

# 확장성 있는 프레젼스 기능을 가진 무선 인스턴트 메시징 시스템 설계 및 구현

## (A Design and Implementation of a Mobile Instant Messaging System with Extensible Presence Capabilities)

나 재 욱<sup>†</sup>      최 진 영<sup>†</sup>      조 현 덕<sup>†</sup>  
(Jaewook Nah)      (Jinyeong Choi)      (Hyunduk Cho)

김 용 훈<sup>†</sup>      이 진 구<sup>†</sup>      박 종 태<sup>\*\*</sup>  
(Yonghoon Kim)      (Jingu Lee)      (Jongtae Park)

**요약** 본 논문에서는 확장성 있는 프레젼스 서비스를 제공하는 모바일 인스턴트 메신저를 설계 및 구현하였다. 기존의 여러 가지 인스턴트 메시징 시스템은 상호연동성과 확장성에 문제가 있다. 상호연동성 문제를 해결하기 위하여 국제표준인 IETF의 프레젼스 서비스 구조를 채택하였다. 그리고 무선 이동통신에서 다양한 새로운 프레젼스 기능을 수용하기 위하여 OMA의 IMPS 구현모델에서 정의된 확장성있는 프레젼스 서비스 속성을 구현하였다. 그리고 인스턴트 메시징에서 사용자의 임의 설정에 따라 프레젼스 정보 표현이 가능하도록 확장성 있는 프레젼스 서비스 라이브러리를 설계 및 구현하였다. 구현된 라이브러리를 무선 E-mail 시스템 구현에 적용하여 확장성 있는 프레젼스 서비스 기능을 확인하였다.

**키워드** : 프레젼스 서비스, 인스턴트 메시징, IMPS, SIMPLE, SIP

**Abstract** In this paper, we have designed and implemented mobile instant messenger which has extensible presence service capability. Most previous work on instant messaging system has some limitation with regard to interoperability and extensibility. To solve these problems, we have employed the presence service architecture of IETF. In order to provide extensible presence capability in the wireless mobile communication, we have employed the presence service attributes which are defined in OMA's IMPS architecture. In particular, we have designed and implemented the presence service libraries for manipulating the presence information in response to user's requirement. Finally, we have developed the wireless E-mail service, employing the presence service architecture, to verify the extensibility.

**Key words** : Presence Service, Instant Messaging, IMPS, SIMPLE, SIP

### 1. 서론

초고속 인터넷과 융합형 네트워크 및 서비스가 발전함에 따라 다양한 킬러 응용 프로그램이 개발되고 있다.

그중 인스턴트 메시징 서비스는 대표적인 킬러 응용 프로그램의 하나이다. 인스턴트 메시징 서비스는 사용자들 간의 의사 전달의 편의성을 증대시키기 위해 기존의 채팅 서비스와 차별화 된 기능을 제공하고 있다. 일반적인 인스턴트 메신저는 단문 통신, 파일전송, 음성 및 비디오 통신 서비스 등을 제공하며 최근에는 이동통신망을 이용하여 이동단말과 유선단말 사이의 인스턴트 메시징 서비스까지 제공하고 있다. 인스턴트 메시징 서비스에는 사용자의 간단한 상태 정보를 나타내는 프레젼스 서비스 기능이 탑재되어 있다.

프레젼스 서비스란 대화자의 현재 상태를 표시해주는 서비스를 말한다. 마이크로소프트사의 MSN 메신저에서

<sup>†</sup> 학생회원 : 경북대학교 정보통신학과  
jwnah@ee.knu.ac.kr  
hdcho@ee.knu.ac.kr  
jingu@ee.knu.ac.kr  
semodori@ee.knu.ac.kr  
yhhkim@ee.knu.ac.kr

<sup>\*\*</sup> 종신회원 : 경북대학교 전자전기컴퓨터학부 교수  
jtpark@ee.knu.ac.kr

논문접수 : 2005년 8월 11일  
심사완료 : 2006년 3월 23일

대화 상대의 대화참여 여부를 표시해주는 것이 대표적인 예이다. 현재 마이크로소프트의 MSN 메신저, AOL의 AIM, Yahoo의 Yahoo 메신저 등에서 프레젼스 서비스를 제공하고 있다. 프레젼스 서비스 기능이 부가된 인스턴트 메시징의 관련 국제 표준 기술로는 IETF의 SIP 기반의 SIMPLE(SIP for Instant Messaging and Presence Leveraging Extensions)과 OMA(Open Mobile Alliance)의 Messaging 및 IMPS(Instant Messaging and Presence Service)기술이 있다[1].

그러나 위에서 언급한 인스턴트 메신저에는 몇 가지 문제점이 있다. 첫째, 기존의 메신저 서비스는 독자적인 프로토콜을 사용하기 때문에 다른 벤더의 메신저 서비스와 상호 호환성을 가지지 못한다는 것이다. 그로 인하여 인스턴트 메시징 서비스 업체별로 각각의 메신저 프로그램을 설치하여 사용해야 하는 번거로움이 있다. 둘째, 기존의 메신저 서비스에 탑재된 프레젼스 서비스는 매우 단순하여 대화자의 다양한 상태 정보를 표현하는 기능이 미흡하다. 일부의 메신저 서비스에서 몇 가지 상태를 미리 설정하여 두는 경우도 있긴 하지만 그 또한 대화자의 다양한 상태정보를 표현하기에는 부족하다.

이러한 문제점을 해결하기 위해서는 다음과 같은 연구가 필요하다. 첫째, 표준기술을 적용한 상호연동 가능한 프레젼스 서비스의 개발이다. 표준 인스턴트 메시징 프로토콜을 사용하면 상호 호환성을 확보 할 수 있다. 기존의 연구에서 SIP 표준과 OMA의 IMPS 표준의 동시제공 필요성이 언급되어있다[2]. 이와 관련해 IETF의 SIP, SIMPLE 워킹그룹과 OMA의 Message와 Presence and Availability 워킹그룹에서 표준을 연구하고 있다. 이러한 표준 기술을 사용하면 하나의 인스턴트 메신저에서 다른 메시징 서비스와 연동가능하다. 둘째, 확장 가능한 프레젼스 서비스를 개발하는 것이다. 단순한 프레젼스 기술과 달리 확장 가능한 프레젼스 기술은 풍부한 상태 정보 표현이 가능하다. 이는 이동 통신망을 기반으로 이동 단말 무선 인스턴트메시지에 꼭 필요한 기능이다. 왜냐하면 이동통신망에는 사용자 및 단말의 상태 정보 표현이 유선망 기반의 단말의 그것과 다르게 아주 다양한 형태로 표현될 수 있기 때문이다.

본 논문에서는 앞에서 언급한 인스턴트 메시징 및 프레젼스 서비스의 문제점을 해결하기 위해서 확장성 있는 프레젼스 서비스를 제공하는 모바일 인스턴트 메신저를 설계하고 구현하였다. 구체적으로 IETF의 SIMPLE 기술과 OMA의 IMPS에 정의된 프레젼스 속성을 적용하여 시스템 구조를 설계하고 응용서비스를 구현하였다. IETF의 SIMPLE 워킹그룹에서 제안된 기술은 기본적인 프레젼스 모델을 탑재한 인스턴트 메시징 및 프레젼스 서비스 구조를 제시하고 있다. OMA에서는

IETF SIMPLE의 프레젼스 서비스 기능을 보완한 무선 이동통신 환경에서 프레젼스 서비스를 제공하기 위한 속성(Attribute)들을 정의한다. 확장 가능한 프레젼스 서비스는 사용자의 능력(capabilities)이나, 특징(characteristics)과 통신 의지(willingness) 및 수단(means)의 변화 등을 동적으로 나타낼 수 있게한다. 여기서 수단(means)은 인스턴트 메신저, 이동전화, e-mail, 게임 등일 수 있고, 의지(willingness)는 “급한 용무만 연락”, “e-mail로 연락” 등을 나타낼 수 있다. 능력(capabilities)과 특징(characteristics)은 음성메일 가능, 음성통신 혹은 비디오 통신 가능, 다국어지원 등의 기계적인 성능 및 사용자 환경 상태 등을 나타낼 수 있게 한다. 그외 사용자의 상태뿐만 아니라 추상화 기술을 적용하여 사람의 상태뿐만 아니라 단말, 서비스 등의 상태를 표현하는 것도 가능하기 때문에 이동통신망 제어 및 관리에도 응용될 수 있다[1].

