

정보가전기기간 상호운용성 지원을 위한 OSGi 기반 UPnP 기술

김 동 희*, 이 화 영*, 임 경 식*, 조 총 래**

경북대학교 컴퓨터과학과 컴퓨터통신·이동컴퓨팅 연구실, 한국전자통신연구원 홈네트워크팀

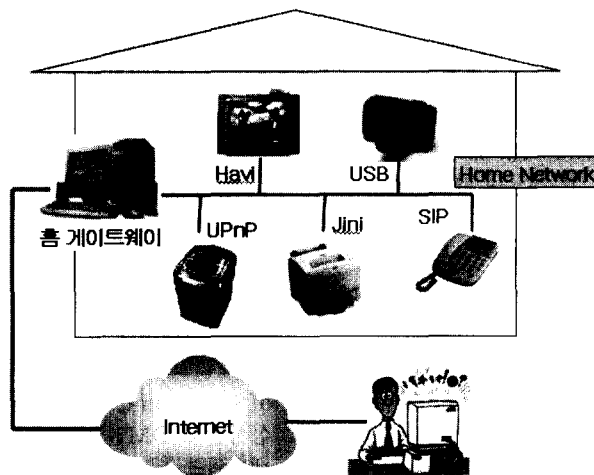
I. 서 론

인터넷은 월드와이드웹이 탄생한 이후 질적, 양적으로 급성장하여 2001년 현재 인터넷 사용자는 국내에서는 2천 4백만 명이 있으며, 전 세계적으로는 5억 이상이다(정보통신부, 인터넷이용자 통계, 2001년 12월 31일). 이러한 인터넷의 대중화와 함께 ISDN, ADSL, 케이블 모뎀 등에 의한 맥내 네트워크 장비의 고속화를 바탕으로 일반 가정에서의 인터넷에 대한 요구가 확산되고 있으며, 이러한 요구를 만족시킬 수 있는 홈 네트워크 기술에 대한 관심이 증대되고 있다.

홈 네트워크 기술은 <그림 1>과 같이 맥내의 가전 기기들을 HomePNA, Bluetooth, PLC 등으로 네트워크를 구성한 후, 제어 미들웨어를

이용하여 사용자가 이들을 제어할 수 있도록 하여주는 기술이다. 특히, 제어 미들웨어는 홈 네트워크에 연결되는 가전 기기들을 발견하고 관리하며, 사용자가 이들을 제어할 수 있도록 인터페이스를 제공하는 역할을 수행한다. 현재 이러한 제어 미들웨어 기술로는 UPnP, Jini, Havi, LonWorks 등의 기술이 제안되고 있다.

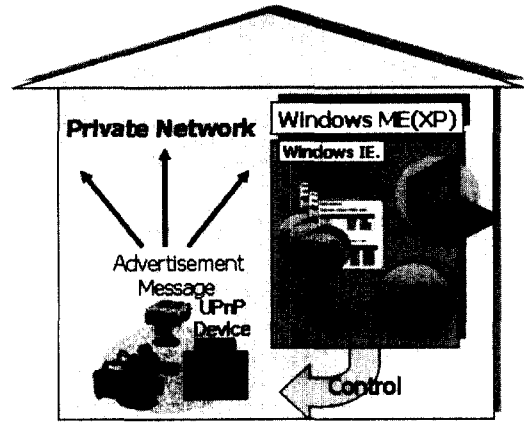
마이크로소프트사에서는 UPnP를 그리고 썬 마이크로시스템사에서는 Jini를 각각 제안하였으며, 이들은 IP를 기반으로 장치의 제어와 공유를 제공한다. Havi는 엔터테인먼트 분야를 위하여 제안된 미들웨어 기술로서 소니, 미쯔비시 등의 AV 가전 업체들이 중심이 되어 제안한 기술이다. 그러나 현재 어떠한 기술도 확실한 우위를 점유하지는 못한 상황으로 선진 기업에서는 다수의 표준화 작업에 참여하여 앞으로 우위를 점유하게



<그림 1> 홈 네트워크 구성도

될 기술에 대하여 준비를 하고 있다.

본 고에서는 홈 네트워크의 제어 미들웨어중 하나인 UPnP와 홈 게이트웨이 기술인 OSGi에 대하여 살펴보고 이들의 연동방안에 대하여 기술하고자 한다. 2장과 3장에서는 이 두 미들웨어의 개요 및 그 구현사례에 대하여 소개한다. 그리고 4장에서는 OSGi상에 UPnP를 탑재한 시스템에 어떠한 것이 있으며, 이들이 제공하는 서비스에 대하여 간략하게 소개한다. 그리고 5장에서는 향후 홈 네트워크의 제어 미들웨어 기술의 전망을 기술한 후 결론을 맺도록 한다.



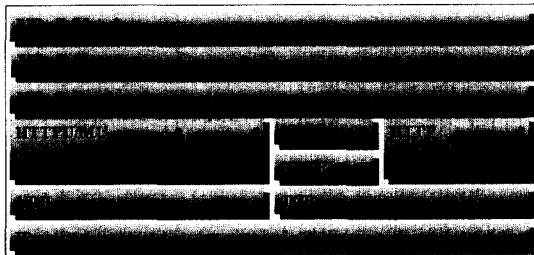
<그림 3> Windows ME/XP의 동작 구조

II. UPnP 기술

1. UPnP 개요

UPnP 포럼은 1999년 10월 18일 설립되었으며, 가전제품, 컴퓨터, 홈 오토메이션, 모바일 디바이스 등 각 분야의 460여 업체들이 참여하고 있다. 이 포럼은 인터넷 기반의 커뮤니케이션 표준을 바탕으로 하여 장치와 서비스의 표준안을 제정하고 발표하고 있다. 이는 장치들을 가정이나 사무실에서 쉽고 간단하게 네트워크를 구축하여 이들을 제어하는 것을 목적으로 한다.

