

이종(異種) 모바일 단말 간 인스턴트 메신저 설계 및 구현

Design and Implementation of an Instant Messenger between the two different domain mobile terminal systems

방지웅*, 김대원*

Ji-woong Bang*, Dae-won Kim*

요약

최근 우리 사회는 정보화 시대를 거쳐 유비쿼터스 산업 시대의 도래로 인해 기존의 고정적인 네트워크 시스템에서 벗어나 이동 및 휴대성이 증대된 모바일 단말을 이용하는 통신 서비스 사용이 급증하는 양상을 띄고 있다. 모바일 단말의 다양화로 인해 이종 모바일 단말간의 통신이 어려워지고 있다. 사용되는 운영체제 및 하드웨어가 다른 이종 단말간의 데이터 통신을 위해서는 데이터 변환이 필요하다. 논문에서 제안한 인스턴트 메신저 설계 방식을 이종 단말기인 MRP-S3C2440A (Windows CE 5.0)과 Nexio XP30 (Windows CE .NET 4.1)에 적용하였다. 중앙 서버는 단말 정보를 갖고 있어 송신 단말의 메시지를 수신 단말의 특성에 맞게 변환한 뒤 수신 단말에게 전송하였다. 수신 측 단말은 별도의 변환 과정 없이 바로 단말기에 출력하였다. 논문에서는 중앙 서버를 이용한 이종 단말 간의 인스턴트 메신저 설계 방법을 제안하고 구현하였다.

Abstract

Recently, we are living in the world of information and ubiquitous technology strayed away from the fixed network systems resulting in an increase of using mobile terminals which is convenient for mobility and functionality. Because the diversification of mobile devices to communicate between disparate mobile terminals is difficult, the two kinds of operating systems and hardware for data communication between the terminals is needed to convert the data. The proposed instant messenger design in MRP-S3C2440A (Windows CE 5.0) and Nexio XP30 (Windows CE .NET 4.1) were tested and applied. Usually the mobile terminal has an information on a central server. Sender terminal receives the message according to the characteristics of the terminal after converting was sent to the receiving terminal. Receiving-side terminal without a separate conversion process, it was the output terminals. This paper deals with the proposal and implementation of designing an instant messenger application program between the two different mobile terminals and network systems.

Keywords : Mobile, Instant Messenger, Terminals, Client, Server

I. 서론

네트워킹은 현대인에게 있어 몸의 일부가 되어가고 있는 필수적 IT (Information Technology) 기술이다. 인터넷이라는 거대한 네트워크는 사람들로 하여금 수많은 정보를 편리하게 이용할 수 있도록 해준다. 또한 전 세계를 연결하는 하나의 수단이 이미 되었다. 이러한 인터넷이라는 네트워크를 사용하기 위해서는 기존의 데스크탑 PC 보유가 필수적으로 요구되었다. 하지만 IT 기술의 눈부신 발달로

인해 집이나 회사 뿐만 아니라 움직이면서 언제 어디서나 인터넷의 정보를 이용할 수 있는 시대가 도래 하였다. 예를 들자면 특정 통신사의 무선 인터넷 서비스나 WIBRO (Wireless Broadband) 서비스가 그것이다. 편리하게 들고 다닐 수 있어 인터넷과 다른 사람들과의 정보 교류에 알맞은 모바일 단말은 그 시장의 크기가 비약적으로 커져가고 있다. 이러한 추세는 여러 단말 장치나 시스템의 연결을 통하여 하나의 네트워크 서비스를 언제 어디서나 이용할 수 있는 유비쿼터스 컴퓨팅 환경으로 계속 진화하고 있다.

유비쿼터스 컴퓨팅 환경이란 각종 시스템과 여러 이동 장치들 간의 동적인 네트워크 구축을 통하여 사용자들이 이러한 시스템을 원하는 곳에서, 원하는 시간에 자유로이

* 단국대학교

투고일 : 2010. 2. 8 수정완료일 : 2010. 4. 27

게재확정일 : 2010. 10. 29

사용할 수 있는 환경을 말한다. [1] 이러한 환경은 고정적인 유선 네트워크와 무선 네트워크 방식의 통합으로 구축된다. 최근 들어 기존의 유선 네트워크에서 모바일 단말을 이용한 무선 네트워크를 사용하는 방식으로 그 정보의 흐름이 바뀌어 가고 있다. 현대인에게 있어 시간과 정보는 곧 가치 있는 재화로 인식되는 요즘 시대에 모바일 단말은 생활의 필수품이 되어가고 있다. 이러한 추세를 이야기하듯 어린아이부터 노인에게 이르기까지 사회의 다양한 연령층에서 모바일 단말을 사용하고 있다. 우리나라 전 국민의 대다수인 약 90%정도가 셀룰러 폰을 가지고 있다는 사실은 모바일 시장이 얼마나 비약적으로 발전 하였는가를 보여준다. [2] 그 중 무선 네트워크의 중요한 요소를 차지하는 단말은 그 기능이 하드웨어적으로 특성화되어 필요한 기능에 따라 다양한 제품이 시장에 출시되고 있다. 구매자의 기능적인 욕구에 따라 시장은 매년 최신 기술이나 혹은 특성화에 필요한 기술들만을 단말에 이식하여 판매하고 있어 이른바 단말 기종의 춘추전국시대를 맞고 있다. 이러한 단말들은 보통 인터넷 및 사용자의 개인정보 그리고 문서 저장 공간, 멀티미디어, 엔터테인먼트, 오락 등을 위해 활용될 뿐만 아니라 가족, 친구 간 또는 직장 동료등과 같은 사람들과의 의사소통을 위한 통신에도 사용된다. 대표적인 예가 바로 모바일 인스턴트 메신저이다. 모바일 인스턴트 메신저란 모바일 단말을 사용하여 실시간으로 자신의 상태를 표시하면서 통신을 원하는 상대방과 단말기를 통하여 대화 및 파일 전송 등과 같은 작업을 수행하는 메신저 서비스를 말한다. [3] 이러한 메신저 서비스는 기존 데스크탑 PC 간의 유선 네트워크를 통해 먼저 활성화되었는데 최근 이동성이 편리한 모바일 단말을 활용한 모바일 메신저 서비스의 이용이 증대되고 있다. 현재 사용되는 셀룰러 폰을 이용한 모바일 메신저로는 KT와 SKT 통신사에서 개발한 'KT 모바일 메신저', 'SKT 모바일 메신저'와 (주)버디버디에서 개발한 '버디버디 모바일 메신저', 가입한 이동통신사에 상관없이 최대 4명까지 채팅이 가능한 온세텔레콤의 '친친프리' 등이 있다. 또한 PDA를 활용한 모바일 메신저로는 SK 커뮤니케이션즈사에서 개발한 '네이트온'과 국외의 경우 윈도우 CE OS에서 사용가능하며 마이크로소프트사에서 개발한 'MSN 메신저'와 '야후 메신저', 팜 OS에서 사용가능한 'ICQ' 등이 있다. [4] 주목할만한 점은 기술적인 발달 및 모바일 단말 사용자의 기능적인 욕구가 증대되는 상황에서 인터넷과 이를 이용한 다양한 응용 서비스가 가능한 PDA와 스마트 폰의 사용이 증가하고 있다는 점이다. 이는 단순한 전화통화 기능을 벗어나 다양한 서비스와 멀티미디어 콘텐츠를 활용할 수 있는 PDA 또는 스마트 폰은 사용자에게 매우 매력적인 단말이기 때문이다. 이러한 PDA와 스마트 폰의 사용자가 증가하면서 다양한 제품이 생산되고 또한 거기에 탑재되어 있는 운영체제 또한 다양해지고 있다.