본 논문에서는 IETF의 SIMPLE 구조에서 OMA의 확장성 있는 이동통신 프레젼스 서비스가 가능하도록 시스템을 설계하고 구현하였다. 구현시스템에서 각각의 모듈은 API 형태로 제공되게 하여 새로운 프레젼스 서비스가 추가되었을 경우에도 프레젼스 구조변경 및 생성이 용이하도록 하였다. 또한 프레젼스 서비스 기능을 라이브러리 형태로 구현하여 제공함으로써 전체 인스턴트 메시징 서비스 시스템에서 탑재가 용이하도록 하였다. 특히 구현된 확장성 있는 프레젼스 서비스 기능을 E-mail 시스템에 적용하여 E-mail 알림과 검색 기능을 프레젼스 기술로 표현함으로써 언제 어디서나 다수 계정의 E-mail의 내용을 즉시 파악할 수 있는 첨단 모바일 응용 서비스를 개발하였다.

2장에서는 인스턴트 메시징 및 프레젼스 서비스에 관련된 기존 연구 및 표준화 동향에 대해서 분석한다. 3장에서는 본 논문에서 제안한 인스턴트 메시징 및 프레젼스 서비스 시스템을 구성하는 서버와 클라이언트 시스템의 구조 및 세부모듈의 기능을 소개한다. 그리고 E-mail 시스템에 적용한 확장성 있는 모바일 프레젼스 시스템을 구현한다. 마지막으로 4장에서는 향후 연구 내용을 소개하고 결론을 맺는다.

## 2. 기존 연구 및 표준화 기술 동향

### 2.1 인스턴트 및 프레젼스 관련 기술 국제 표준화 동향

인터넷 표준 규격을 제정하고 있는 IETF에서는 인스턴트 메시징 및 프레젼스와 관련된 표준 제정을 위하여 다양한 워킹그룹을 결성하였다. 대표적인 워킹그룹으로는 SIMPLE(SIP for Instant Messaging and Presence Leveraging Extensions)과 IMPP(Instant Messaging and Presence Protocol)를 들 수 있다.

IMPP 워킹그룹은 인스턴트 메시징 및 프레전스 프로토콜과 데이터 포맷의 표준을 정의하기 위해서 결성된 워킹그룹이다. 현재 IMPP 워킹그룹에서는 인스턴트 메시징과 프레전스 프로토콜을 위한 추상적인 기본모델을 제시한다. 또한 인스턴트 메시징 및 프레전스 서비스에 대한 용어 정리와 시스템 보안을 위한 최소 요구사항을 정의하고 있다. 이와 관련된 내용은 IMPP 워킹그룹에서 제정한 표준인 RFC2778과 RFC2779에서 정의하고 있다. 표 1은 IETF의 SIMPLE 워킹그룹과 IMPP 그룹에서 제정한 표준 기술 목록이다.

IMPP 표준이 인스턴트 메시징 및 프레전스 프로토콜 구성에 필요한 기본 모델과 요구사항들을 정의하고 있다면 SIMPLE 워킹그룹에서는 SIP 표준 프로토콜을 기반으로 인스턴트 메시징 및 프레전스 프로토콜에 적합한 서비스 표준제정을 목적으로 결성되었다. 현재 SIMPLE 워킹그룹의 표준에서는 인스턴트 메시징 서버를

거치지 않고 UA(User Agent)간에 직접 주고받을 수 있도록 정의하며 UA가 서버에 로그인시 서버에서 IP주소와 사용 중인 포트 번호 및 현재 위치를 알려주며 특정 UA에 대한 프레전스 정보를 클라이언트 UA에게 전송하는 기능들을 정의하고 있다[3].

이동통신 시장에서는 2002년 6월 기존 이동통신 네트워크에 적용하기 위한 응용표준을 연구하는 표준화 포럼들을 통합하여 Market Driven 방식의 표준화를 지향하는 새로운 표준화 기구인 OMA를 결성하였다. OMA는 2G, 2.5G에 적용하는 독자적인 인스턴트 메시징 및 프레전스 서비스를 제정하고 있다. OMA에서는 기존 이동통신망에서 프레전스 서비스를 적용하기 위해서 결성된 Wireless Village Forum의 프레전스 규격을 따르는 Presence & Availability 워킹그룹을 결성하였다[4]. Presence & Availability 워킹그룹에서는 모바일 디바이스로 정의되는 클라이언트와 프레전스 정보를 저장하

표 1 IETF 인스턴트 메시징 및 프레전스 관련 표준 기술

SIMPLE	RFC3856	프레전스의 가입과 통지를 위해서 사용하는 SIP에 대해서 기술
	RFC3857	SIP이벤트 프레임을 위한 watcher 이벤트 임시 패킷 정의
	RFC3858	Watcher가 자원에 대한 정보 요청하는 인자를 정의하기 위하여 사용하는 XML문서 규격 정의
	RFC3994	새로운 상태의 메시지 타입과 XML Namespace가 운반하는 메시지 결합에 관한 정보를 정의
IMPP	RFC2778	프레전스 및 인스턴트 메시징 서비스 추상적 기본모델 제시 다양한 컴포넌트 및 새로운 용어에 대한 정의 인터넷 상의 사용자 실시간 통신 가능한 상태 및 통신 방법 정의 인스턴트 메시징 서비스를 위한 기본 모델을 소개
	RFC2779	프레전스 및 인스턴트 메시징 서비스 프로토콜 요구사항 기술 프레전스 서비스를 위한 프로토콜 요구사항 정의 인스턴트 메시징 서비스를 위한 프로토콜 요구사항 정의 시스템 보안을 위한 최소 요구사항 정의

표 2 인스턴트 메시징 및 프레전스 기술 기존 연구 분석

논문 제목	IETF SIMPLE	OMA IMPS	Extensible Presence Service
Mobile Instant Messaging[9]	No	No	No
The War of Presence and Instant Messaging[10]	Yes	Yes	No
Presence Technology[11]	Yes	No	No
Presence Service In 3G Networks[12]	Yes	No	Yes
Extensible Mobile Presence[13]	Yes	No	Yes
Instant Messaging and Presence Service Using SIMPLE[14]	Yes	No	No
Instant Messaging & Presence Management in Mobile Ad-Hoc Networks[15]	Yes	No	No
Standard SIP-Based Instant Messaging and Presence API for Networked Devices[16]	Yes	No	No
Personalization for SIP Multimedia Communications with Presence[17]	Yes	No	Yes
Fieldcast2: flexible P2P architecture for presence information sharing[18]	Yes	No	Yes
Integrating Service Discovery Protocol with Presence-based Communications for Ad hoc Collaborative Scenario[19]	Yes	No	Yes
Document presence notification services for collaborative writing[20]	Yes	No	Yes

고 있는 서버에 대한 표준들을 정의 한다. 관련 표준 내용들은 클라이언트와 서버간의 통신규약을 정의하는 CSP(Client Server Protocol), 서버와 서버간의 통신규약을 정의하는 SSP(Server Server Protocol), 외부네트워크와의 프레젼스 정보교환을 위해서 사용하는 SAP(Service Access Point), 프레젼스 전송을 위해서 사용되는 명령들을 정의하는 CLP(Command Line Protocol) 그리고 XML을 이용하여 프레젼스의 속성들을 정의하고 있다[5-8].

