 어플리케이션을 사용하기 어렵다.

본 논문에서는 P2P기반의 분산 협업 시스템인 PECOLE+에 대하여 기술한다. PECOLE+는 개방형 P2P 프로토콜인 JXTA를 이용하여 다중 그룹 관리를 지원하는 그룹 관리 서비스(Group Management Service)와 그룹을 기반으로 동작하는 협업 어플리케이션의 효과적인 동작을 지원하는 작업장 관리 서비스 (Workspace Management Service)를 제공한다. 그룹 관리 서비스는 참여 그룹 변경, 복수 그룹 참여, 그룹 탐색, 협업 사용자 초대 등 다중 그룹 관리 기능을 제공하고, 작업장 관리 서비스는 협업을 위한 어플리케이션들을 사용자간에 상호 동작하기 위한 작업 공간(Workspace)을 제공하는 작업장 서비스(Workspace Service)와 다양한 사용자의 요구를 반영하여 개발된 협업 어플리케이션을 PECOLE에서 쉽게 동작할 수 있도록 플러그인 서비스(Plug-in Service)를 제공한다. 개발된 PECOLE+는 실제 협업 환경에서 벌어지는 복잡하고 다양한 상황에 대처할 수 있으며,

효과적인 협업을 위한 도구로 이용될 수 있다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 협업 환경과 P2P 컴퓨팅에 관련된 기존 연구를 살펴보고 3장에서는 다중 그룹의 지원을 위한 그룹 관리 서비스에 대해서 설명하고 4장에서는 그룹 기반으로 실행되는 협업 어플리케이션을 효과적으로 관리할 수 있는 작업장 관리 서비스에 대해서 설명한다. 5장에서는 PECOLE+의 확장된 기능과 성능을 기존의 PECOLE과 비교하고 마지막으로 6장에서 본 연구에 대한 결론과 추후 연구방향에 대하여 기술한다.

II. 관련 연구

1. JXTA

JXTA는 Sun Microsystems에 의해서 개발된 개방형 P2P 프로토콜로 Structured /Unstructured 의 중간 형태의 구조를 가지며 P2P 응용에 대한 상호운용성과 플랫폼/언어 독립성, 그리고 어떠한 장치에도 존재할 수 있는 편재성을 제공한다. 네트워크 가능한 어떠한 장치- 휴대폰, PDA, PC, NetBook 등-에서도 피어 상호간의 통신과 협업이 가능하기 때문에 많은 연구에서 다루어지고 있으며 현재의 TCP/IP 기반 네트워크의 한계를 넘어 인터넷의 범위를 확대하려는 의도에서 설계되었다. JXTA는 P2P 솔루션을 개발하기 위한 공통의 핵심 기능을 제공하여 개발자가 P2P 어플리케이션 개발에 집중할 수 있도록 도와준다. 또한, 네트워크 하부기술에 독립적으로 TCP/IP, UDP/IP, bluetooth 그리고 다른 유무선 기술에서도 작동할 수 있으며 특히, NAT(Network Address Translation), DHCP, firewall등의 제한적인 네트워크에서도 통신할 수 있다. [그림 1]은 물리 네트워크계층위에 형성된 JXTA 가상 네트워크인 오버레이 네트워크를 보여준다.

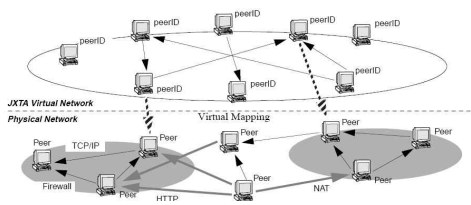


그림 1. JXTA 오버레이 네트워크
Fig 1. JXTA Overlay Network

JXTA는 [그림 2]와 같이 3개의 논리적 계층으로 나뉘며 각 계층은 하위 계층의 기능위에 구축된다. JXTA 코어 계층은 피

어, 피어 그룹, 네트워크 전송(파이프, 엔드포인트, 메시지), 광고, 식별자, 프로토콜(탐색, 통신, 모니터링), 보안과 인증과 같이 P2P 솔루션에 필수적인 구성 요소를 제공한다. JXTA 서비스 계층은 피어 자원 탐색, 피어 문서 공유, 피어 인증 같은 다양한 P2P 응용에서 요구하는 기능을 제공한다. JXTA 응용 계층은 서비스 계층의 기능 위에서 동작하는 서비스로 구성되며 인스턴트 메시징과 같은 P2P 응용을 제공한다.

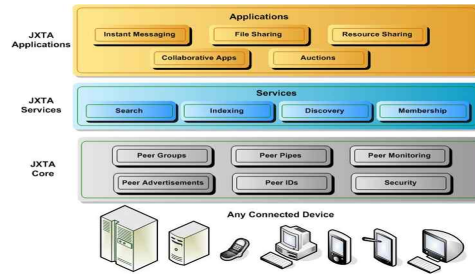


그림 2. JXTA 3계층 구조
Fig 2. JXTA 3-tier Architecture

1.1 JXTA PeerGroup

JXTA의 피어 그룹은 공통 관심사를 가지는 피어들의 동적인 집합으로 고유한 ID를 가지는 가상의 지역 공간을 제공한다. 피어 그룹은 피어 그룹 광고(PGA, Peer Group Advertisement)를 통해서 발행되며 네트워크로 전파된다. 이 피어 그룹 광고를 이용하여 피어의 어플리케이션 요구나 피어간의 상호작용을 위해서 동적으로 피어그룹을 생성한다. [그림 3]은 가상 네트워크에서 형성된 피어그룹의 한 부분을 보여주는데 피어 그룹은 동시에 다수의 그룹에 참여할 수 있으며 자유롭게 피어 그룹을 형성하고 참여할 수 있다. 또한, 피어그룹은 네트워크를 통한 검색이나 발견 요청의 전파(propagation)제한을 위해서 명확한 유효범위(scoping)메커니즘을 제공한다. JXTA 프레임워크는 discovery, resolver, pipe, peer info, rendezvous 같은 기본적인 피어그룹 서비스를 제공하며 피어그룹의 성격에 따라 사용자 정의 서비스를 제공할 수 있다.

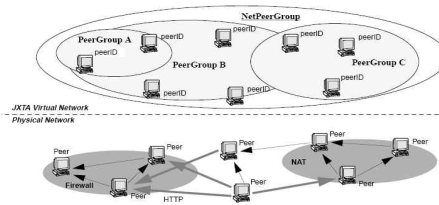


그림 3. 피어그룹 집합
Fig 3. PeerGroup in Virtual Network

2. 협업시스템

가상의 공동 작업공간을 지원하는 시스템의 필요성이 증가함에 따라 공동작업의 편의를 제공하기 위한 다양한 연구가 진행되었다. 이를 지원하는 시스템들은 일반적으로 협업의 수준에 따라 커뮤니케이션 도구, 회의 도구, 협업 관리 도구와 같이 세 개의 범주로 나뉜다. 전자 커뮤니케이션 도구는 메시지나 파일, 데이터, 혹은 문서를 사람들 사이에 주고받을 수 있게 만들어 정보의 공유를 증진시키며 전자메일, 팩스, 음성 메일, 웹 등이 있다. 전자 회의 도구는 좀 더 상호 대화적인 방식으로 정보의 공유를 증진시키는데 네트워크로 연결된 PC 상에서의 화이트 보드를 통한 데이터 회의, 전화를 활용한 음성/화상 회의, 대화방, 전자 회의 시스템 등이 있다. 협업 관리 도구들은 그룹 내의 활동을 관리하고 증진시키는데 사용되며 일정관리 소프트웨어로 전자 캘린더, 프로젝트 관리 시스템, 업무흐름관리 시스템, 지식 관리 시스템 등이 있다. 대표적인 협업지원 시스템으로는 GMD의 BSCW, 캘거리 대학의 TeamRooms[12], 울산대학교에서 개발한 iPlace와 CoSlide 협업시스템, PhpGroupware, 그리고 TikiWiki와 같이 온라인상에서 장기간의 협업을 지원하는 다양한 어플리케이션이 개발되었다.