미국 시장조사업체 가트너의 보고서에 따르면 2008년도

PDA 운영체제 시장을 조사한 결과 마이크로소프트사의 윈도우 CE 계열 운영체제가 전체 시장의 약 48%, 팜스사의 팜 계열 운영체제가 30%를 점유하고 나머지 22% 정도를 Psion사의 EPOC, 국내 PDA전문 업체인 제이텔의 셀빅 OS, 임베디드 리눅스가 차지하고 있다. PDA의 다양한 운영체제가 공존함에 따라 이들 운영체제를 사용하고 있는 PDA간의 통신에 제약이 발생하게 된다. 특성상 하드웨어적인 요소가 탑재되는 운영체제에 영향을 미치게 된다. 동일 기능을 가진 제품도 탑재되어 있는 하드웨어의 특성에 따라 서로 다른 운영체제를 사용하게 된다. 단말기의 수요가 증가함에 따라 그 기능은 점차 향상되었지만, 휴대성 구조로 인해 무선 지원 환경이 제한적이다. 이러한 모바일 시장 상황으로 인해 모바일 소프트웨어 설계에 있어 많은 상황을 고려를 해야한다. PDA 단말기 제조사와 동일한 운영체제 시리즈를 탑재한다 하더라도 운영체제 버전의 차이를 극복해야한다. 기존의 모바일 기반 인스턴트 메신저들은 이종 단말의 특성을 고려하지 않고 서비스하는 경우가 많다. 문제점들을 해결하기 위해서 모바일 기기 간의 상호 약속된 프로토콜 혹은 단문 메시지 시스템, TLV 등을 사용하여 문제를 해결하는 방법이 있다.[5][6] 이러한 해결책은 근본적으로 여러 이종 단말간의 서비스보다는 운영체제가 같은 단말에서 사용되는 기술들이 대부분이다. 이종 단말간의 네트워크 및 멀티미디어 응용 서비스 등과 같은 통합적인 서비스를 지향하는 모바일 인스턴트 메신저를 구현하기 위해서는 많은 제약이 발생할 수 있다. 이러한 문제점을 해결하기 위해서는 이종 단말간의 인스턴트 메신저 서비스를 위해서는 통신 데이터의 변환이 반드시 필요하다. 이러한 변환을 단말 기기에서 수행하는 것 보다는 단말 정보를 갖고 있는 중앙 서버에서 수행됨으로써 단말 기기간의 효율적인 인스턴트 메신저 서비스가 가능하다.[7][8] 본 논문에서는 다양한 단말과 운영체제 간의 제약을 없애고 단말 기종에 무관하게 동작하는 모바일 인스턴트 메신저 용 클라이언트와 서버 설계 방법을 제안하고 구현하였다.

본 논문에서는 2장에서 구현에 필요한 모바일 운영체제와 단말 환경에 대해서 설명하고 3장에서는 인스턴트 메신저 설계에 대하여 묘사하였으며 4장에서는 인스턴트 메신저 클라이언트와 서버의 구현 방법에 대하여 설명하고 마지막으로 5장에서 결론 및 향후 연구 방향에 대하여 설명하였다.

II. 모바일 운영체제(OS) 및 단말 개발 환경

2.1 윈도우 CE

1996년에 발표된 마이크로소프트의 윈도우 CE는 PC가 아닌 장치들을 위한 새로운 임베디드 OS로서 터미널, 휴대용 컴퓨터, 산업용 제어기 및 소형 PDA 단말부터 웹폰, 인터넷 TV에까지 그 활용 범위가 계속 확장되고 있다. 윈도우 CE는 다른 운영체제와 같이 다양한 버전이 계속 업

그레이드 되어 출시 되었는데 초기 윈도우 CE 1.0 부터 가장 최근의 윈도우 임베디드 CE 6.0까지 나온 상태이다. 윈도우 CE 라는 명칭은 커널 이름을 가리키는 것인데 다양한 디바이스 및 시스템에 적용하기 위해서 개발되었기 때문에 윈도우 CE 운영체제가 탑재되는 하드웨어의 특성에 맞게 이름이 서로 다르게 존재한다. 즉 윈도우 CE 2.12 보다 윈도우 CE 2.11이 먼저 출시된 것이 아니라 적용되는 하드웨어의 사양이 서로 다르다는 것을 의미한다. 윈도우 CE가 탑재된 대표적인 제품들로는 사이버뱅크사의 '포즈'와 휴렛팩커드사의 '아이팩'이 있다. [9]

마이크로소프트 사의 대표적인 단말 운영체제로는 윈도우 임베디드 CE에 덧붙여 윈도우 모바일이 있다. 윈도우 임베디드 CE는 매우 정확하게 요구된 시간 안에 응답을 가능케하는 이른바 '하드 리얼 타임' 기반의 운영체제이며 약 700 여개의 컴포넌트가 제공된다. 따라서 임베디드 영역에서 개발자의 요구에 따른 다양한 형태의 운영체제를 개발 할 수 있는데 컴포넌트의 예를 들어, 개발자가 커널만 있는 운영체제를 제작하거나 네트워크 컴포넌트를 이용하여 웹 브라우저만을 내장한 운영체제를 개발할 수도 있다. 필요에 따라 컴포넌트의 조합을 통해 운영체제가 제작되기 때문에 개발자는 운영체제에 포함된 컴포넌트와 그에 따른 API를 공개해야한다. 이를 이용하여 타 기업의 응용 프로그램이 그 운영체제에서 정상적인 동작을 할 수 있다. 또한 윈도우 임베디드 CE의 단점으로는 각 통신사에게 배포한 운영체제에 대한 관리 및 유지의 책임이 있으며 공통적인 API를 제공하지 못하기 때문에 개발자에게는 개발에 어려움이 있을 수도 있다.