표 2는 인스턴트 메시징 및 프레젼스 기술에 대한 관련 연구 논문의 비교 분석을 보여주고 있다. 대부분의 연구에서는 IETF의 SIP/SIMPLE 표준기술을 적용하였고 일부 프레젼스 기술을 적용한 연구에서는 확장성을 고려하고 있다[11,12]. 그러나 확장 가능한 프레젼스 속성에 대한 연구가 부족하고, 확장된 프레젼스 서비스의 구조를 제시하지 않는다.

기존의 모바일 인스턴트 메시징 서비스는 표준 프로토콜에 구애받지 않고 연구되었다[9]. 인스턴트 메시징 서비스의 상호 호환 지원과 확장성 있는 프레젼스 서비스 지원을 위해 IETF의 SIP/SIMPLE 규격을 중심으로 관련 연구가 진행되고 있다[14,16]. 하지만 IETF의 SIP/SIMPLE 규격은 인스턴트 메시징 서비스에 SIMPLE 표준 프레젼스 기술을 적용하는 방법에 국한되어 진행되었고 프레젼스 서비스를 제공하기 위한 모델을 제공할 뿐 구체적인 프레젼스 속성(Attribute)을 제공하지 않는다. 또한 인스턴트 메시징과 관련된 대부분의 연구가 SIMPLE 기반의 프레젼스 서비스를 제공하지만 국제적인 대형 이동통신 벤더들의 표준인 OMA의 표준 기술을 수용하고 있지 않다[12,13,15,16]. 기존의 인스턴트 메시징 및 프레젼스 연구는 주로 유선망 기반의 단말에 운용되는 것을 목표로 하였다. 비록 일부 무선망 기반의 모바일 인스턴트 메시징 서비스 연구도 있었으나 단지 OMA의 표준을 기반으로 서비스를 제공할 뿐 확장성 높은 프레젼스 기술이 적용되지는 못하였다[10]. 이러한 서비스들은 단순히 인스턴트 메시징 서비스에 표준 모델을 적용한 시스템에 국한되었고 서비스 추상화기술을 적용하지 않고 있다[10].

### 3. 프레젼스 기반 인스턴트 시스템 설계

#### 3.1 IMPS를 위한 서버 설계

본 논문에서는 그림 1의 IETF SIMPLE WG의 프레젼스 서비스 모델을 참고하여 설계 및 개발하였다. 프레젼스 서비스 모델은 PRESENCE SERVICE, PRESENTITY, WATCHER로 구성되고 WATCHER는 FETCHER, POLLER, SUBSCRIBER로 구성된다.

PRESENCE SERVICE는 PRESENTITY와 WATCHER

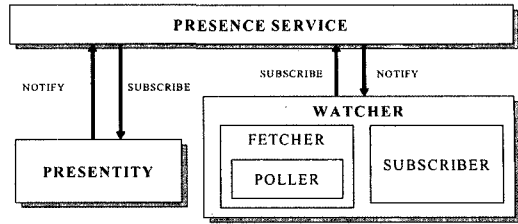


그림 1 Presence Service Model

사이에서 PRESENCE INFORMATION을 관리하고 분배하는 역할을 한다. PRESENTITY는 PRESENCE INFORMATION을 저장하고 분산하는 역할을 하는 클라이언트를 지칭한다. 즉 이 클라이언트의 PRESENCE INFORMATION을 PRESENCE SERVICE에 전송하고 이 정보를 WATCHER에게 알리게 된다. PRESENTITY가 가지고 있는 PRESENCE INFORMATION이 생성되거나 변경되게 되면 NOTIFY 명령을 이용하여 PRESENCE SERVICE에 알린다. WATCHER의 역할은 PRESENCE INFORMATION을 요청하고 수신하는 기능을 가진 Client이다. WATCHER의 FETCHER는 어떠한 PRESENTITY의 현재 PRESENCE INFORMATION 값을 요청한다. 이때 POLLER가 주기적으로 현재 값을 요청하게 된다. SUBSCRIBER는 PRESENTITY의 PRESENCE INFORMATION이 변화되게 되면 NOTIFY 해 줄 것을 요청하는 역할을 한다. 이러한 IETF의 SIMPLE 표준을 기반으로 모델링을 하고 OMA의 프레젼스 정보를 표현하기 위한 XML을 사용하였다. OMA의 프레젼스 정보 표현 예제는 그림 2와 같다. 그림 2는 OnlineStatus를 표현하기 위한 표준 XML 문서이다.

<OnlineStatus>는 OMA Presence Attributes에 정의되어 있는 속성으로써 클라이언트의 On-Line 상태를 표현한다. 이러한 상태 정보는 표 3에서와 같이 정의되어 있다. 일반적으로 Attribute의 Name과 Value Field의 유효성 검사를 나타내는 Qualifier 그리고 Presence Information의 Value를 저장하는 value 항목으로 나누게 된다. 표 3의 Client와 User 항목은 단말의 상태와 사용자의 상태를 구분하고 있으며, Suggest 항목은 프레젼스 서비스를 제공하는 시스템에 기본적으로 제공되어야 하는 항목을 나타내고 있다. 이는 다른 제조업체의 프레젼스 제공을 위한 클라이언트 사이에서 기본적인

```
<PresenceSubList
xmlns="http://www.openmobilealliance.org/DTD/WV-PA1.2">
  <OnlineStatus>
    <Qualifier>T</Qualifier>
    <PresenceValue>T</PresenceValue>
  </OnlineStatus>
</PresenceSubList>
```

그림 2 OnlineStatus 상태 정보 표현 표준 XML문서

표 3 Presence Attribute

	속 성	설 명	Suggested
General	Name	Name of the attribute	
	Qualifier	Validity of Value field	
	Value	Value of the Attribute	
Client Status	OnlineStatus	클라이언트 장비가 서버에 접속 여부	Y
	Registration	클라이언트 장비의 모바일 네트워크에 등록 여부	N
	ClientInfo	클라이언트 정보	N
	TimeZone	클라이언트의 지역 시간 정보	N
	GeoLocation	클라이언트의 지리적 위치 정보	N
	Address	클라이언트의 주소	N
	FreeTextLocation	클라이언트의 임의 주소	N
	PLMN	PLMN 정보	N
User Status	Communication capabilities	통신 환경	N
	UserAvailability	통신을 위한 사용자의 가능상태	Y
	PreferredContacts	선호하는 연락처	N
	PreferredLanguage	선호하는 언어	N
	StatusText	사용자 상태를 표현하기 위한 단문	Y
	StatusMood	사용자의 감정상태	N
	Alias	사용자의 별명	N
	StatusContent	사용자가 자신의 상태 정보를 표현하기 위한 멀티미디어 정보	N
	ContactInfo	사용자에게 연락하기 위한 방법정의	N
InfoLink	기타 사용자 정보를 알리기 위한 URL 주소	N	

상호 운용성을 보장하기 위함이다.