3. PECOLE

멀티미디어 분산 협업 환경으로 개발된 PECOLE은 오픈 P2P 프레임워크인 JXTA를 기반으로 다자간 채팅, 화상회의, 화면 공유 등의 다양한 협업 어플리케이션을 제공하며 XML 메시지 기반의 통신 메커니즘을 이용하며 세션이라고 부르는 그룹 단위의 작업공간을 이용하여 실시간 멀티미디어 협업을 수행한다. [그림 4]는 PECOLE의 논리적인 계층을 보여주며 CA (collaborative application), WM (workspace manager), SM (session manager), CM (communication manager)계층으로 구성되어 있다.

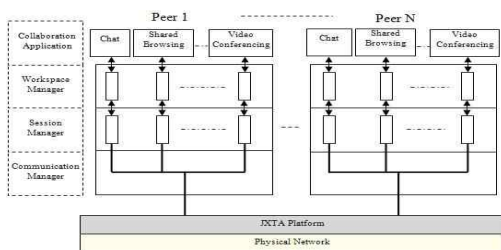


그림 4. PECOLE의 구조
Fig 4. Architecture of PECOLE

CA 계층은 PECOLE의 협업 어플리케이션과 어플리케이션의 프로토콜을 관리한다. WM 계층은 텍스트, 그래프, 이미지 같은 협업을 수행하는데 이용되는 데이터의 집합을 관리하고 사용자 데이터와 행위에 관련된 다양한 이벤트를 동기화한다. SM 계층은 협업을 위한 작업공간(workspace)인 세션의 생성, 관리, 소멸 등의 생명주기를 관리한다. CM 계층은 JXTA MulticastSocket을 이용하여 협업 사용자들의 메시지 전송 서비스를 제공한다.

[표 1] 는 다양한 협업 어플리케이션을 제공하는 PECOLE의 주요 기능을 나타낸다.

표 1. PECOLE의 주요 기능
Table 1. Functions of PECOLE

<ul style="list-style-type: none"> - JXTA 오버레이 네트워크에 접속 - 동일한 도메인에 속하는 피어와 그룹 발견 - 그룹 생성, 참여, 탈퇴 - 웹 브라우징 협업 (Collaborative web browsing) - 그룹 채팅 (Collaborative group chatting) - 화상/음성 회의 (Collaborative audio/video conferencing) - 다중언어를 지원하는 협업 (Multilingual collaboration) - Moderation for floor control - Shared telepointer demonstration
--

PECOLE은 음성과 화상 데이터를 효과적으로 처리하기 위해 JXTA MulticastSocket을 이용하여 각각의 음성 스트림과 화상 스트림 채널을 이용한다.[그림 5]는 5명의 사용자가 PECOLE을 이용하여 협업을 수행하는 모습을 보여주고 있다.



그림 5. 음성/화상 회의
Fig 5. Audio/Video Conferencing

또한 PECOLE은 [그림 6]과 같이 피어가 서로 다른 언어를 이용하여 웹 브라우저를 공유하는 다국어 협업(multilingual collaboration)과 Telepointer를 이용한 협업을 지원한다.

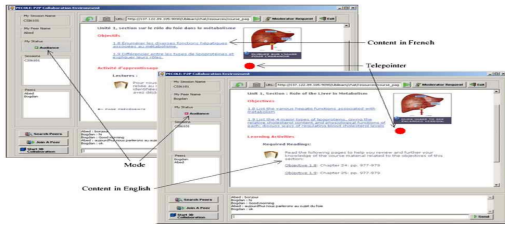


그림 6. Telepointer와 다국어 협업
Fig 6. Telepointer and Multilingual Collaboration

III. PECOLE에서의 다중 그룹 지원

1. PECOLE에서 그룹 관리의 문제점

PECOLE은 피어의 세션을 단위로 협업을 위한 작업공간을 제공하며, 새로운 세션을 생성하거나 세션에 참여하는 기능을 제공하고 있다. [그림 7]은 JXTA 피어그룹 광고를 이용하여 현재 ULASN과 PSLAB으로 구성된 PECOLE의 세션을 다른 사용자들에게 광고하는 예를 보여준다. PECOLE은 공동 작업을 수행하는 그룹을 구별하기 위하여 그룹 아이디인 <GID>를 사용하며 사용자를 의미하는 <Name>을 사용하여 세션과 사용자를 구분하게 된다. PECOLE 세션과 참여한 사용자의 계층을 나타내는데 하나의 <GID>에 복수의 사용자가 그룹의 구성원으로 참여하는 계층 구조를 가진다. 이러한 계층구조는 <GID>에 대하여 1차원적인 그룹 분류를 지원하기 때문에 구현이 단순하다는 장점이 있지만 한 그룹에 참여하면서 동시에 다른 그룹에 참여하거나 다른 그룹의 광고를 수신하는 복잡한 그룹 관리에 있어서 단점이 있다.

```

세션광고 1.
<jxta:PGA>
<GID>urn:jxta:uuid-6B1FB303...</GID>
<MSID>urn:jxta:uuid-DEADBE...</MSID>
<Name>ULSAN</Name>
<Desc>JOHN</Desc>
</jxta:PGA>
세션 광고 2.
<jxta:PGA>
<GID>urn:jxta:uuid-6B1FB303...</GID>
<MSID>urn:jxta:uuid-DEADBE...</MSID>
<Name>PSLAB</Name>
<Desc>ALICE</Desc>
</jxta:PGA>
    
```

그림 7. PECOLE 세션 광고 예제
Fig 7. Example of PECOLE Session Advertisement

[그림 8]은 PECOLE에서 구현한 세션을 사용자가 발견하는 Discovery 매커니즘을 나타낸다. PECOLE의 세션 Discovery 매커니즘은 동일한 <GID>에서 발행한 광고만을 수집하여 세션과 사용자를 구별한다. 이것은 다른 <GID>를 가지는 그룹의 광고를 수신할 수 없게 되어 한 그룹에 참여하고 있으면서 다른 그룹에 참가할 수 없게 된다.

```

// 세션에 대한 광고를 수신했을 경우에 사용자 구별
if(advGen instanceof PeerGroupAdvertisement){
    PeerGroupAdvertisement peerAdv3 =
        (PeerGroupAdvertisement) advGen;
    remPeerName1 = peerAdv3.getDescription();
    Name1 = peerAdv3.getName();
}
// 파이프, 피어에 대한 광고를 수신했을 경우에 사용자
구별
else if(advGen instanceof PipeAdvertisement){
    PipeAdvertisement peerAdv4 =
        (PipeAdvertisement) advGen;
    remPeerName1 = peerAdv4.getDescription();
    Name1 = peerAdv4.getName();
} else if(advGen instanceof PeerAdvertisement){
    PeerAdvertisement peerAdv4 =
        (PeerAdvertisement) advGen;
    remPeerName1 = peerAdv4.getName();
    Name1 = peerAdv4.getDescription();
}
...
}
    
```

그림 8. PECOLE의 세션 Discovery 매커니즘
Fig 8. Session Discovery Mechanism of PECOLE

```

// 그룹에 참여한 사용자가 "Chair"일 경우
if (myPeerName.equalsIgnoreCase("Chair")){
    Owner = true;
    pcessModeratorToken = true;
    jLabelStatus.setText("Owner / Moderator");
} else{
// 그 외의 사용자일 경우
    Owner = false;
    pcessModeratorToken = false;
    jLabelStatus.setText("Audiance");
}
}
    
```

그림 9. PECOLE에서의 사용자 관리
Fig 9. User Management in PECOLE

[그림 9]는 특정 세션에서 사용자의 관리 권한을 피어 이름으로 사용자의 권한을 부여하는 메커니즘을 보여준다. 세션의 관리자일 경우 PECOLE은 그룹에 참여하는 사용자의 권한을 세션의 소유자(Owner/ Moderator)로서 권한을 설정하고 그 외의 이름인 경우에는 참가자(Audiance)로 정의하고 있다. 이렇게 피어

이름만을 가지고 사용자에 대한 권한을 부여하는 것은 사용자 관리에 있어서 구현이 단순하지만 그룹에 참여하여 협업을 수행하고 있는 동안에 권한이 변경되는 것과 같은 유동적인 동작은 지원하지 않고 있지 않다.