표 1. 윈도우 CE 5.0과 윈도우 임베디드 CE 6.0의 비교
Table 1. Comparison of Windows CE 5.0 and Windows embedded CE 6.0

	윈도우 CE 5.0	윈도우 임베디드 CE 6.0
라이선스 모델	Core Professional Professional plus	Core Professional
개발 툴	Platform Builder 5.0	Visual Studio 2005 통합
프로세스 수	32개 프로세스 지원	32,000개 프로세스 지원
가상메모리	프로세스 당 32MB	프로세스 당 2GB
BIB, REG 에디터	제공하지 않음	에디터 제공
CE component 제공	CEC 형태	XML 형태
NMD	별도 제공	포함되어 있음
Feature Pack DRM1.0	별도 제공	포함되어 있음

윈도우 모바일은 윈도우 임베디드 CE와 이를 포함하여 개발되는 제품의 특성에 따라 완전히 독립된 하나의 컴포넌트 집합으로 개발된다. 즉, 시스템 개발자들이 운영체제 시스템 레벨에서의 변경 및 추가가 불가능하므로 윈

도우 모바일에서는 공통적인 API를 제공하며 이를 개발자가 이용하면 이종 컴파일 환경인 크로스 플랫폼에서도 동작한다. 윈도우 모바일의 관련 개발 툴인 윈도우 모바일 SDK의 지원 및 관리에 대한 책임은 마이크로소프트사가 지고 있다. [10]

윈도우 임베디드 CE 6.0은 기존 윈도우 CE 계열의 운영체제에 비해 개발자가 작업을 하기에 편리하다는 점과 마이크로소프트 사의 소스 공유 정책에 따라 많은 부분 공개되어 소프트웨어 개발에 용이하다는 이점이 있다. 또한 '비주얼 스튜디오 2005'에 통합 되어 개발 비용과 시장 진입 시간이 단축되고 차세대 커널 기술과 확장된 메모리 기술 등 다수를 포함하고 있다는 장점이 있다. [표 1]은 기존의 단말 OS 환경인 윈도우 CE 5.0과 임베디드 6.0의 사양을 비교한 것이다. [표 1]에서 볼 수 있듯이 기존 윈도우 CE 5.0용 응용 프로그램을 개발하기 위해서는 플랫폼 빌더 5.0을 사용해야한다. 윈도우 임베디드 6.0으로 버전이 업그레이드 되면서 기존 플랫폼 빌더 5.0을 사용한 독립적인 개발 방식 대신 '비주얼 스튜디오 2005'에 통합되어 개발자 위주의 통합적인 프로젝트 진행이 가능하게 되었다. [그림 1]은 윈도우 CE의 기본 구조를 보이고 있다.

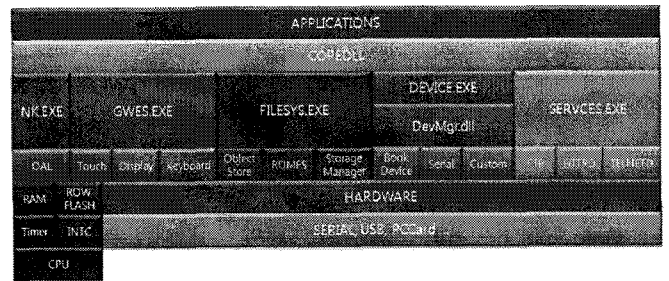


그림 1. 윈도우 CE 구조

Figure. 1. Structure of Windows CE

[그림 1]에서 볼 수 있듯이 하드웨어적 기반 위에 모듈 형식의 커널이 존재하고 그 위의 계층에 다양한 응용 프로그램과 셸을 얹어 색다른 기능을 사용할 수 있다. [11] 또한 디스크 저장 장치를 사용하지 않고 설정이 가능하며, 커널은 1 메가 바이트 이하의 메모리에서도 동작할 수 있도록 설계 되어 있다. 이는 작은 저장 공간의 장치에서 사용하기에 최적의 환경을 제공한다. 윈도우 CE에서의 용이한 개발 환경 구축을 위하여 하드웨어 업체들에게 소스 코드가 제공되어져 있다. 이를 기반으로 한 윈도우 CE 운영 체제 이미지 및 사용자 정의 도구인 플랫폼 빌더 같은 제품이 소스 코드를 포함한 형태로 제공되고 있고 이는 윈도우 CE 기반의 자유로운 단말 응용 프로그램 개발에 유용하다고 할 수 있다.

2.2 단말 환경

PDA 보드를 사용하는 CPU에 따라 하드웨어를 초기화하는 어셈블러 파일에서부터 부트 스트랩, 운영체제 초기화, 사용자 모듈 등이 달라진다. 먼저 운영 체제가 동작할

수 있도록 하기 위해서는 BSP (Board Support Package)를 설치해야 한다. BSP로 SDB (Standard Development Board)에서의 운영체제를 실행하고 지원하기 위한 소프트웨어 패키지에는 개발 도중 운영체제 이미지를 다운로드 하는데 사용되는 부트 로더(boot loader), 커널 이미지에 연결되어 하드웨어 초기화 및 관리를 지원하는 OAL (OEM Adaptation Layer), 런 타임에 '온 보드' 상태이거나 또는 연결된 주변 장치를 지원하기 위한 장치 드라이버, 그리고 환경변수와 .bib 및 .reg 파일 수정을 통해 BSP를 재구성하기 위한 환경 파일 등이 있다. BSP는 PDA의 CPU에 따라서 다르게 제공되는데, 본 연구에 사용된 PDA 보드에 설치된 CPU는 ARM 사의 MRP-S3C2440A로 삼성 SMDK 2410이 사용되었다. [그림 2]는 PDA 보드를 다이어그램으로 나타낸 것으로서 보드의 기본적인 구성요소를 보이고 있다. [12]