이 밖에도 OMA에서는 필요시에 Presence Attribute를 확장 할 수 있도록 표준이 작성되어 있다. 본 논문에서는 확장된 프레젼스 정보를 제공하기 위하여 추가적인 Presence Attribute를 정의하고 있다. 아래의 그림 3은 e-mail 정보를 프레젼스 정보를 XML로 표현하는 예를 보여 준다.

```

<PresenceSubList
xmlns="http://www.openmobilealliance.org/DTD/WV-PA1.2"
xmlns:Ext="http://www.extentionpresence.com">
<Email>
  <Qualifier> T </Qualifier>
  <Ext:Index>1 </Ext:Index>
  <Ext:From>alice@aaa.com </Ext:From>
  <Ext:To>bob@bbb.com </Ext:To>
  <Ext:Subject> Hello </Ext:Subject>
  <Ext:Contents>Mail Contents</Ext:Contents>
</Email>
</PresenceSubList>
    
```

그림 3 확장된 프레젼스 정보 표현 XML 문서

확장된 프레젼스 정보를 표현하기 위하여 새로운 DTD를 정의하였다. 새롭게 정의된 DTD는 표준 DTD가 있는 URL 정보 뒤에 확장하여 DTD의 URL 위치를 표현한다. <Email> tag는 본 논문에서 확장된 프레젼스 정보를 표현하는 예를 보여준다. <Email> Tag에는 하위 튜플로써 <Qualifier>, <Ext:Index>, <Ext:From>, <Ext:To>, <Ext:Subject> 그리고 <Ext:Contents>이 있다. <Qualifier>는 OMA의 표준 tag로써 유효한 값을 포함하는지를 Boolean 값으로 표현한다. 유효한 값인지의 검사는 DTD에 알려지지 않은 속성이거나 정의

```

<?xml version="1.0" encoding="UTF-8">
<!ELEMENT PresenceSubList (OnlineStatus?, Registration?, ClientInfo?, ... .. Email?)>
<!ELEMENT Qualifier (#PCDATA)>
<!ELEMENT PresenceValue (#PCDATA)>
... 중간 생략 ...
<!ELEMENT Email(Qualifier?, Ext:Index?, Ext:From?, Ext:To?, Ext:Subject?, Ext:Contents?)>
<!ELEMENT Qualifier (#PCDATA)>
<!ELEMENT Ext:Index (#PCDATA)>
<!ELEMENT Ext:From (#PCDATA)>
<!ELEMENT Ext:To (#PCDATA)>
<!ELEMENT Ext:Subject (#PCDATA)>
<!ELEMENT Ext:Contents (#PCDATA)>
    
```

그림 4 E-Mail 프레젼스 정보 표현을 위한 DTD

되지 않은 속성인지를 검사하게 된다. e-mail 표현을 하기 위한 DTD는 그림 4와 같다. 그리고 Index, From, To, Subject, Contents는 POP3를 이용하여 해당하는 메일 서버로부터 정보를 가져 와서 표현하게 된다.

그림 5는 IMPS서비스를 제공하기 위한 시스템 구성도를 나타내고 있다. 시스템의 부하를 분산시키기 위해서 외부에서 들어오는 데이터나 메시지를 전송하기 위한 프록시 서버와, 사용자 및 프레젼스 정보를 관리하는 프레젼스 서버, 메시지 교환을 위한 메시지 프록시 서버로 구분하였다. 또한, 사용자 및 프레젼스 정보들을 저장하기 위한 데이터베이스가 존재한다.

그림 6은 단말기간의 메시지나 멀티미디어 데이터 전송을 위해 단말기와 서버간의 메시지 흐름도를 나타낸 것이다. 사용자는 프레젼스 서버에 REGISTER 메시지를 전달함으로써 인증을 받고, 프레젼스 서버는 데이터베이스에 이미 저장된 사용자의 버디 리스트와 상태 정

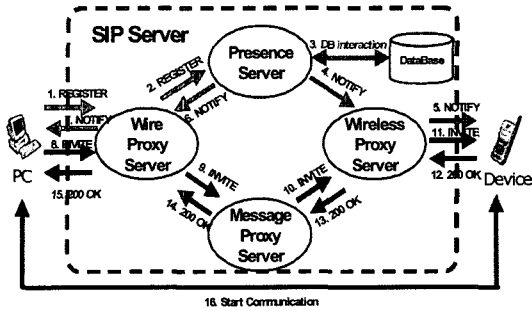


그림 5 IMPS 서비스 시스템 구성도

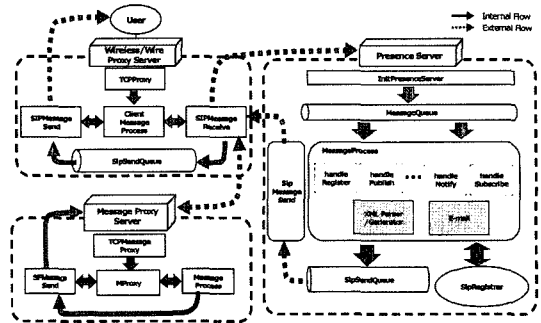


그림 7 SIP 서버 전체 구성도

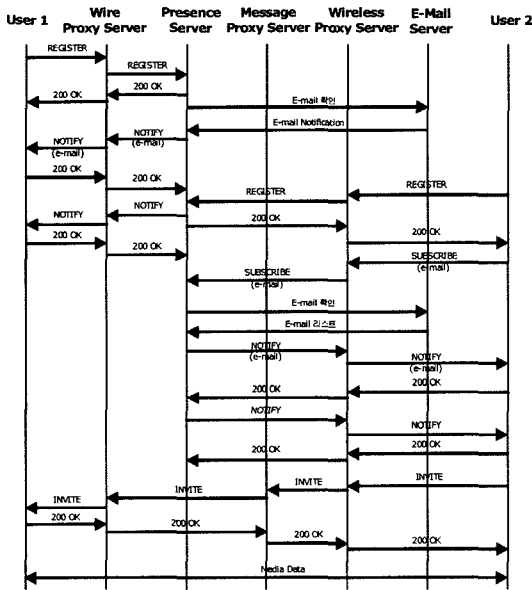


그림 6 메시지 흐름도

보를 참조하여 응답 메시지 NOTIFY 메시지를 통해 그 정보를 사용자에게 전달하고, PUBLISH 메시지를 통해 사용자의 상태 정보를 서버에게 알린다. 그리고 버디리스트 추가 삭제를 하기 위해서 SUBSCRIBE / UNSUBSCRIBE 메시지를 사용한다. 메시지 및 멀티미디어 데이터를 송수신하기 위해서 INVITE 메시지를 통해 단말기간의 세션을 맺는다. 이 때 메시지 프록시 서버는 초대될 사용자의 연결된 세션에 대한 IP주소와 포트 정보를 유지하고 있으므로 사용자 간의 멀티미디어 데이터 전송이 가능하다.

**3.2 IMPS를 위한 SIP 서버 구현**

서버 시스템 구현 환경은 커널 버전 2.4.18-3인 Linux 환경에서 GCC/G++를 이용하여 프레젠턘 서버를 비롯한 프록시 서버와 메시지 프록시 서버들을 구현하였다. 구현된 서버는 Mobile-to-Mobile, Mobile-to-PC

그리고 PC-to-PC의 형태로 인스턴트 메신저 서비스 지원이 가능하다.

그림 7은 SIP 서버의 전체적인 시스템 구성도를 나타낸 것이다. SIP서버를 크게 세 부분으로 나눌 수 있는데 각 서버들은 응답 및 요청 메시지를 구성하는 모듈과 메시지들을 송수신하는 모듈이 존재한다.

**3.2.1 프록시 서버와 메시지 프록시 서버 세부 모듈**

Proxy 서버는 세션 단위로 관리되며 멀티 캐스트를 지원한다. 그러므로 여러 명의 사용자에게 동시에 메시지를 보낼 수 있다. 이동 단말기의 경우 이동통신 사업자 자체망의 IP주소와 포트를 사용하기 때문에 단말기에게 수신 되어야 할 인스턴트 메시지들을 포워딩하기 위해 TCP 기반의 Proxy 서버와 Message Proxy 서버가 필요하다. 그림 8에서는 이러한 IM 서비스를 제공하기 위한 프록시 서버와 메시지 프록시 서버를 구성하는 세부 기능 모듈을 보여주고 있다.