2. 그룹 관리 서비스의 구현

2.1 그룹 관리 기능의 정의

1절에서 살펴본 것처럼 1차원적인 그룹 분류를 지원하는 PECOLE의 단점을 보완하기 위하여 다수의 사용자가 효과적인 협업이 가능하도록 복잡한 그룹의 관리가 가능한 그룹 관리 기능을 정의한다. [표 2]에서 정의한 그룹의 기능을 그룹 관리 서비스(Group Management Service)로 제공하여 다중 그룹 관리가 가능하도록 PECOLE을 확장한다.

표 2 다중 그룹 관리를 위한 기능 정의
Table 2 Function Definition for Managing Multiple Groups

기능	설명
그룹의 생성	사용자들의 협업을 수행하는 그룹 생성
그룹의 삭제	사용자들의 협업을 위해 생성한 그룹 삭제
멤버십 인증	그룹에 참여할 때 멤버십으로 참여가 가능한지 인증을 요구
그룹 탐색	현재 협업이 이루어지고 있는 그룹 탐색
그룹 참여	생성되어 있는 여러 그룹에 참여, 복수의 그룹에 참여가 가능함
그룹 탈퇴	참여한 그룹에서 필요한 경우 탈퇴가 가능하도록 한다.
그룹 내의 상호작용	그룹에 참여하는 사용자간에 어플리케이션을 이용하여 협업을 수행

2.2 그룹 관리 서비스의 구현

2.1절에서 설계한 그룹 관리 기능은 PECOLE+에서 그룹 관리 서비스로 제공된다. 그룹 관리 서비스는 그룹의 생성, 삭제, 멤버십 구성, 삭제, 참여, 탈퇴, 발견, 인증과 같은 기능을 제공하여 다양한 상황이 발생할 수 있는 그룹간의 상호작용을 관리할 수 있다. 또한 JXTA의 피어그룹 메커니즘을 이용하여 복수의 피어그룹에 참여한 후 하나의 피어그룹을 하나의 세션으로 이용할 수 있도록 [그림 10]과 같은 구조로 PECOLE의 Session Manager계층을 복수의 피어그룹의 관리가 가능한 그룹 관리 서비스로 확장한다.

그룹 관리 서비스는 피어그룹을 나타내는 Group과 피어그룹에서 발생한 이벤트를 감지하는 GroupListener, 그리고 Group의 전반적인 피어그룹 관리와 이벤트를 제어하는 GroupController

로 구성된다. Group은 그룹을 생성하거나 특정 그룹에 참여할 때마다 피어그룹 객체를 생성하고 GroupListener를 등록한다. GroupListener는 피어그룹에서 발생하는 모든 이벤트를 감지하고 GroupController에 처리를 요청한다. GroupController는 피어가 참여하고 있는 Group 리스트와 탐색된 Group리스트를 관리하고 피어그룹을 생성하거나 참여, 그리고 탈퇴 등의 Group서비스와 GroupListener의 서비스 요청을 처리하게 된다.

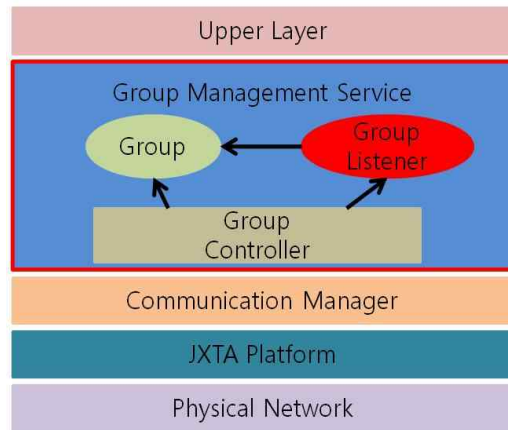


그림 10. 그룹 관리 서비스의 구조
Fig 10. Architecture of Group Management Service

협업을 수행하는 사용자들에 의하여 생성되거나 삭제되는 그룹은 그룹 관리 서비스에서 ArrayList를 이용하여 그룹 클래스를 관리한다. [그림 11]은 생성되거나 삭제된 그룹에 해당하는 Group 클래스에 대하여 관리하는 핵심 코드를 보여준다.

```
private final ArrayList<Group> joinedGroups = new
ArrayList<Group>();
private void addToJoinedGroups(Group p_group){
// 생성된 그룹에 대한 그룹 클래스의 추가
joinedGroups.add(p_group);
}
private void removeFromJoinedGroups(Group p_group){
// 삭제된 그룹에 대한 그룹 클래스의 삭제
joinedGroups.remove(p_group);
}
```

그림 11. 그룹 관리 서비스에서의 Group 클래스 관리
Fig 11. Group Class Management in Group Management Service

Group 클래스는 JxtaObjectRepository를 이용하여 피어에 해당하는 PeerNode와 그룹에 해당하는 GroupNode를 추가/삭제/검색하는 기능을 제공하며 GroupListener 인터페이스의 groupEvent() 메소드를 구현함으로써 이용한다. [그림 12]에서와 같이 notifyListeners 메소드에서 구현한 groupEvent()를

실행하여 그룹 이벤트를 처리한다.

```
s = s != null ? s.trim() : null;
if (this.listeners != null && this.listeners.size() > 0) {
    for (Iterator<GroupListener> iterator =
        listeners.iterator();
        iterator.hasNext();) {
        GroupListener groupListener =
            iterator.next();
        // 그룹 이벤트를 등록함
        GroupListener.groupEvent(this, s, p_re);
    }
}....
```

그림 12. 그룹 이벤트 처리
Fig 12. Processing Group Events

사용자는 네트워크에서 동작하는 하나의 피어로 정의되어 PECOLE+에서 동작한다. PECOLE+는 피어들이 협업을 수행할 수 있도록 논리적인 그룹공간을 제공할 수 있다. 피어는 그룹 관리자로서 그룹을 생성할 수 있는데, [그림 13]은 하나의 피어가 그룹을 생성하는 핵심코드를 보여준다.

```
PeerGroup ppg = parent.getPeerGroup();
PeerGroupAdvertisement pga = null;
// 새로운 그룹을 위한 피어 그룹 광고를 생성
pga = PeerGroupUtil.create(ppg, name, description,
    password, expiration);
if (pga != null) {
    PeerGroup pg = null;
    // 그룹을 생성함
    pg = ppg.newGroup(pga);
    // 그룹이 성공적으로 생성된 경우 그 그룹에 참가
    Group g = new Group(pg, parent);
}
```

그림 13. 피어그룹 생성
Fig 13. Creating PeerGroups

PeerGroupUtil.create() 메소드는 생성한 그룹을 다른 피어들에게 알리기 위한 PGA(PeerGroupAdvertisement, 피어 그룹 광고)를 생성하는데, PGA는 피어들에게 광고하여 생성된 그룹에 참가할 수 있도록 한다. PGA를 파라미터로 가지는 PeerGroup 클래스의 newGroup() 메소드는 피어가 그룹 관리자로서 논리적인 그룹을 생성하고, 그룹 관리 서비스에서 생성된 그룹을 관리하기 위하여 Group 클래스를 생성하여 그룹의 관리자로 참가하게 된다.