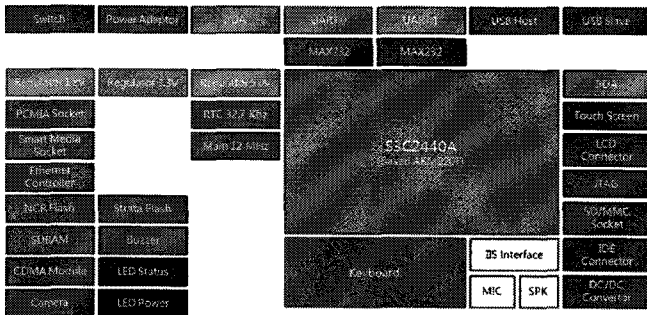


그림 2. PDA(MRP-S3C2440A) 보드 다이어그램
Figure. 2. Board diagram of PDA (MRP-S3C2440A)

본 연구에서는 PC상에서 서버의 역할을 할 수 있도록 윈도우즈 CE의 OS를 구성한다. 디바이스 드라이버 및 응용 프로그램 개발, OS 디버깅 및 각종 테스트를 할 수 있는 GUI (Graphic User Interface) 기반의 통합 개발환경인 플랫폼 빌더를 이용하여 OS의 구성요소들을 선택하였다. PDA에 맞는 장치 드라이버들을 빌드하여 윈도우즈 CE 5.0 환경을 구성한 후 이를 PDA에 포팅 하였다. 클라이언트로서 무선 단말 환경으로 사용된 넥시오는 한글 윈도우즈 CE .NET 4.1 및 PXA255 400MHz의 CPU를 탑재하여 상용화 된 제품으로 무선 랜(802.11b, Wi-Fi) 환경이 내장되어 있다. 네트워크 환경을 제공하기 위한 PDA의 인터넷 환경은 유선 랜을 통해 이루어지도록 하고, 넥시오 단말은 KT가 제공하는 무선 상용화 서비스인 네스팟을 통해 무선 인터넷을 사용하였다.

III. 통신 응용 시스템의 설계

본 연구에서 구현된 모바일 인스턴트 메시저는 일반적인 메시저 구조에서 많이 사용하는 TCP/IP 기반의 클라이언트-서버 구조를 차용하였다. 이는 본 연구를 위한 시스템에서 사용하는 PDA, 넥시오 단말, 그리고 서버 컴퓨터 모두가 TCP/IP를 기반으로 하는 네트워크 통신 시스템 사용

이 가능하기 때문이다. 이 통신 응용 시스템에서 사용되는 PDA는 윈도우즈 CE 5.0을 기반으로 하고 무선 단말인 넥시오는 윈도우즈 CE .NET 4.1을 사용한다. 서버는 일반적인 윈도우즈 XP 기반의 32비트 컴퓨터 상에 구현하였으며 동시 접속은 컴명까지로 제한을 았 구현하클라이언트가 송신한 데이터를 수신한 서버는 특정 목적지로의 전송이 아닌, 통신 응용 시스템을 시험하기 위해 접속한 모든 대상으로 데이터를 전달하는 방식인 브로드캐스팅 방식을 사용했 무 이는 본 연구가 단말기 간의 소켓통신 구현을 목적으로 진행한 것이기 때문이다. 무선 랜 기능 미 그리므로 인하여 PDA는 유선 인터넷 환경에서 네트워크 소켓통신을 사용하였다. 넥시오는 KT가 제공하는 상용화 서비스인 네스팟이 가능한 지역에서 무선 인터넷 응용 상 통신을 한다. PDA의 무선 인터넷을 통한 단말기 간 통신은 넥시오의 무선 인터넷을 통한 통신 성공을 전제로 한다. 하였으며 3]은 본 연구에서 진행된 PDA의 개발환경과 통신 프로그램간의 개략적인 관계 및 설계 모델을 나타내고 있다.

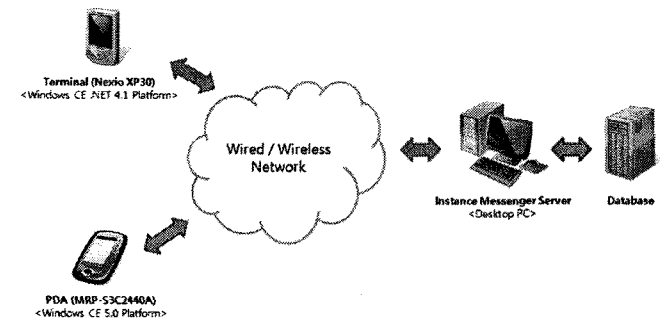


그림 3. 모바일 인스턴트 Messenger 구축 모델
Figure. 3. Construction model of mobile instant messenger

[그림 3]의 PDA와 무선 단말인 넥시오 장치에 삽입되는 클라이언트 프로그램은 일반 데스크탑 컴퓨터에서 개발된 후 각각 PDA (MRP-S3C2440A) 보드와 넥시오에 포팅 되었다. 그 후 서버 프로그램이 탑재되어 있는 인스턴트 메시저 서버에 접속을 하기 위해서 각각의 클라이언트 프로그램은 서버의 CM (Connection Manager)에 접속한다. 이 후 서버는 수신 받은 메시지를 단말과 PDA의 환경적인 문제점과 하드웨어적 기능 지원 등과 같은 다양한 요소들의 차이점을 극복하기 위해 CMM (Converting Message Manager)에 전송한 후 서로 다른 이종 모바일 클라이언트의 환경에 맞는 메시지를 전송한다. 모바일 인스턴트 메시저 서버는 각각의 단말과 PDA에서 발생하는 플랫폼의 차이점과 하드웨어적 기능의 차이, 같은 하드웨어를 기반으로 했을지라도 버전의 차이에서 오는 문제점 등 다양한 이슈들을 해결하기 위해 모바일 기기에 맞게 메시지를 변화 시키고 캡슐화 하는 방법을 채택하고 있다. [그림 4]는 이러한 메시지를 변환하기 위한 서버의 아키텍처이다. [그림 4]에 인스턴트 메시저의 서버와 그 구조를 나타낸 바와 같이 전체적인 구조는 클라이언

트 연결을 담당하는 CM과 이들 각각의 클라이언트들이 전송하는 메시지를 변환하기 위한 CMM 두 부분으로 구성되었다. CM은 네트워크 소켓, authority 모듈, 터미널 매핑, 메시지 변환기 등으로 구성되었고, 네트워크 소켓은 서버가 클라이언트의 접속을 대기하는 listen 소켓으로 구성되었다. 모바일 단말 사용자가 클라이언트 프로그램을 사용하여 유선 또는 무선의 네트워크를 통해 서버에 연결할 경우 네트워크 소켓은 접속된 클라이언트의 접속 정보를 authority 모듈에 넘겨주고 authority 모듈은 클라이언트에게 사용자 ID와 비밀번호를 요청한다.