**3.2.1.1 TCPProxy**

TCPProxy 모듈은 단말기가 접속을 하면 해당 세션

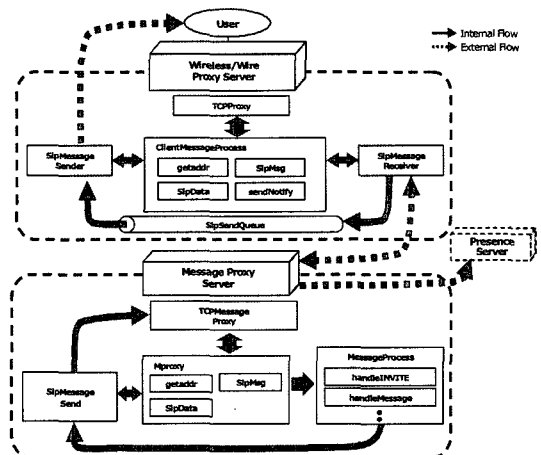


그림 8 프록시 서버와 메시지 프록시 서버 시스템 세부 구성도

을 유지하고 세션에 대한 정보들을 가지고 있다. 이 정보들은 프레젼스 서버나 메시지 프록시 서버들에 의해 처리된 메시지들을 다시 포워딩하기 위해 이동 단말기와 TCP 기반으로 세션을 유지한다.

3.2.1.2 ClientMessageProcess

ClientMessageProcess 모듈은 이동 단말기 혹은 메시지 프록시 서버로부터 받은 메시지를 구성한다. 이때, SIP 메시지 정보와 필드 정보, 단말기에 대한 접속 정보는 getaddr, SipMsg, SipData와 같은 SIP의 동작을 구현하는 클래스 및 함수에서 메시지를 처리한다. sendNotify 함수는 이동 단말기가 임의 종료가 될 경우, 프레젼스 서버에게 이동 단말기의 상태정보를 전송하기 위한 메시지를 구성한다.

3.2.1.3 SIPMessageSend와 SIPMessageReceive

SIPMessageReceive 모듈은 이동 단말기, 프레젼스 서버 혹은 메시지 프록시 서버로부터 응답 메시지들을 받고, SIPSendQueue에 저장된 메시지들을 SIPMessageSend 모듈에서 이동 단말기로 보내는 역할을 한다.

3.2.1.4 TCPMessageProxy

TCPMessageProxy 모듈은 프록시 서버와 접속을 유지하고 프록시 서버의 SIPMessageSend 모듈로부터 메시지를 수신한다. 또한, MessageProcess에서 처리된 응답메시지를 프록시 서버로 송신하기 위해 프록시 서버와 TCP 기반으로 세션을 유지한다.

3.2.1.5 MProxy

MProxy 모듈은 프록시 서버의 ClientMessageProcess 모듈과 유사한 역할을 하며, 프록시 서버에서 TCPMessageProxy모듈로 수신된 INVITE, MESSAGE와 같은 SIP 메시지들에 대한 필드 정보를 저장한다. 이 정보들은 MessageProcess에서 메시지들을 구성하기 위해 필요로 한다.

3.2.1.6 MessageProcess

MessageProcess 모듈은 MProxy에서 넘겨받은 각각의 요청 메시지들과 필드 정보를 통해 ACK, BYE 등과 같은 응답 메시지들을 구성하고, 요청 메시지 처리를 위한 handleInvite, handleMessage 등과 같은 기능별 함수들로 구성되어 있다.

3.2.1.7 SIPMessageSend

MessageProcess모듈에서 구성된 응답 메시지들을 SIPMessageSend 모듈에서 프록시 서버 혹은 프레젼스 서버로 전송하는 역할을 한다.

3.2.2 프레젼스 서버 구조

IMPS 서비스를 제공하는데 있어 가장 중요한 역할을 하는 프레젼스 서버 시스템 세부 구성은 그림 9와 같이 4개의 기능모듈 InitPresenceServer, MessageProcess, SipRegistrar, SIPMessageSend와 메시지를 임시 저장

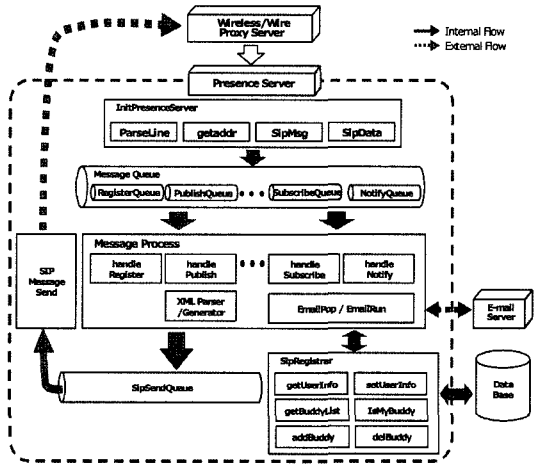


그림 9 프레젼스 서버 시스템 세부 구성도

하기 위한 SipSendQueue, MessageQueue로 구성되어 있다. 4개의 기능 모듈은 클래스 및 함수로 구성되어 있다. 또한, 사용자 정보 및 상태 정보를 저장하는 데이터베이스가 존재한다. 프레젼스 정보를 표현하기 위해서는 OMA에서 정의한 프레젼스 XML 스키마를 사용한다.

3.2.2.1 InitPresenceServer

서버간의 메시지 송수신을 하기 위해서 메시지 프록시 서버와 TCP 기반으로 연결을 하고 프레젼스 서버와 UDP 기반으로 연결을 한다. 이 모듈에서는 각 서버들의 연결을 설정하고, 메시지를 수신해서 Message Process 모듈로 넘겨준다. 이 때 수신된 요청 및 응답 메시지들을 처리하기 위해 스레드를 생성시킨다.

3.2.2.2 MessageProcess

MessageProcess모듈은 프록시 서버로부터 송신된 각 메시지를 처리하기 위해 기능 함수들의 스레드를 생성한다. handleNotify, handleSubscribe 등과 같은 기능 함수들은 MessageQueue의 각 버퍼에 메시지 속성별로 구분 되어진 메시지들을 전달 받고 응답할 메시지들을 구성한다. 그리고 MessageQueue에서 저장된 메시지들을 바탕으로 SIP Registrar 모듈과 연동을 통해 프레젼스 정보를 데이터베이스에 저장을 하거나, 또 다른 메시지들이 구성할 수 있다. 이 때, XML Generator/Parser 모듈에서 프레젼스 정보들을 XML로 구성하거나 파싱한다. 데이터베이스에 저장된 프레젼스 정보들은 XML형태로 구성되고 NOTIFY 메시지에 추가되어 단말기로 전송된다.

EmailPop 클래스는 사용자가 처음 서버에 접속을 했을 때 기존의 데이터베이스에 등록된 사용자의 이메일 계정과 암호를 통해 외부 이메일 서버와 접속을 하고, 이메일 서버로부터 새로운 메일이 왔음을 수신한다. 그

리고 이 클래스에서 다시 새로운 메일이 수신되었음을 단말기에게 알려준다. 이 때, POP3 (Post Office Protocol version 3)정보를 적재한 NOTIFY 메시지를 사용하여 주기적으로 알려준다.

EmailRun 클래스는 사용자가 새로운 메일 정보를 요청할 경우에 POP3정보를 적재하고 있는 SUBSCRIBE 요청 메시지를 이동 단말기로부터 수신 받고, POP3를 사용하여 외부 이메일 서버에 접속을 통해 읽지 않은 메일 리스트를 수신한다. 그리고 이 클래스에서 메일 정보를 적재하고 있는 NOTIFY 메시지를 이동 단말기에게 송신함으로써 사용자는 메일을 확인할 수 있다.

3.2.2.3 SIPRegistrar와 데이터베이스

SIPRegistrar 모듈은 Message Process 모듈에서 받은 메시지들을 데이터베이스에 접근해서 사용자들의 등록 여부를 판단하고, 사용자의 버디 리스트와 프레전스 정보를 불러오는 역할을 한다. 이 모듈에 포함된 getUserInfo(), setUserInfo()는 사용자에 대한 ID, 이메일 및 상태 정보를, getBuddyList(), isMyBuddy(), addBuddy(), delBuddy()는 사용자의 버디 리스트나 버디 추가 및 삭제 그리고, 버디 상태 정보를 추출하거나 저장하는 역할을 한다.