[그림 14]는 그룹을 생성했을 때 피어들에게 전달되는 PGA이다. 멤버십관련 인증을 요구할 때는 <Svc> 속성이 추가되어 다른 피어들에게 광고를 하게 된다. 인증을 요구하지 않을 때는 <Svc> 속성이 추가되지 않는다.

현재 생성되어 있는 그룹에 대한 검색은 JXTA의 Discovery 서비스를 이용하여 DiscoveryListener 인터페이스를 구현함으

로써 가능하다. 그룹 관리자가 광고한 PGA를 검색하여 생성된 그룹을 찾을 수 있고, PGA를 통하여 검색한 그룹에 대해서는 Group 클래스를 생성하여 관리한다. [그림 15]는 DiscoveryListener 인터페이스를 구현한 코드를 보여준다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxta:PGA>
<jxta:PGA xml:space="default" xmlns:jxta="http://jxta.org">
  <GID>
    urn:jxta:uuid-3F2A81E655E34830B...
  </GID>
  <MSID>
    urn:jxta:uuid-DEADBEEFDEAFBABA...
  </MSID>
  <Name>
    Wind
  </Name>
  <Svc>
    <MCID>
      urn:jxta:uuid-DEADBEEFDEAFBABA..
    </MCID>
    <Parm type="Param">
      <login>
        jxtauser:DTIA:
      </login>
    </Parm>
  </Svc>
</jxta:PGA>
```

그림 14. 인증을 요구하는 PGA
Fig 14. PGA Requiring Authentication

```
PeerGroupAdvertisement pga = null;
Object o = null;
// 생성한 그룹에 대한 PGA를 검색
while (advertisements != null
    &&
    advertisements.hasMoreElements()) {
    o = advertisements.nextElement();
    if (o instanceof PeerGroupAdvertisement) {
        pga = (PeerGroupAdvertisement)o;
        // PGA가 있을 경우 Group 클래스 생성하고 추가
        Group go = new Group(collaborationMain, pga,
            group);
        add(new GroupNode(go));
    }
}
```

그림 15. 피어 그룹 검색
Fig 15. Searching for PeerGroups

생성한 그룹에 참여하는 것은 그룹 관리자로부터 전달 받은 PGA를 그룹 관리 서비스에서 객체화함으로써 그룹을 구성하는 피어들과의 상호작용을 지원한다. 피어간의 상호작용은 registerCommandDialogPipeListener() 메소드를 실행하여 dialogs/pipe 리스너를 등록함으로써 가능하다. [그림 16]은 그룹에 참여하는 코드의 일부분을 나타낸다.


```

if (group.isVisible()) {
    //dialogs/pipe 리스너를 추가함
    registerCommandDialogPipeListener(group);
}
GroupNode groupNode = new GroupNode(group);
if (group.getParentGroup() != null) {
    groupNode.setParent(
        new
        GroupNode(group.getParentGroup()));
} else {
    groupNode.setParent(JxtaObjectRepository.getRoot());
}
addJxtaNode(groupNode);
PeerNode ownPeerNode =
    new PeerNode(new Peer(pipeAdvertisement), group);
group.setOwnCommandId(pipeAdvertisement.getPipeID());
addJxtaNode(ownPeerNode, true);
restore(group);
if (discover) {
    discover(group);
}
//참여하는 그룹의 객체를 추가
addToJoinedGroups(group);
...
    
```

그림 16. 그룹 참여
Fig 16. Joining Groups

피어가 피어그룹을 객체화할 때는 JXTA의 MembershipService의 구현클래스와 Credential 인터페이스를 이용하여 그룹에 참여하기 위한 인증 절차를 구현한다. [그림 17]은 인증 절차를 구현한 핵심 코드의 일부분이고, 피어그룹을 탈퇴하기 위해서는 [그림 18]과 같이 피어그룹의 MembershipService에서 resign() 메소드를 호출하여 탈퇴할 수 있다.

```

...
MembershipService ms = pg.getMembershipService();
// MembershipService에 대하여 그룹에 참여하는 것을 인증
AuthenticationCredential authCred =
new AuthenticationCredential(pg, "Authentication", null);
InteractiveAuthenticator auth =
    (InteractiveAuthenticator) ms.apply(authCred);
if (auth.interact() && auth.isReadyForJoin()) {
    ms.join(auth);
}
...
    
```

그림 17. 피어그룹 인증
Fig 17. Authenticating PeerGroups

```

...
unregisterCommandDialogPipeListener(peerGroup);
// 그룹에서 탈퇴
peerGroup.getMembershipService().resign();
...
    