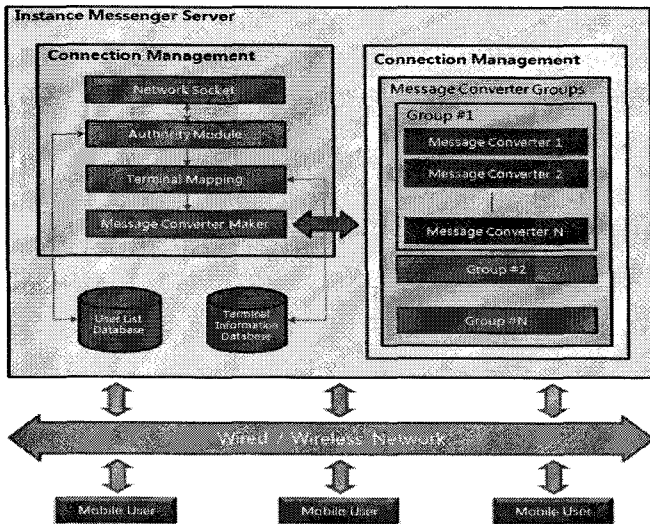


그림 4. 인스턴트 Messenger Server 아키텍처
Figure. 4. Architecture of instant messenger server

이러한 요청을 받은 클라이언트는 사용자 정보를 서버에 전송하고 authority 모듈은 ULD (User List Database)와 통신하여 이들 사용자의 정보를 검사하고 인증된 사용자가 아닌 경우 서버 등록 실패 메시지를 전송한다. 만약 인증된 사용자인 경우 authority 모듈은 소켓 정보와 사용자의 ID 정보를 터미널 매핑에 넘겨주는데 터미널 매핑은 접속된 클라이언트 모바일 기기의 환경 정보가 저장된 TID (Terminal Information Database)에 해당 클라이언트의 정보를 요청한다. 이때 해당 정보가 없을 경우 클라이언트에게 메시지 변환에 필요한, 하드웨어적으로 지정된 몇 개의 정보와 플랫폼 정보를 수집하도록 요청하고 사용자 ID와 함께 저장한다. 이후 이들 정보는 단말 간의 메시지 변환에 사용된다. 터미널 매핑에서 이러한 정보가 수집 완료되면 접속된 클라이언트를 위한 메시지 변환기를 생성하기 위해 MCM (Message Converter Maker)에 정보를 전송하게 되는데. MCM은 수집된 정보를 가지고 접속된 클라이언트를 위한 그룹을 생성한다. 이 그룹은 메시지 변환기 그룹이라고 불리는데, 하나의 대화 단위로서 여러 사용자간의 그룹 채팅을 위한 모임이다. 하나의 그룹에는 2명 이상의 클라이언트가 구성될 수 있는데 이들은 모두 같은 대화에 참여할 수 있다. 그룹은 생성 순서에 따른 관리를 위한 번

호가 매겨지는데, 예를 들어 새로이 생성된 그룹은 #1, #2 ... #N 까지 번호가 부여된다. MCM은 메시지 변환기를 그룹 안에 생성하고 클라이언트마다 생성되는 메시지 변환기에 번호를 부여하여 그룹 내에서 구분이 되도록 하였고 한 그룹 내에 다수의 메시지 변환기가 번호로서 구분된다. 이들 메시지 변환기 그룹은 CMM이 관리한다. CMM은 여러 접속된 클라이언트들을 구분하고 각각 전송되어진 메시지에 구별한다. 또한 그룹 정보와 번호를 가지고 전송해야 할 클라이언트로 구분하며 전송된 메시지를 해당 그룹 내에 있는 클라이언트에게 전송하는 역할을 한다. CMM은 그룹과 메시지 변환기를 제거하는 역할을 담당하는데 클라이언트에서 정상적인 종료가 이루어지면 CMM은 해당 그룹 내에 존재하는 메시지 변환기를 삭제한다. 그리고 그룹 내의 모든 메시지 변환기가 삭제된 경우 해당 그룹을 삭제한다. 또한 클라이언트의 접속이 끊어진 경우를 체크하기 위해 CMM은 클라이언트에 연결 체크 데이터를 보내고 타임아웃을 적용하여 일정 시간 내에 해당 클라이언트의 메시지 전송이 이루어지지 않으면 해당 그룹 내의 메시지 변환기를 소멸시키고 연결을 닫는다.

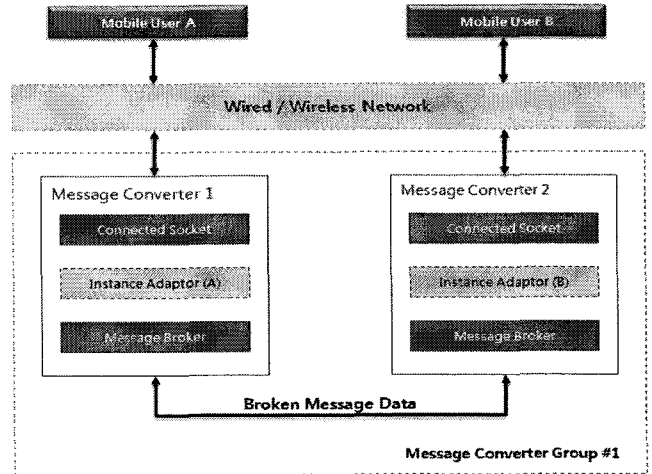


그림 5. 메시지 변환기 Group의 예
Figure. 5. Example of message converter group

[그림 5]는 서버에 서로 다른 모바일 사용자가 클라이언트 프로그램을 사용하여 접속할 때 메시지 변환기 그룹의 예를 나타낸 것이다. [그림 5]는 두 대의 서로 다른 모바일을 사용하는 사용자 A와 B가 인스턴트 메신저 서버에 접속하는 예를 나타낸다. 메시지 변환기는 MCM에서 생성되는데 해당 클라이언트와의 연결을 위한 연결 소켓과 인스턴트 어댑터, 메시지 브로커 등으로 구성 되어진다. 연결 소켓에서 연결된 클라이언트가 수신 받은 메시지를 인스턴트 어댑터에 넘겨준다. 이때 인스턴트 어댑터는 CM의 터미널 매핑에서 수집된 모바일 환경 정보를 가지고 변환에 필요한 정보로 만들어진다. 인스턴트 어댑터는 연결 소켓에서 보내는 정보를 변환하지 않는다. 그 이유는 인스턴트 어댑터는 다른 모바일 클라이언트에서 보내어진 메시지 정보를