데이터베이스는 사용자 정보와 모든 프레전스 정보를 저장하고 있다. 그림 10은 데이터베이스 구조를 나타내고 있다. 구현된 데이터베이스 시스템은 mySQL을 사용하였다. 데이터베이스는 단말기와 서버간의 연결을 통해 사용자 정보와 이메일 계정을 등록을 하고, 사용자의 버디 리스트와 상태정보를 추가 삭제할 수 있다. 또한 다자간의 인스턴트 메시징을 위한 그룹 정보들을 저장하게 된다.

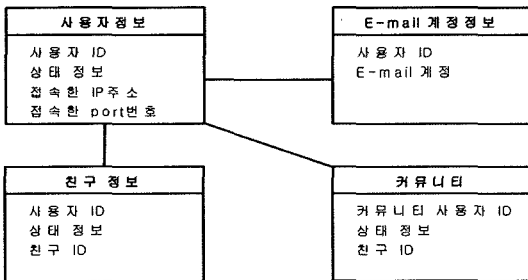


그림 10 데이터베이스 스키마

3.2.2.4 SIPMessageSend

프레전스 정보를 전달하기 위해서 MessageProcess 모듈에서 SipSendQueue에 메시지를 저장하게 되면 SipSendQueue는 SIPMessageSend 모듈을 이용하여 전송하게 된다. 이 SIPMessageSend 모듈은 단말기와 연결된 프록시 서버의 IP주소와 포트로 NOTIFY, PUB-

LISH 등과 같은 메시지들을 전송한다. 이를 통하여 손쉽고 원활한 통신을 지원한다.

3.3 IMPS서비스를 위한 SIP 클라이언트 구현

그림 11은 실제 PDA, 핸드폰과 PC 상에서 구현된 클라이언트를 보여 주고 있다. IMPS 클라이언트는 PC 상에서는 MS VC++ 6.0 MFC를 기반으로 개발 되었고 핸드폰에서는 BREW 1.03 버전에서 개발 되었다. PDA에서는 Embedded VC++ 4.0 환경에서 개발 되었다. 클라이언트는 디바이스 별로 다른 환경에서 개발 되었지만 주요 모듈은 같은 구조로 설계 되었다. 그림 12는 클라이언트 시스템의 주요 함수 구성도를 보여 주고 있다.

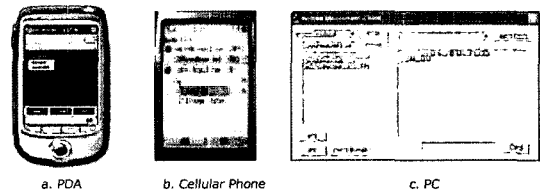


그림 11 IMPS 클라이언트 시스템 구성

3.3.1 SIP Stack

SIP Stack은 실질적인 메시지를 생성하는 역할을 한다. 상위 trans\_manage모듈로 부터 받은 데이터를 SIP Format에 맞게 구성한 패킷에 적재하여 전송하거나 전송 받은 패킷을 파싱하여 필요한 정보를 추출해 상위 레이어로 전달한다. SIP Parser() 함수는 핵심 모듈 중 하나로서 하위 네트워크 레이어에서 데이터를 입력으로 받는다. SIP\_Parser() 함수는 Get\_header() 함수를 호출하여 메시지의 첫 번째 header 스트링을 분리하고 메시지 타입을 판별한다. 판별된 메시지 타입에 따라 각 메시지별 파서인 Parser\_Invite(), Parser\_Message() 등의 함수를 호출하여 나머지 부분을 파싱한다.

마지막 헤더는 Length 헤더로 고정되어 있다. Length 헤더 다음에는 XML\_Parser() 혹은 SDP\_Parser()를 호출하여 페이로드에 있는 데이터를 파싱한다. 각 메시지별 파서 함수는 Get\_header() 함수를 호출하여 메시지를 다시 header 단위로 분리하고 필요한 정보를 추출한다. Get\_header() 함수는 메시지를 header 단위로 분리하는 역할을 담당하는 모듈이다.

SIP\_Maker() 함수는 상위 trans\_manage 모듈로부터 받은 데이터를 SIP 메시지로 구성한다. 각 애플리케이션에 의해 설정된 값에 따라 생성할 메시지 타입을 결정하고 Make\_Invite(), Make\_Subscribe()등과 같은 각 메시지 생성 모듈을 호출하고 이때 trans\_manage모듈로부터 전달 받은 데이터를 파라미터로 넘겨준다. 각 메시지 생성 모듈에 의해 생성된 메시지에 따라 SDP\_



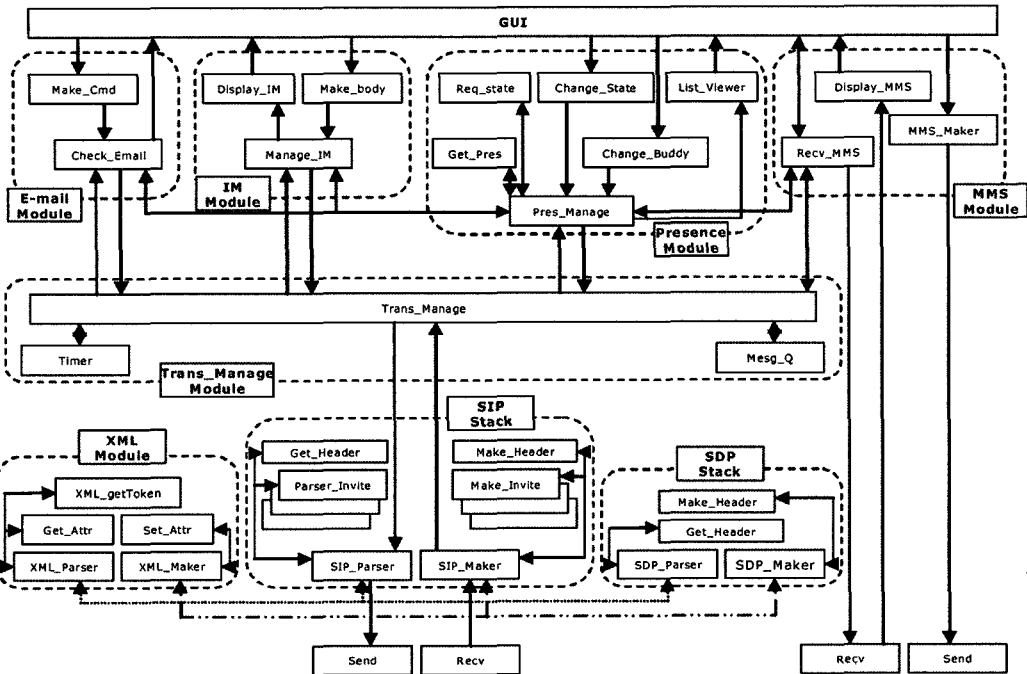


그림 12 IMPS 클라이언트 구조

Maker() 혹은 XML\_Maker()가 호출되어 페이로드를 생성한다. 각 메시지 생성 모듈은 SIP\_Maker()로부터 받은 데이터를 참조하여 각 메시지를 포맷에 맞게 생성한다.

3.3.2 SDP Stack

SDP Stack 모듈은 메시지에 포함되는 세션 정보를 생성 또는 파싱한다. SDP\_Parser() 함수는 전달 받은 INVITE 메시지에서 세션 정보를 추출하여 메시지 콘텐츠 타입의 판별이나 대화 상대의 주소를 추출하는 역할을 담당한다. Get\_header() 함수를 호출하여 메시지를 header단위로 분리하고 각 header를 파싱하여 정보를 추출해낸다. SDP\_Maker() 함수는 메시지 전송 시 페이로드로 전달될 SDP를 생성한다. 이미 설정된 단말의 IP 주소나 전달될 메시지 콘텐츠 타입을 SDP 포맷에 맞게 SDP 메시지를 생성한다.