```

그림 18. 피어그룹 탈퇴
Fig 18. Leaving PeerGroups

IV. 작업장 관리 서비스

1. PECOLE에서 협업 어플리케이션 관리의 문제점

PECOLE에서는 동작하는 어플리케이션을 관리하기 위하여 Workspace Manager 계층을 제공한다. 이 계층은 Collaboration Application 계층에서 제공되는 협업 어플리케이션들을 관리하는 책임을 가진다. 협업 어플리케이션을 관리하기 위하여 Workspace Manager 계층에서는 각각의 협업 어플리케이션에 대응되는 Manager 컴포넌트를 가지게 되는데, 여기에는 Session Manager와 웹 브라우징과 그룹 채팅을 관리하는 Browser Manager 및 Chat Manager 등이 있다.

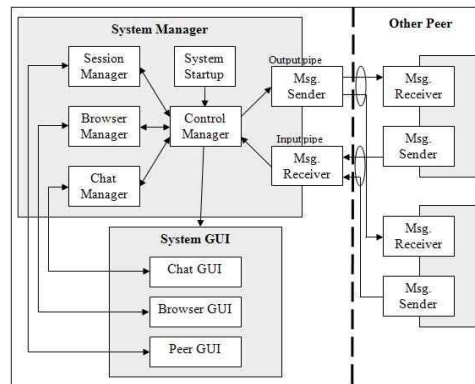


그림 19. PECOLE의 브라우저 컴포넌트 다이어그램
Fig 19. Browser Component Diagram of PECOLE

[그림 19]에서 볼 수 있는 각각의 어플리케이션 Manager는 System GUI에 포함되어 있는 협업 어플리케이션의 GUI에 대응이 되는 형태를 가지게 되는데, 이러한 PECOLE의 구조 하에서는 협업 어플리케이션들을 그룹별로 실행하도록 하고, 한 그룹에서 어플리케이션의 이벤트가 발생할 경우, 동일한 그룹의 사용자들에게 전송하면서 다른 그룹에는 영향을 미치지 않도록 보장하는 어플리케이션 동기화 문제의 해결이 필요하다. 또한 PECOLE에서 기본적으로 제공하고 있는 웹 브라우징이나 채팅 외에도 협업을 위한 어플리케이션은 다양하며 사용자의 필요성에 따라 개발될 수 있다. 이렇게 새로운 어플리케이션을 사용자가 PECOLE에서 사용하기 위해서는 해당 어플리케이션에 대응되는 Manager를 추가하여 관리할 수 있도록 지원하는 것이 바람직하지만 현재 PECOLE에서는 이러한 기능을 지원하고 있지 않다.

2. 작업장 관리 서비스의 구현

1절에서 살펴본 문제점을 보완하기 위해 작업장 서비스 (Workspace Service)와 플러그인 서비스(Plug-in Service)로 구성된 작업장 관리 서비스를 구현한다. 작업장 서비스는 3장에서 구현한 그룹 관리 서비스의 상위에 위치하며 활성화된 그룹을 기반으로 그룹단위의 협업을 가능하게 하는 물리적인 작업장(Workspace)을 제공한다. 플러그인 서비스는 제공되는 협업 어플리케이션을 플러그인 형태로 추가 및 제거가 가능하며, 새로운 어플리케이션의 개발을 용이하게 한다. [그림 20]은 작업장 관리 서비스와 관련된 계층구조를 나타낸다.

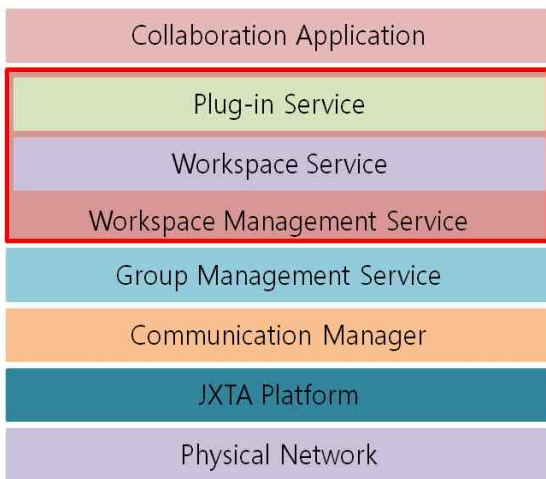


그림 20. 작업장 관리 서비스를 포함하는 PECOLE+ 계층 구조
Fig 20. PECOLE+ Hierarchical Structure with Workspace Management Service

[그림 21]은 그룹 관리 서비스에서 활성화되어 있는 그룹들과 작업장 관리 서비스에서 관리하는 작업장과 플러그인 서비스의 관계를 나타낸다. 그룹 관리 서비스 계층에서 관리하는 피어그룹에 대하여 작업장 관리 서비스에서는 피어 그룹에 대응되는 작업장을 제공한다. 플러그인 서비스에서 관리하는 협업 어플리케이션들은 작업장 별로 실행하도록 구현하고, 각각의 작업장에서는 실행되는 협업 어플리케이션들을 동기화하도록 한다.

결국 제공되는 작업장 관리 서비스는 그룹 별로 관리하는 협업 어플리케이션들의 이벤트를 동기화하고, 현재 소속된 그룹에서 발생된 협업 이벤트에 대하여 같은 그룹의 다른 사용자들에게 전송하여 같은 그룹의 다른 사용자로부터 전달 받은 이벤트를 관리하고 처리하는 책임을 가진다.

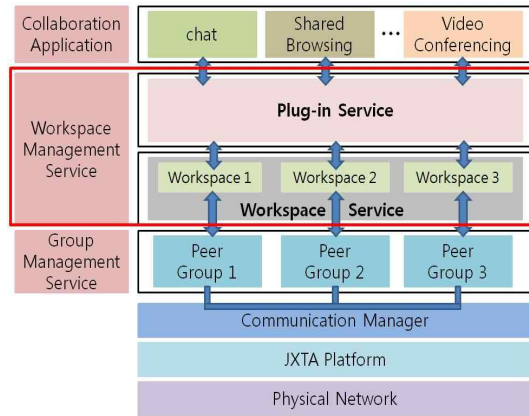


그림 21. 그룹 관리 서비스와 작업장 관리 서비스와의 관계
Fig 21. Relationship between Group Management Service and Workspace Management Service

2.1 작업장 서비스의 구현

작업장 서비스는 협업 어플리케이션 플러그인이 동작하기 위한 독립된 작업공간으로 Dialog를 제공한다. Dialog는 사용자의 협업 어플리케이션 이벤트를 하위의 그룹 관리 서비스 계층을 통하여 그룹 내의 사용자들에게 전송하거나 전달받은 이벤트를 관리하고 처리하는 책임을 가진다. [그림 22]는 작업장 서비스의 구조를 나타낸다.

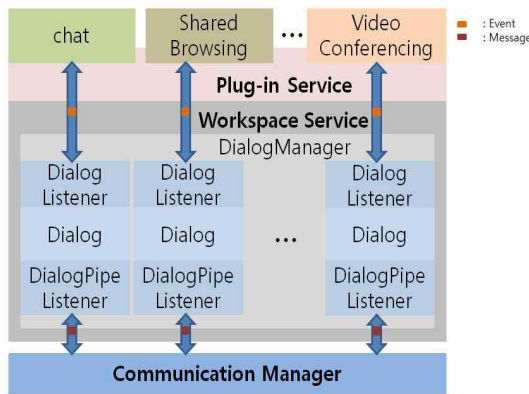


그림 22. 작업장 서비스의 구조
Fig 22. Architecture of Workspace Service

DialogManager는 작업 공간인 Dialog와 이벤트 통신을 위한 DialogPipeListener, 해당 플러그인에서 발생하는 이벤트를 감지하고 처리하는 DialogListener를 관리한다. 각각의 이벤트는 DialogMessage 인터페이스를 구현한 DialogMessageWrapper 클래스로 구현하여 사용자들에게 전달한다. 작업장 서비스는 이벤트의 송·수신을 관리하는 책임을 가지므로, 그룹 내, 외부의 사용자

들에게 메시지를 전달하는 구성요소로서 JXTA에서 제공하는 논리적인 통신 채널인 Pipe의 유형에 따라 DialogManager의 인스턴스를 생성한다. Pipe는 협업 어플리케이션에서 가장 적합한 Pipe를 사용하도록 구현한다. [표 3]은 Pipe의 타입에 따라 반환하는 DialogManager의 종류를 나타낸다.

표 3. DialogManager의 종류
Table 3. Types of DialogManager

TYPE	설명
PropagateType	신뢰할 수 없고 안전이 보장되지 않는 멀티캐스팅 파이프를 이용하는 DialogManager를 반환
UnicastType	신뢰할 수 없고 안전이 보장되지 않는 유니캐스팅 파이프를 이용하는 DialogManager를 반환
UnicastSecureType	신뢰할 수 있고 보안이 보장되는 유니캐스팅 파이프를 이용하는 DialogManager를 반환

[그림 23]은 DialogManager 인스턴스를 생성하는 getInstance() 메소드의 핵심 코드를 보여주고, [그림 24]는 그룹과 파이프광고를 통하여 DialogManager가 getDialog() 메소드를 실행해서 새로운 Dialog를 생성하는 코드는 보여준다.