자신의 모바일 환경에 맞게 변환시키고 캡슐화 하는 역할을 담당하기 때문이다. 즉, [그림 5]에서 모바일 사용자 A가 보낸 메시지는 서버의 연결 소켓에 들어오고 이 때 들어온 메시지는 인스턴트 어댑터에서 변환할 필요가 없기 때문에 메시지 브로커로 보내진다. 메시지 브로커는 서버 내에서 사용하는 캡슐화 되지 않은 메시지로서 서버 내에서 공통 메시지 (BMD: Broken Message Data)로 통한다. 또한 메시지 브로커는 모바일 사용자 A로부터 전송되어진 메시지를 BMD로 변환하고, 이때 변환에 필요한 정보는 인스턴트 어댑터 A에서 사용한 정보를 사용한다. 인스턴트 어댑터는 서버에 접속된 각각의 클라이언트에 맞도록 요청에 의해서 생성되는데, 고정적인 어댑터가 아닌 요청에 의해서만 생성 되도록 한 모듈을 단순화하여 메시지 처리 속도의 향상을 꾀하였다.

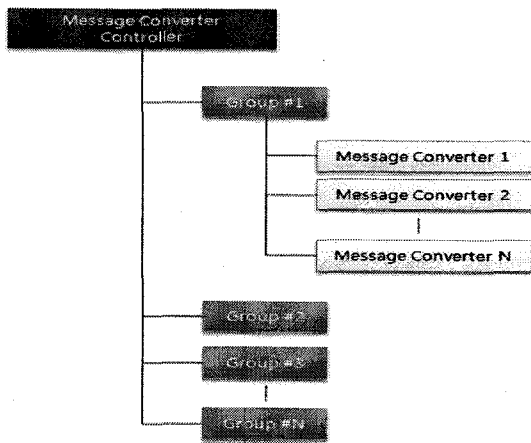


그림 6. 메시지 변환기 트리 구조

Figure. 6. Tree structure of message converter

BMD는 다시 같은 그룹 #1 내에 존재하는 모바일 사용자 B의 메시지 변환기 2로 전송된다. 전송된 메시지는 메시지 브로커에 전달되는데 이때 이미 메시지는 BMD 형태이기 때문에 바로 인스턴트 어댑터 B로 전달된다. 인스턴트 어댑터 B는 모바일 사용자 B의 클라이언트 환경에 맞게 메시지를 변환하고 캡슐화한다. 이때 변환된 메시지는 연결 소켓을 통해서 모바일 사용자 B의 클라이언트 프로그램에 전달된다. 지금까지의 구조는 서로 다른 단말이 1:1 통신 상태를 유지할 경우에 대한 내용이다. 다수 그룹과 메시지 변환기의 효율적인 관리를 위해 트리 구조를 사용하는 구조는 [그림 6]과 같다. [그림 6]에 보이듯이 CMM 내의 메시지 변환기 그룹들은 MCC를 루트로 하여 다수의 그룹을 관리한다. MCC는 그룹 정보를 가지고 있는데 각각의 그룹에 대하여 생성 및 소멸을 관장한다. 또한 각기 다른 그룹들의 정보를 실시간으로 관리하고 서버의 메모리 효율과 메시지 변환기 간의 정상적인 메시지 전송을 돕기 위해 MCC를 사용한다. 다음 절에서는 모바일 인스턴트 메신저 클라이언트 및 서버 구현 결과에 대하여 보이고 설명한다.

IV. 모바일 인스턴트 메신저 클라이언트 및 서버 구현 결과

본 연구에서는 윈도우즈 CE 계열 운영체제를 탑재한 PDA 보드 (MRP-S3C2440A)와 상용화 단말인 삼성 넥시오-XP30을 사용하여 모바일 인스턴트 메신저를 설계하고 구현하였다. 임베디드 개발 환경은 유니코드를 기본으로 하는데 윈도우즈 CE API를 사용하여 구축된 각 클라이언트는 송수신 시 사용할 데이터 구조에 특정한 약속을 한다. 본 연구에서 구현된 데이터 구조는 문자열 전송만을 구분하도록 설계되었는데 이는 서버를 통한 PDA와 XP30 간의 기본적인 무선 네트워크 환경을 구축하는데 우선순위를 맞추었기 때문이다. 추후 패킷 세분화 설계 작업을 통하여 다양한 데이터 활용이 가능하다. 그 중 음성 데이터의 송수신은 VoIP (Voice over Internet Protocol)를 가능하게 한다. [그림 7]은 각각의 PDA와 단말에서 발생하는 신호의 흐름도이다.

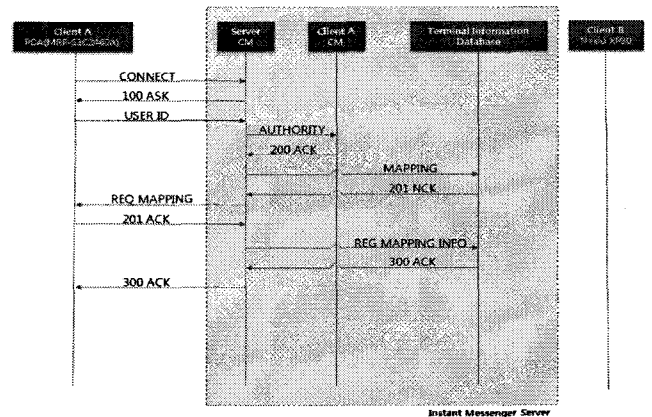


그림 7. PDA의 서버 등록을 위한 메시지 흐름도

Figure. 7. Message flow diagram for server registration of PDA

[그림 7]의 메시지 흐름도는 인스턴트 메신저 서버에 클라이언트 B인 단말이 우선적으로 접속하여 클라이언트 A인 PDA의 접속을 기다리고 있는 상태이다. PDA는 서버에 접속하기 위해서 'CONNECT' 메시지를 보내어 접속을 시도한다. 서버에 접속이 완료된 후 클라이언트 A는 사용자의 아이디를 서버에 전송한다. 서버의 CM은 클라이언트 A로부터 수신된 사용자 아이디를 authority 메시지로 캡슐화하여 ULD에 보내고 사용자를 확인한다. 등록된 사용자인 경우 '200 ACK' 메시지를 보내고 TID에게 클라이언트 A 제품의 특성 혹은 하드웨어적 환경에 대한 정보를 가져오기 위해 'MAPPING' 메시지를 요청한다. TID에 요구정보가 등록되어 있을 경우 '300 ACK' 메시지를 전송하여 바로 메시지 변환기를 생성하지만 요청 정보가 없을 경우 클라이언트 A에게 해당 정보를 요청하기 위해서 서버 CM에게 TID는 '201 NCK'를 전송한다. CM은 클라이언트 A에게