3.3.3 XML

XML 모듈은 SIP 페이로드에 적재된 프레젠테이션 정보를 파싱, 저장하고 PUBLISH 메시지에 적재될 프레젠테이션 정보를 표현한다. XML\_Parser() 함수는 Notify 메시지의 페이로드로 오는 XML구문을 파싱한다. SIP Stack으로부터 전달 받은 스트링의 XML 애트리뷰트를 판별하고 Get\_Attr() 함수를 호출하여 필요한 엘리먼트의 값을 추출한다. Get\_Attr() 함수는 해당 애트리뷰트에 따라 XML\_getToken() 함수를 이용하여 엘리먼트의

값을 추출하고 이를 저장한다. XML\_Maker() 함수는 PUBLISH 메시지에서 사용자에게 의해 지정된 상태 정보를 XML 구문에 맞게 생성하여 SIP 메시지에 적재한다.

3.3.4 Trans\_Manage

Trans\_Manage 모듈은 메시지의 확실한 트랜잭션을 확립한다. Trans\_Manage() 함수는 송신한 메시지의 응답이 타임아웃이 될 때까지 도착하지 않으면 재전송하고 도착한 메시지에 대한 200OK 응답 메시지를 보낸다. 이때 MESSAGE 메시지는 제외한다. 멀티미디어 전송에서 Recv\_MMS() 함수에게 INVITE 메시지의 SDP 데이터를 보내고 리턴 되는 값에 근거하여 200OK 응답 메시지를 대화 상대에게 보낸다. Timer() 함수는 메시지를 전송 시에 설정 되어 응답 메시지가 오면 취소된다. 메시지가 도착하지 않았을 시에는 Mesg\_Q() 함수에서 해당 Cseq 넘버를 찾아 저장된 메시지를 재전송하도록 한다. Mesg\_Q() 함수는 전송되는 메시지 스트링을 Cseq 넘버와 함께 저장하고 검색하여 해당 위치를 반환한다.

3.3.5 Instant Message Management

Instant Message Management 모듈은 사용자로부터 입력 받은 내용을 즉각적으로 상대방에게 전송하고 수신한 메시지를 사용자에게 표현해 주는 기능을 가진다. Make\_body() 함수는 사용자로부터 입력을 받은 단문을 하위 trans\_manage 모듈에 넘겨준다. Display\_IM() 함

수는 전달 받은 메시지를 사용자에게 표현하는 역할을 담당한다. Manage\_IM() 함수는 Make\_body() 함수에 의해 생성된 메시지 내용과 MESSAGE 메시지를 구성하기 위해 필요한 정보를 설정하여 하위 trans\_manage 모듈에 전달한다. 또 수신한 메시지가 있을 때는 Presence Management 모듈의 Req\_state() 함수를 호출하여 사용자의 현재 상태를 확인한다. 이때 현재 상태가 대기상태일 때는 새로운 대화 창을 생성하고 다른 사용자와의 대화중일 때는 대화 창에 단순히 수신한 메시지를 전송한 사용자에 대한 정보와 함께 추가해 준다.

그림 13은 인스턴트 메시지 모듈을 구동 시켜 대화가 이루어지는 화면이다. 버디 리스트 중 대화상대를 선택하여 Send\_IM 버튼을 누르면 Make\_body() 함수가 호출되고 이어 Manage\_IM() 함수가 호출되어 인스턴트 메시지가 생성된다. 나머지 하위의 모듈들을 거쳐 메시지는 상대방에게 전달된다.

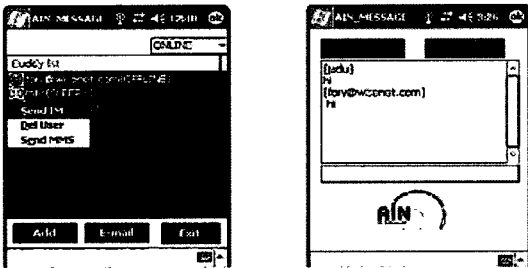


그림 13 인스턴트 메시지 시스템

3.3.6 E-mail Management

E-mail Management 모듈은 사용자의 이동 단말에서 자신의 메일 계정으로 전송되는 이메일을 확인 할 수 있는 기능을 제공한다. 이 모듈은 클라이언트가 최초로 활성화 될 때 서버로부터 수신한 이메일 리스트를 사용자에게 표현한다. 추가적으로 사용자의 요구에 따라 Make\_Cmd() 함수는 POP3 명령을 생성하여 하위 trans\_manage 모듈에게 넘겨준다. Check\_Email() 함수는 서버로부터 수신된 이메일 관련 정보를 해석하여 리스트 메뉴로 표현할 것인지 메시지로 보여 줄 것인지를 판단하여 상위 GUI에게 넘겨준다.

그림 14는 최초 활성화시에 도착한 메시지 수를 나타내는 화면과 E-mail 버튼을 선택했을 시에 도착한 이메일의 리스트를 보여주는 화면이다. 사용자가 이메일에 대해서 추가적으로 정보를 요구할 시에 Make\_Cmd() 함수로 명령을 생성하여 서버로부터 수신된 이메일 리스트 정보를 나타낸다. 현재 시스템에서는 POP3를 이용하여 텍스트 메일로 변환하여 지원하고 있지만 MIME (Multipurpose Internet Mail Extensions) 기능을 추가

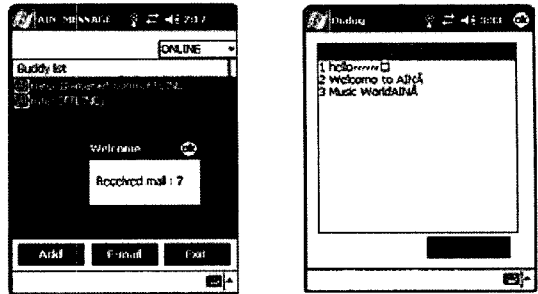


그림 14 이동 이메일 시스템

하여 멀티미디어 메일을 확장 지원 할 수도 있다.

3.3.7 Presence Management

Presence Management 모듈은 자신의 프레전스 정보를 변경하거나 버디를 등록 또는 삭제하는 등의 프레전스 정보를 관리한다. Change\_Stage() 함수는 사용자로부터 자신의 상태 정보 변경 요청이 있을 시에 이미 정의된 상태 정보 중 매칭 되는 상태를 검색하여 PUBLISH 메시지를 사용하여 서버에 전송한다. Change\_Buddy() 함수는 새로운 버디를 추가 하거나 기존의 버디 리스트에서 특정 버디를 삭제 시에 사용자가 지정하는 버디를 SUBSCRIBE 또는 UNSUBSCRIBE 메시지를 사용하여 추가, 삭제한다. Get\_Pres() 함수는 다른 모듈에서 버디 리스트의 프레전스 정보를 요청 시에 해당 프레전스 정보를 찾아 그 값을 넘겨준다. Req\_state() 함수는 사용자의 현재 상태 정보를 저장하고 다른 기능 모듈에서 현재 사용자의 상태 정보를 요청 시에 이를 반환한다.

3.3.8 MMS

MMS 모듈은 Signal 메시지 이외의 이미지나 사운드와 같은 멀티미디어 데이터의 전송을 담당한다. Recv\_MMS() 함수는 대화 상대방으로부터 INVITE 메시지가 도착하면 사용자로부터 허용 여부를 질의하고 수락 시 SIP Stack에서 사용하는 소켓과는 다른 별도의 소켓을 생성하고 Listen 대기 상태가 되고 동시에 대화 상대방에게 200OK 응답 메시지를 보내어 새로 생성된 소켓의 IP 주소와 포트번호로 멀티미디어 메시지를 보내도록 한다. MMS\_Maker() 함수는 멀티미디어 메시지를 전송할 상대방에게 먼저 INVITE 메시지를 보내어 먼저 세션을 확립한다. 상대방부터의 200OK 응답 메시지가 있을 시에는 페이로드에 실려 있는 IP 주소와 포트번호를 확인하고 사용자에 의해 선택 되어진 멀티미디어 데이터를 전송한다. Display\_MMS() 함수는 SDP에 의해 정의된 메시지 타입을 보고 재생기를 선택하여 재생한다.