```
DialogManager dm = null;
name = name != null ? name.trim() : null;
type = type != null ? type.trim() : null;
if (g != null && name.length() > 0 && type.length() > 0) {
    DialogManager tdm = null;
    // 조건문을 사용하여 type과 Pipe의 형태가 같은지 비교하여 DialogManager 인스턴스를 생성
    if (PipeService.PropagateType.equals(type)) {
        tdm = new PropagateDialogManager(name, type);
    } else if (PipeService.UnicastType.equals(type)) {
        tdm = new UnicastDialogManager(name, type);
    } else if (PipeService.UnicastSecureType.equals(type)) {
        tdm = new UnicastSecureDialogManager(name, type);
    }
    ...
    // 생성한 DialogManager 인스턴스를 저장함
    dm = tdm;
    if (managers == null) {
        managers=newHashMap<Group, List<DialogManager>>();
    }
    List<DialogManager> ml = managers.get(g);
    if (ml != null) {
        if (!ml.contains(dm)) {
            ml.add(dm);
        }
    } else {
        ml = new ArrayList<DialogManager>();
        ml.add(dm);
        managers.put(g, ml);
    } }
    // DialogManager 인스턴스를 반환
    return dm;
}
```

그림 23. getInstance() 메소드 구현
Fig 23. Implementation of getInstance() method

```
...
Dialog d = null;
if (c != null && Dialog.class.isAssignableFrom(c) &&
    g != null && pa != null && collaborationMain != null) {
    Dialog td = null;
    // 조건문을 사용하여 전달받은 class c의 타입에 따라서 Dialog를 생성함
    if (c == OneToOneCommandDialog.class) {
        td = new OneToOneCommandDialog
            (g, pa, collaborationMain);
    }
    else {
        DialogManager.IDialogProvider provider
            = dialogProvider.get(c.getName());
        if (provider != null) {
            td = provider.createDialog(g, pa, collaborationMain);
        }
    }
    // 생성한 Dialog를 저장
    d = getDialog(g, td);
}
// Dialog를 반환
return d;
}
```

그림 24. getDialog() 메소드 구현
Fig 24. Implementation of getDialog() Method

[그림 25]는 Dialog 객체가 DialogPipeListener로부터 전달된 메시지를 DialogListener로 전달하는 receive() 메소드의 핵심코드를 보여 준다.

```
...
// 현재 생성되어 있는 리스너에서 해당 리스너로 이벤트를 전달함
if (w != null && w.isValid() && this.listeners != null) {
    synchronized (listeners) {
        for (DialogListener listener : this.listeners) {
            try {
                // 등록된 DialogListener에게 전달함
                listener.receive(w);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
    ...
}
```

그림 25. Dialog의 receive() 메소드 구현
Fig 25. Implementation of receive() Method in Dialog