PDA 환경 정보를 요청하는 'REQ MAPPING' 메시지를 전송한다. 또한 수신 받은 클라이언트 A는 해당 정보를 수집하고 이를 '201 ACK' 메시지로 CM에게 전송하고 이를 수신 받은 CM은 바로 메시지 변환기를 생성하고 동시에 TID에게 PDA의 환경정보를 저장하기 위해 'REG MAPPING INFO' 메시지를 전송한다. 그 후 클라이언트 A는 메시지 변환기 그룹에 참가하기 위한 대기상태가 된다. [그림 8]은 클라이언트 A와 클라이언트 B가 생성한 채팅방에 참여하기 위해 동작되는 메시지의 흐름도이다.

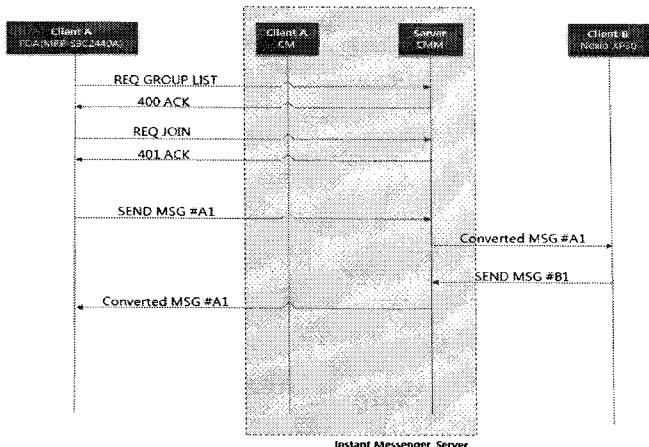


그림 8. PDA의 접속 이후에 채팅을 위한 그룹 참여 메시지 흐름도

Figure. 8. Message flow diagram of group participation for a chatting after PDA connection

클라이언트 A는 서버 CMM에게 현재 그룹 정보를 요청하는 'REQ GROUP LIST' 메시지를 전송한다. 그리고 CMM은 현재 그룹 정보를 클라이언트 A에게 '400 ACK'로 전달한다. 이후 클라이언트 A 사용자는 현재 개설되어 있는 채팅방 정보나 대기실의 현재 사용자에 대한 정보를 확인한다. 클라이언트 B가 생성한 채팅방에 참여하기 위해 'REQ JOIN' 메시지를 보낸다. 'REQ JOIN'에는 클라이언트 B가 생성한 그룹 번호가 들어있어 CMM은 이를 확인할 수 있다. 이후 CMM은 클라이언트 A의 메시지 변환기를 클라이언트 B가 속한 메시지 변환기 그룹에 포함한다. 그룹에 정상적으로 클라이언트 A가 포함되었다면 CMM은 클라이언트 A에게 '401 ACK' 메시지를 전송한다. 이로써 클라이언트 A는 클라이언트 B가 생성한 채팅방에 참여할 수 있게 된다. 이 후 클라이언트 A에서 문자열 메시지를 CMM에게 전송하면 CMM은 클라이언트 A의 메시지 변환기를 거쳐 클라이언트 B의 메시지 변환기를 통한 후 변환된 메시지를 클라이언트 B에게 전송한다. 각각의 클라이언트의 메시지는 각 클라이언트의 하드웨어 또는 운영체제의 특성에 따라 변환되어 전송된다. [그림 9]는 실제 PDA를 바탕으로 시연된 실행 장면을 보이고 있다. 실제 테스트에서 사용한 PDA는 WinCE 4.1을 사용하고 단말로 사용하는 XP30은 WinCE 5.0사용한다. PDA에서는 멀티바이트 문자열을 사용하고 단말에서는 유니코드 문자열을 사용하고 있다.