그림 15는 PDA 클라이언트 송신측에서 선택하여 보낸 이미지를 수신측에서 수신하여 화면에 나타내는 화

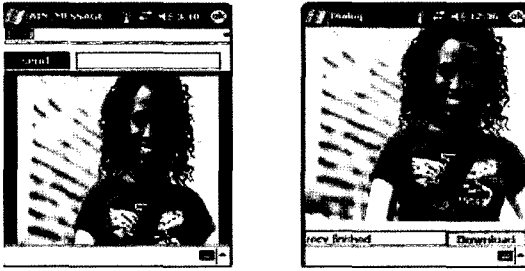


그림 15 MMS 메시지 전송 시스템

면이다. 전송할 이미지가 선택되면 송신측 클라이언트는 INVITE 메시지를 보내고, 수신측은 INVITE 메시지의 SDP 부분에서 송신측의 IP 주소와 포트 넘버를 확인하고 해당 주소로 접속하여 이미지를 수신 받게 된다. MMS 모듈은 현재 bmp, jpeg, gif 등의 이미지 전송 및 재생을 담당하고 있다. 추가적으로 오디오 재생기, 비디오 재생기를 탑재하여 화상 회의 등의 응용 프로그램으로 확장이 가능하다.

#### 4. 결론

본 논문에서는 IETF와 OMA의 프레젼스 서비스와 관련된 국제 표준화 동향 및 산업계 동향에 대하여 연구 하였다. 기존의 인스턴트 메신저에 국제 표준 프레젼스 서비스 기술을 적용하여 다른 인스턴트 메신저와의 상호 호환성을 보장할 수 있으며, 확장된 프레젼스 서비스를 제공할 수 있도록 설계 및 구현하였다. 확장된 프레젼스 서비스의 응용으로써 본 논문에서는 POP3 프로토콜을 이용하여 사용자 메일의 상태 정보를 나타내는 이메일 응용프로그램을 구현하였다. 또한 인스턴트 메시지와 멀티미디어 메시지 전송이 가능하도록 설계 및 구현하였다. 구현된 시스템은 각 기능 블록으로 모듈화 되어 새로운 응용 서비스를 추가하기에 용이하다.

본 논문에서 구현된 시스템을 기반으로 하여 Context-Aware 서비스, Location Based Service, Home Network Service 그리고 Ubiquitous 서비스 등의 다양한 응용 프로그램의 기반 기술이 될 수 있다. 정보를 표현하기 위한 기술은 프레젼스 서비스 중심으로 발전할 것으로 예상되고, 따라서 많은 산업 분야에서 사용되어 킬러 응용으로 사용될 전망이다. 이를 위하여 보다 안정성이 보장되어야 할 것이며 보안기능이 확대 되어야 할 것이다.

#### 참고 문헌

[1] M. Day, J. Rosenberg and H. Sugano, "A Model

for Presence and Instant Messaging," RFC 2778, Feb., 2000.

- [2] L. Rishi and S. Kumar, "Presence and its effect on network," *Personal Wireless Communications*, Jan. 2005, pp. 368-372.
- [3] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," RFC 3856, Aug., 2004.
- [4] Open Mobile Alliance. Open Mobile Alliance Principles [Online]. Available : <http://www.openmobilealliance.org/docs/OMA-principles.pdf>
- [5] "WV-049 Presence Attributes-Approved Version 1.2," OMA, Jan., 2005.
- [6] "WV-050 Presence Attribute DTD and Examples-Approved Version 1.2," Open Mobile Alliance, Jan., 2005.
- [7] "WV-051 Command Line Protocol-Approved Version 1.2," Open Mobile Alliance, Jan., 2005.
- [8] "WV-052 Server-Server Protocol Semantics Document - Approved Version 1.2," Open Mobile Alliance, Jan., 2005.
- [9] R. Parviainen and P. Parnes, "Mobile instant messaging," *Telecommunications, ICT 2003*, Feb., Mar., 2003, pp. 425-430.
- [10] M. Debbabi and M. Rahman, "The war of presence and instant messaging: right protocols and APIs," *Consumer Communications and Networking Conference*, Jan., 2004, pp. 341-346.
- [11] Vaughan-Nichols and S.J., "Presence technology: more than just instant messaging," *IEEE Computer*, Oct., 2003 pp. 11-13.
- [12] C. Faure, "Presence service in 3G networks," *3G Mobile Communication Technologies*, May, 2002, pp. 511-515.
- [13] Kanter, T.G., "Extensible mobile presence," *Mobile and Wireless Communications Network*, Sept., 2002, pp. 661-664.
- [14] J. Zhigang, J. Hui, and H. Lu, "Instant messaging and presence services using SIMPLE," *TENCON 2004*. Nov., 2004, pp. 157-159.
- [15] D. Greene and D. O'Mahony, "Instant messaging & presence management in mobile adhoc networks," *Pervasive Computing and Communications Workshops*, Mar., 2004, pp. 55-59.
- [16] M. Debbabi, M. Rahman and W. Lin, G. Jay, and S. Srikrishna "Standard SIP-based instant messaging and presence APIs for networked devices," *IETF Networked Appliances*, Oct. 2002, pp. 59-64.
- [17] D. Jiang, T. Yeap, L. Logrippo and R. Liscano, "Personalization for SIP Multimedia Communications with Presence," *International Conference on Services Systems and Services Management*, June 2005, pp. 1365-1368.
- [18] A. Kanamaru, Y. Tsuchiya, T. Fujisaki, and K. Takahashi, "Fieldcast2: flexible P2P architecture for presence information sharing," *Pervasive Com-*

puting and Communications Workshops, March 2004, pp. 98-103.

- [19] R Liscano, A Ghavam and M. Barbeau, "Integrating Service Discovery Protocol with Presence-based Communications for Ad hoc Collaborative Scenario," Communication Networks and Services Research, May 2004, pp. 357-361.
- [20] A. Moran, J Favela, A. Martinez, and D. Decouchant, "Document presence notification services for collaborative writing," Seventh International Workshop on Digital Object Identifier, Sept. 2001 pp. 125-133.



나재욱

2001년 경북대학교 농업경제학과(경제학사)/컴퓨터공학과(공학사). 2003년 경북대학교 정보통신학과(공학석사). 2003년~현재 경북대학교 정보통신학과 박사과정  
관심분야는 SIP 및 SIP 응용, RTLS (Real Time Locating System)



최진영

2002년 동서대학교 정보통신공학과(학사)  
2004년~현재 경북대학교 전자공학과 석사과정. 관심분야는 SIP, LBS



조현택

2004년 대구대학교 통계학과(학사). 2004년~현재 경북대학교 정보통신학과 석사과정. 관심분야는 RFID



김용훈

2004년 대구가톨릭대학교 컴퓨터공학과(학사). 2004년~현재 경북대학교 전자공학과 석사과정. 관심분야는 RFID



이진구

2001년 대전대학교 정보통신학과(학사)  
2003년 경북대학교 정보통신학과(석사)  
2003년~현재 경북대학교 정보통신공학과 박사과정. 관심분야는 무선 인터넷, 차세대 인터넷



박종태

1978년 경북대학교 전자공학과(공학사)  
1981년 서울대학교 전자공학과(공학석사). 1987년 미국 미시건 대학교 전기컴퓨터학과(공학박사). 1989년~현재 경북대학교 전자전기컴퓨터학부 교수. 2000년~2003년 IEEE Technical Committee on Information Infrastructure(TCII) 의장. 1987년 미국 AT&T Bell 연구소 연구위원. 1984년~1987년 미국 CITI (Center for Information Technology Integration) 연구원. 관심분야는 차세대 통신망 관리, 네트워크 이동성 관리, 무선 센서 네트워크 및 BcN 관리, SIP 기반의 모바일 멀티미디어 시스템 개발