```
boolean isDispatched = false;
if (this.pipe != null) {
    try {
        // 이벤트를 전송하기 위하여 파이프에 전달함
        isDispatched = this.pipe.sendMessage(msg);
    } catch (IOException ioe) {
        ...
    }
}
return isDispatched;
}
```

그림 26. Dialog의 dispatch() 메소드 구현
그림 26. Implementation of dispatch() Method in Dialog

[그림 26]은 Dialog가 플러그인 서비스로부터 전달되는 이벤트를 처리하는 dispatch() 메소드의 핵심코드를 보여 준다. pipe 객체의 sendMessage() 메소드를 실행하여 이벤트를 처리한다. 성공하면 true, 실패하면 false를 반환하여 isDispatched 변수에 저장한다.

2.2 플러그인 서비스의 구현

플러그인 서비스는 PluginContainer 인터페이스를 상속받은 DefaultCollaborationPginContainer 클래스에 의해서 내부 플러그인 클래스 파일과 외부의 Jar파일로 배포되는 플러그인의 설치, 실행, 종료 등의 다양한 플러그인의 생명주기를 관리한다. [표 4]는 PluginContainer 인터페이스의 주요 기능을 나타낸다.

표 4. PluginContainer 인터페이스
Table 4. PluginContainer Interface

인터페이스	기능
void updateGroupState(Group p_group)	그룹 상태 정보 갱신
void notifyAboutGroupJoin(Group p_group)	그룹에 참여를 알림
void notifyAboutGroupResign(Group p_group)	그룹 탈퇴를 알림
Plugin[] getPlugins()	설치된 모든 플러그인을 반환
Plugin getPlugin(Class className)	플러그인을 반환
public interface ISelectionProvider{ ISelectableNode[] getSelectedNodes(); }	모든 플러그인이 플러그인 컨테이너에 등록되기 위해서 구현해야할 인터페이스
ISelectableNode[] getSelectedNodes()	플러그인 컨테이너에서 현재 선택된 플러그인 노드를 반환
void removeSelectionProvider(ISelectionProvider p_component)	컨테이너 선택 공급자 리스트로부터 해당 플러그인 컴포넌트를 제거
void registerSelectionProvider(ISelectionProvider p_component)	선택 공급자 리스트에 컴포넌트를 등록

표 5. IPluginNotificationHandler 인터페이스
Table 5. IPluginNotificationHandler Interface

인터페이스	기능
void groupJoined(Group p_group)	그룹에 참여할 때
void groupResigned(Group p_group)	그룹을 떠날 때
void groupStateChanged(Group p_group)	그룹의 속성이 변할 때

[표 5]의 IPluginNotificationHandler 인터페이스는 플러그인에 발생한 그룹 이벤트를 관리하고 처리하는 기능을 제공한다. 또한 새로 개발된 협업 어플리케이션을 PECOLE+의 플러그인으로 동작할 수 있도록 [표 6]과 같이 Plugin 인터페이스로 정의하였다. 협업 어플리케이션은 Plugin 인터페이스를 구현한 PluginBase 클래스를 상속받아 구현하면 PECOLE+의 협업 어플리케이션 플러그인으로 동작할 수 있다.

표 6. Plugin 인터페이스
Table.6 Plugin Interface

인터페이스	기능
void init (PluginContainer c)	플러그인 초기화
void start()	플러그인 활성화
void stop()	플러그인 비활성화
void destroy()	플러그인 파괴
boolean isRunning()	플러그인 동작유무 확인
IPluginNotificationHandler getPluginNotificationHandler()	PluginNotificationHandler는 플러그인의 통신을 위해서 사용 플러그인 관리자에게 플러그인의 이벤트 정보를 알림

V. 동작 및 성능

1. PECOLE+의 사용자 인터페이스

PECOLE+는 기본적으로 JXTA 네트워크에 연결하기 위한 로그인 설정 화면을 이용한다. 로그인 설정 화면은 사용자 인증과 피어 역할, 랑데부/릴레이/프락시 서버 등의 네트워크를 구성하기 위한 값을 설정한다. [그림 27]은 사용자의 편의성을 고려한 인터페이스를 제공하는 PECOLE+의 협업 시스템의 GUI를 나타낸다.

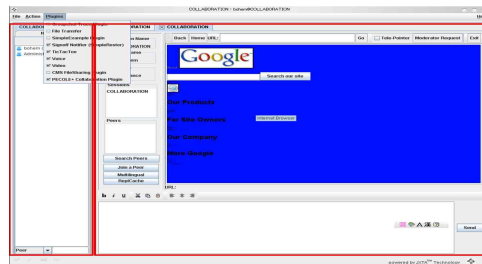


그림 27. PECOLE+의 사용자 인터페이스
Fig 27 User Interface of PECOLE+

메인 화면은 그룹관리 창과 협업 플러그인 창으로 구성되

며 탭 방식으로 표현된다. 협업 플러그인 창에 나타나는 협업 어플리케이션은 다이얼로그로 구현되어, 상단의 메뉴바의 Pugins 메뉴에 구성된 협업 어플리케이션은 협업 플러그인 창에 위치하게 되며 체크-인체크를 이용하여 실행과 정지가 가능하다.

[그림 28]은 그룹 관리 창으로 트리 형식으로 그룹의 계층적인 관계와 현재 참여중인 그룹을 나타낸다. Network 탭에 사용자에게 발견된 그룹의 전체 목록을 보여주며 그룹을 탐색·생성하고, 혹은 참여하고 관리할 수 있는 기능을 제공한다. ①은 현재 그룹 관리를 위한 팝업 메뉴를 나타내고, ②는 그룹을 생성하는 폼을 보여준다. ③은 그룹을 성공적으로 생성한 화면이며, ④는 현재 참여하고 있는 그룹에 대하여 Leave Group 메뉴가 추가되어 그룹을 탈퇴할 수 있다.

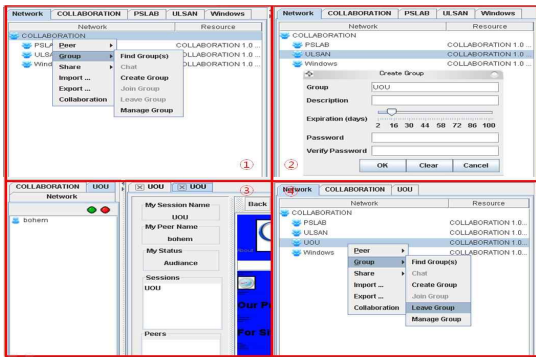


그림 28. 그룹 관리 인터페이스
Fig 28. Interface for Group Management

[그림 29]는 현재 사용자가 참여하고 있는 그룹의 사용자를 나타내어 참여한 사용자를 탐색하거나 사용자의 상태 정보 확인 등의 기능을 제공한다. ①과②는 사용자가 현재 참여하고 있는 그룹을 나타내며, ③과④는 해당 그룹에 참여하고 있는 사용자 아이디를 나타낸다. ③과④ 동일한 아이디로써 같은 사용자이고, ULSAN과 Wind라는 그룹에 동시에 참여하고 있는 것을 보여주고 있다.



그림 29. 그룹에 참여하고 있는 사용자
Fig 29. Users Joining a Group

[그림 30]은 협업 플러그인 창에서 여러 그룹에서 협업을 수행하는 동작을 보여준다. 각각의 탭에는 현재 작업 중인 그룹의 이름이 표시되며 같은 그룹에 속한 협업 어플리케이션 플러그인은 동일한 탭 이름을 가진다. 사용자는 탭을 이동하면서 플러그인으로 구현된 다양한 협업 어플리케이션을 동시에 실행할 수 있다.

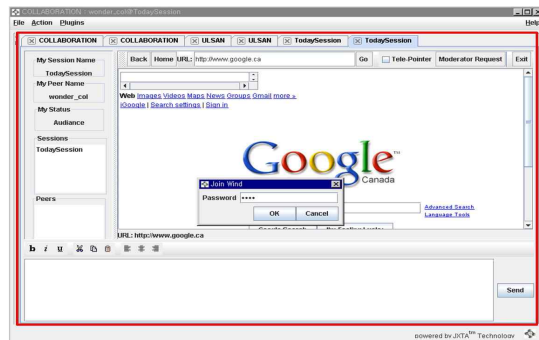


그림 30. 그룹별 협업 어플리케이션의 동작
Fig 30. Execution of Collaborative Applications for Each Group

[그림 31]은 다중채팅의 예를 보여준다. 복수의 그룹에 참가하여 사용자와 서로 대화를 나누는 화면이다. ①과②는 각각 UOU와 Wind라는 그룹을 나타내며 bohem이라는 사용자가 두 개의 그룹에 복수로 참여하고 있다. ③과④는 각 그룹의 협업 플러그인 창을 나타내며 현재는 채팅 어플리케이션을 이용하여 그룹 참가자들과 협업을 수행하고 있다.

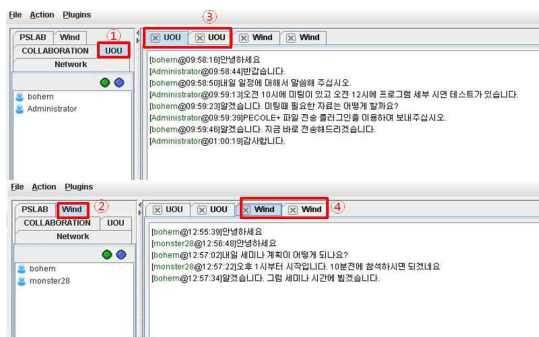


그림 31. PEOCLE+에서 다중 채팅
Fig 31. Multiple Chatting on PEOCLE+

[그림 32]는 그룹 내의 사용자에게 파일을 전송하는 기능을 가지는 파일 전송 어플리케이션을 구현하여 플러그인 형태로 실행하는 화면을 보여준다. ①은 플러그인으로 동작하는 형태를 보여주며 체크 상태일 경우 어플리케이션이 실행된다. ②는 특정 파일을 전송하기 위한 탐색기로서 파일이 있는 경로로 이동 후 선택하면 파일을 전송할 수 있다.

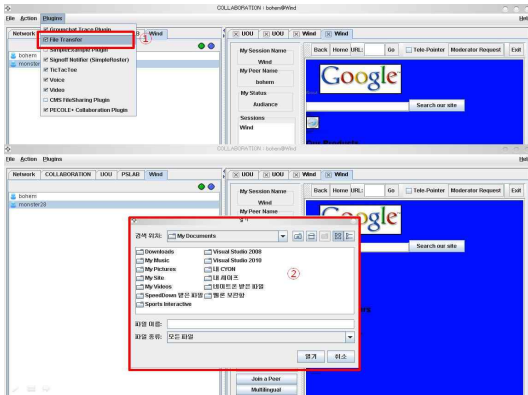


그림 32. 파일 전송 어플리케이션의 동작
Fig 32. Execution of File Transfer Application

2. PECOLE+의 성능

[표 7]은 기존 멀티미디어 협업 환경인 PECOLE과 새롭게 확장된 PECOLE+의 기능과 성능을 비교한다. PECOLE+는 기존 PECOLE의 모든 기능을 이용할 수 있으며 이와 더불어 효과적 협업을 지원하기 위한 추가적인 기능들을 제공하고 있다. 다중그룹 관리 기능은 실제 협업 환경에서 벌어지는 복잡하고 다양한 상황에 대처할 수 있고, 작업장 서비스의 제공으로 사용자에게 그룹 단위의 작업에 효율적인 협업도구로써 제공된다. 또한 플러그인 프레임워크는 기존 협업 어플리케이션의 확장과 새로운 어플리케이션의 개발을 용이하게 한다.

표 7. PECOLE과 PECOLE+의 기능 비교
Table 7. Comparing Functions of PECOLE and PECOLE+

기능	PECOLE	PECOLE+	
프로토콜	JXTA	JXTA	
채팅	O	O	
shared browsing	O	O	
shared telepointer	O	O	
multipoint-to-multipoint	O	O	
audio/video conferencing	O	O	
multilingual collaboration	O	O	
JXTA 환경 설정	X	O	
플러그인 서비스	X	O	
작업장 서비스	X	O	
그룹관리	그룹생성	O	O
	다중그룹생성	X	O
	비공개그룹생성	X	O
	그룹탈퇴	X	O
	그룹삭제	X	O