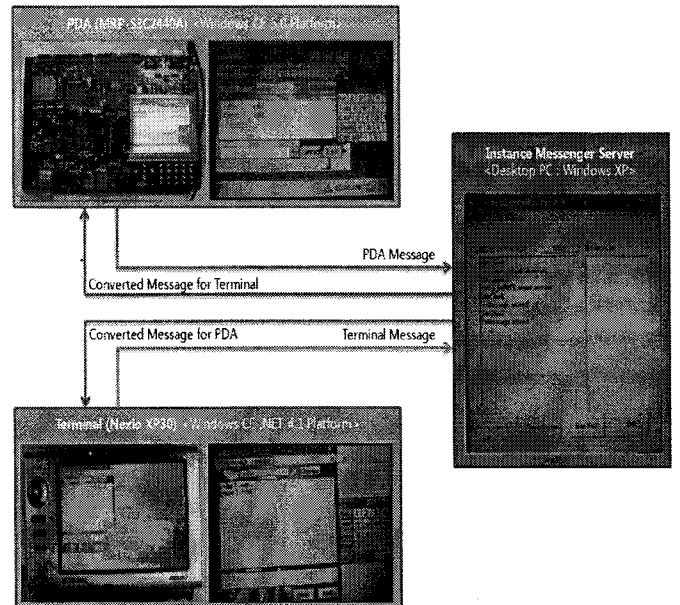


그림 9. PDA와 단말간의 실제 전송 화면

Figure. 9. Screen shot of real-time transmission between PDA and terminal

각각의 기기에서는 서버로 보내지는 서로 다른 구조의 문자열을 가지고 있다. 이를 해결하기 위해서 인스턴트 메신저 서버 내의 CMM에서 이들 멀티바이트를 유니코드로 또는 유니코드를 멀티바이트로 변환하는 과정을 메시지 변환기 모듈을 통하여 해결하였다. 서버와 단말기 간의 메시지 전달을 위해서 전달 메시지를 서버에 전송하고 수신 단말에게 전송하기전에 서버의 단말 정보를 분석하여 수신 단말의 특성에 맞게 변환한 뒤에 전송하는 과정을 거친다. 이러한 과정을 통해서 이종 단말간의 데이터 전송 시 문제가 발생하는 것을 줄일 수 있다. 또한 중앙에서 단말 정보를 관리하기 때문에 새로운 기기 단말이 출시되었다고 서버에 단말 정보를 입력함으로써 즉시 메신저 서비스를 할 수 있는 장점을 갖고 있다. 본 구현 결과에서는 단순히 문자열 변환을 통한 테스트이지만 서로 다른 기종간의 네트워크 전송에 문제가 있을 경우 이를 서버에서 해결할 수 있었다. 그리고 이종 기기간의 프로그램을 구동 시키는데 복잡한 수정 없이 서버에 하나의 모듈 추가로 위와 같은 문제점을 해결 할 수 있다. 기존에는 이러한 문제를 해결하기 위해서 단말기 안에 기기들 간의 통신을 위한 미들웨어 소프트웨어를 설치하는 경우가 대부분이다. 단말기에 설치된 이러한 미들웨어 소프트웨어는 수많은 단말기에 대한 정보를 저장하기가 어렵고 또한 업그레이드가 쉽지 않다. 이러한 방식보다는 중앙의 서버를 통한 문제해결이 관리자 입장에서 관리가 용이하고 이종 단말기 간의 통신 비용 등의 부담을 효과적으로 줄일 수 있다.

V. 결론 및 향후 연구 계획

본 논문에서는 TCP/IP를 기반으로 한 서버와 윈도우즈

CE 5.0 기반 PDA (MRP-S3C2440A) 및 윈도우즈 CE .Net 4.1 기반 무선 단말 (넥시오-XP30)의 클라이언트 간 통신을 구현하였다. 개발에서 사용된 클라이언트 단말기는 윈도우즈 CE 5.0과 Win CE .NET 4.1로 구동 환경의 차이가 있는 상태이며 PDA를 구성하고 있는 교육용 보드와 일반 상용화된 스마트 단말간의 통신 구현을 위해 메시지 분석기를 활용하여 데이터의 송수신이 가능하도록 만들었다는 점에서 의의가 있다고 볼 수 있다. 본 연구의 결과를 이용하여 다음 단계로는 패킷 구조 개선을 통한 이종 모바일 단말 기기간의 음성 및 영상 통화 기능을 활용할 수 있다. 또한 이러한 연구를 통해 무선 환경에서의 데이터 통신과 사용자 간의 단말기 활용도 및 인터넷 기반에서의 정보통신 활용도를 한층 높일 수 있을 것으로 기대된다. 앞으로 셀룰러 폰, PDA, 휴대용 PC, 스마트 폰 등 무선 단말기들의 활용 범위는 다양한 분야에서 전문성과 기능성의 측면에서 급성장하고 있으며, 통합화, 융합화 되어가고 있다. 2010년 현재, 애플의 앱스토어와 구글의 애플리케이션 마켓 등 다양한 모바일 애플리케이션 시장이 성장하고 있다. 이러한 상황에서 이종 단말 기기간 데이터 통신, 음성 및 영상 통화를 비롯하여 개인 및 기업, 단체 또는 특정 전문 집단에서의 무선 단말기의 활용도가 무한히 성장할 것으로 추측된다. 각종 단말 기기 특성 따른 데이터 전송 시 발생하는 문제점을 해결하기 위해서 논문에서는 중앙 서버를 이용한 인스턴트 메신저 시스템을 제안하였다. 중앙 서버 방식을 수 많은 모바일 기기의 특성을 관리하기 용이하고 단말 기기의 부담을 줄이는 장점을 갖고 있다. 제안된 시스템을 확장하여 현재 상용화된 구글의 Android Platform, WIPI, Qualcomm의 Brew Mobile Platform 등 다양한 모바일 기기 환경속에서 적용할 수 있는 모바일 인스턴트 메신저 시스템을 구현할 계획이다.

참고 문헌

[1] 양순옥, 김성석, 정광식, "유비쿼터스 컴퓨팅 개론", 한빛미디어(단행본), 2008.
 [2] 모바일 타임즈. <http://www.cellular.co.kr>
 [3] 김병만, "임베디드 모바일 프로그래밍", 한티미디어(단행본), 2007.
 [4] 최정윤, "모바일 메신저 기술과 특허동향", 한국특허정보원, 2004.
 [5] 강동훈, 모바일 인스턴트 메신저 시스템 구현에 대한 연구, 국민대학교, 석사논문, 2004.
 [6] 강선영, 최종원, 휴대단말기와 PC간의 양방향 메시지 전달을 위한 인스턴트 메신저의 설계 및 구현, 숙명여자대학교, 한국정보과학회, 추계학술발표회 27권 2호, 2000
 [7] 나성준, 김동욱, 이진하, 정희진, 신동렬, 멀티 에이전트 기반의 모바일 메신저 서비스의 설계와 구현, 한국방송공학회, 동계학술대회, pp.44, 2007

[8] 김종민, 이성용, 최창열, 리눅스 기반 무선 인스턴트 메신저 시스템의 설계 및 구현, 한국정보과학회, 춘계학술발표회 31권 1호, 2004
 [9] Douglas Boling, "Programming Microsoft Windows CE .NET (3rd Ed.)", 정보문화사, 2004.
 [10] W.E.E.G (Windows Embedded Expert Group), <http://cafe.naver.com/wincepro.cafe>,
 [11] 고재관, "윈도우즈 임베디드 CE 프로그래밍", 정보문화사, 2006.
 [12] MDS Technology, <http://www.mdstec.com>



방 지웅(Ji-woong Bang)

2008년 2월 단국대 컴퓨터과학(공학사)
 2010년 2월 단국대 컴퓨터과학(공학석사)
 2010년 2월~현재 단국대 컴퓨터과학 박사과정
 YAMAIA-NGTMS 기술연구소 연구원
 ※주관심분야: 모바일 응용, 임베디드 시스템 인공지능



김 대원(Dae-won Kim)

1993년 2월 중앙대 전자공학과(공학석사)
 1996년 5월 USC(So. Cal.)-EE (공학석사)
 2002년 5월 ISU(Iowa St.)-EE (공학박사)
 2002년 5월 ~ 2004년 8월 삼성전자 TN총괄
 통신연구소 차세대 단말기술 Lab, 책임 연구원
 2004년 9월 ~ 현재 단국대학교천안캠퍼스 공학대학
 컴퓨터학부 멀티미디어공학과 부교수
 ※주관심분야: 디지털신호처리, 모바일 응용, NDE