그룹검색	X	O
피어검색	O	O
다중그룹참여	X	O

[그림 33]은 PECOLE과 PECOLE+의 성능 비교를 나타낸다. 두 시스템간의 성능을 비교하기 위하여 PECOLE과 PECOLE+가 설치된 두 대의 PC에서 채팅 어플리케이션을 실행하였다. 두 대의 PC는 메시지를 송신하는 송신 피어와 메시지를 수신하는 수신 피어로 구성된다. 송신 피어에서는 채팅 어플리케이션에서 100개의 메시지를 메시지 당 1초 간격으로 송신하고, 수신 피어에서는 100개의 메시지를 수신하여 전체 메시지의 전송 시간을 측정하였다. 전체 메시지의 전송시간은 첫 번째 메시지가 도착한 시간으로부터 마지막 메시지가 도착한 시간의 차이이며 이것을 이용하여 두 시스템의 성능을 비교하였다.

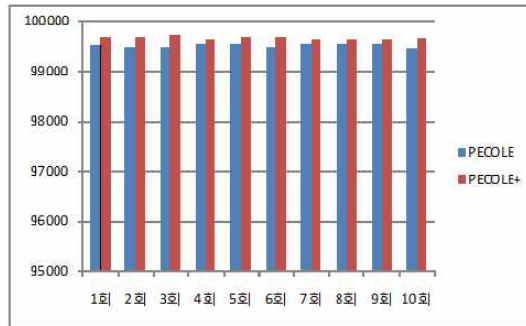


그림 33. PECOLE과 PECOLE+의 성능 비교
Fig 33. performance comparison of PECOLE and PECOLE+

그래프의 가로축은 메시지의 전송시간을 측정한 순서를 의미하고, 세로축은 메시지의 전송시간을 밀리세컨드 단위로 나타낸 것을 의미한다. 측정 결과 PECOLE+의 성능은 기존의 PECOLE과 비교했을 때 그룹 관리 서비스와 작업장 관리 서비스가 추가 되어 약간의 부하가 발생하지만 사용자의 입장에서는 거의 차이가 없다고 판단된다.

VI. 결론 및 향후 연구 과제

본 논문에서는 개방형 P2P 프로토콜인 JXTA를 기반으로 동작하는 분산 협업 환경인 PECOLE+의 개발에 대하여 기술하였다.

기존 PECOLE에서는 하나의 세션을 그룹으로 사용하여 협업을 수행하도록 구현 되어, 다중 그룹의 운용이 불가능하고,

다양한 기능을 포함하는 협업 어플리케이션의 개발, 설치, 배포가 어려움이 있다. 또한 PECOLE에서 다중 그룹을 지원하더라도 그룹별로 어플리케이션을 실행할 때 어플리케이션 간의 동기화 문제를 가지고 있다. 이러한 문제점을 PECOLE을 확장하여 그룹 관리 서비스와 작업장 관리 서비스를 제공하는 PECOLE+를 개발함으로써 해결하였다.

PECOLE+의 그룹 관리 서비스는 실제 협업 환경에서 벌어지는 복잡하고 다양한 협업 환경을 지원하기 위하여 참여중인 그룹의 변경, 복수 그룹 참여, 그룹 탐색 등, 다중 그룹을 운용할 수 있는 기능을 제공한다. 작업장 관리 서비스는 작업장 서비스와 플러그인 서비스를 포함하는데, 작업장 서비스에서는 PECOLE+에서 동작하는 협업 어플리케이션을 그룹 단위로 동기화 하여 그룹 단위의 협업을 효과적으로 수행할 수 있도록 한다. 플러그인 서비스에서는 협업을 위한 다양한 어플리케이션의 배포, 설치, 실행이 용이하며, 사용자의 다양한 요구가 반영된 협업 어플리케이션 개발의 편의성을 제공하기 위하여 협업 어플리케이션의 표준 인터페이스를 정의하였다. PECOLE+에서 제공 되는 서비스는 다중 그룹과 협업 어플리케이션의 사용을 효과적으로 지원하므로 사용자에게 보다 효과적인 협업 활동을 지원할 수 있을 것이다.

향후 과제로는 현재 인터넷과 같은 고정된 네트워크 인프라에서 동작하는 PECOLE+를 네트워크 인프라를 통하지 않고 기기들이 자발적으로 연결될 수 있는 MANET(Mobile Ad hoc Network) 환경에서 동작할 수 있도록 지원하고, 파일 공유나 자발적인 가상 커뮤니티 같은 다양한 협업 서비스를 지원하는 SOA(Service-Oriented Architecture)구조의 P2P 기반 협업 서비스 환경을 연구할 계획이다.

참고문헌

- [1] R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkell, J. Trevor, and G. Woetzel, "Basic support for cooperative work on the World Wide Web," *International Journal of Human-Computer Studies*, Vol.46, No.6 pp.827-846. 1997.
- [2] CoSlide, "Supporting Effective Collaborative Works Based on WebDAV" Dong-Ho Kim; Won-Joon Shin; Jin-Ho Park; Myung-Joon Lee; Strategic Technology, The 1st International Forum on 18-20 Oct. 2006 Page(s):235 - 238
- [3] PhpGroupware, <http://www.phpgroupware.org>
- [4] Wodehouse A; Grierson H; Ion W J; Juster N; Lynne A; Stone A L; 'TikiWiki: a tool to support engineering design students in concept generation', in Proceedings of the International Engineering and Product Design Education Conference, Delft, The Netherlands, September 2004, <http://info.tikiwiki.org/tiki-index.php>.
- [5] Geon-Tae An, Myung-Hee Jung,, Keon-Woong Lee, Nam-Doo Moon, Myung-Joon Lee, "iPlace : A Web-based Collaborative Work System Using Enterprise JavaBeans Technology", *The Korea Information Processing Society Transactions:PartD*, Volume 8D. Issue6, pp.735~746, 2001
- [6] R. Schollmeier. A definition of peer-to-peer networking towards a delimitation against classical client-server concepts. In Proc. of the 7th EUNICE Open European Summer School(EUNICE'01)and the IFIP Workshop on IP and ATM traffic management, Paris(France), Sept. 2001
- [7] Kawashima T, Ma J (2004) TOMSCOP-A synchronous P2P collaboration platform over JXTA. In: Proceeding of the international workshop on multimedia network systems and applications (MNSA'2004), In: Conjunction with the 24th international conference on distributed computing systems. Tokyo, Japan, March 2004, pp 85 - 90
- [8] "PECOLE: P2P multimedia collaborative environment" Saddik A., Rahman A., Abdala S., Solomon B. *Multimedia Tools and Applications* 39(3): 353-377, 2008.
- [9] PHAC: A P2P-based Environment for Distributed Collaborative Applications Adnane CABANI, Srin RAMASWAMY, Mhamed ITMI and J.-P. PECUCHET, 4th International Conference on Distributed Computing and Internet Technology (ICDCIT'07), LNCS 4882, 17-20 December 2007, Bangalore, India. 240-247.
- [10] JCLS: JXTA-Based Collaborative Learning System Zhenhua Wang, Derong Shen, Xiaolu Wang December 2008 CSSE '08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 05 , Volume 05
- [11] JXTA Scalability (2006) <http://platform.jxta.org/java/workinprogress/ScalabilityOverview.pdf>
- [12] M Roseman and S. Greenberg, "TeamRooms : Network places for collaboration," In proceedings of the ACM CSCW'96 Conference on Computer-Supported Cooperative Work, pp.325-333, ACM Press, 1996.

저 자 소 개



김 보 현

2009년 : 울산대학교 컴퓨터정보통신
공학부 졸업(공학사)

2009년~현재 : 울산대학교 컴퓨터정
보통신공학부 석사과정

Email : panicist@mail.ulsan.ac.kr



박 중 문

2008년 : 울산대학교 컴퓨터정보통신
공학부 졸업(공학사)

2010년 : 울산대학교 컴퓨터정보통신
공학부 졸업(석사)

Email : monster28@nate.com



이 명 준

1991년 : 한국과학기술원 전산학과 졸업
(박사)

1982년~현재 : 울산대학교 컴퓨터
정보통신공학부 교수

1993년~1994년 : 미국 버지니아대학
교환 교수

2005년~2006년 : 미국 캘리포니아
주립대학 교환
교수

Email : mjlee@ulsan.ac.kr



박 양 수

1981년 : 서울대학교 대학원 계산통
계 학과 졸업(석사)

1985년 : 서울대학교 대학원 계산통
계학과 수료(박사)

1980년~현재 : 울산대학교 컴퓨터
정보통신공학부 교수

Email : yspk56@ulsan.ac.kr