UPnP는 TCP/IP 기술을 바탕으로 현재 인터넷에서 사용되고 있는 HTTP, SSDP(Simple Service Discovery Protocol), GENA(General Event Notification Architecture), 그리고 SOAP(Simple Object Access Protocol) 등의 프로토콜을 이용하여 장치의 정보를 수집하고



<그림 2> UPnP 프로토콜 스택

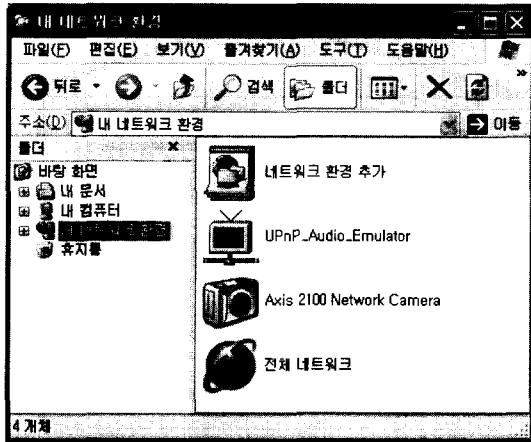
장치를 제어한다.

이 기술은 사용자에게 서비스를 제공하는 맥내망의 장치와 이들을 제어하는 컨트롤 포인트로 구분된다. 그리고 이들의 상호 작용으로 사용자는 자신이 원하는 서비스를 제공받을 수 있으며, 동작 과정은 다음과 같다.

UPnP는 IP 기반으로 동작하므로 장치들은 우선적으로 자신의 IP 주소를 설정하여야 하며, 이는 DHCP(Dynamic Host Configuration Protocol) 또는 AutoIP를 이용하여 설정한다. 이후, 장치들은 SSDP를 이용하여 자신이 제공하는 서비스를 광고하며, 컨트롤 포인트는 SSDP와 GENA를 이용하여 자신이 제어하고자 하는 장치를 찾게된다. 컨트롤 포인트는 장치들을 발견한 후, 제어를 위한 상세문서(장치 디스크립션, 서비스 디스크립션)을 장치로부터 전송받아 이들을 제어할 준비를 마치게 된다. 그리고 사용자의 명령에 따라 또는 미리 설정된 방식에 따라 SOAP을 이용하여 장치들을 제어하게 되며 GENA를 통하여 이벤트를 받게 된다^{[1,2]}}.

2. Windows ME/XP

UPnP의 가장 대표적인 구현물로서 Windows ME/XP 운영체제에 탑재된 UPnP 모듈을 들 수 있다. 이 모듈은 컨트롤 포인트와 이를 사용자가 제어할 수 있는 API로 구성되어 있다.



〈그림 4〉 장치의 존재를 알리는 장치 아이콘

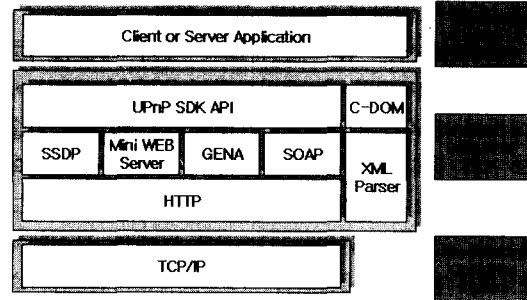
만약 네트워크상에 장치가 존재하게 된다면 윈도우는 〈그림 4〉와 같이 네트워크 환경에 장치의 존재를 아이콘으로써 사용자에게 알려주게 된다. 그리고 사용자는 이 아이콘을 선택함으로써 장치의 제어를 위한 프리젠테이션 문서를 전송 받아 이들을 제어할 수 있다.

프리젠테이션 문서는 비주얼 베이직 스크립트를 포함하는 HTML 문서로서 이는 장치 공급업자들이 사용자들의 편리한 제어를 위하여 제공되는 인터페이스이다. 그리고 비주얼 베이직 스크립트에서는 윈도우용 UPnP API를 호출하여 장치를 제어할 수 있도록 되어 있다.

이는 운영체제 내부에 UPnP 모듈을 탑재하고 아이콘 형식으로 인터페이스를 제공함으로써 사용자에게 편리함과 친숙함을 제공한다. 그러나 사용자는 이 시스템을 사용할 경우 홈 네트워크 내에서만 장치를 제어할 수 있는 단점을 가지고 있다. 이는 윈도우 운영체제가 장치의 광고 메시지를 받을 수 없는 지역에 존재할 경우 장치 아이콘이 사라지기 때문이다.

3. Intel UPnP SDK

Intel UPnP SDK에는 개발자들이 리눅스 환경에서 컨트롤 포인트와 장치를 개발할 수 있도록 프로토콜과 XML DOM 라이브러리, 그리고 기타 웹 관련 API를 제공한다. 현재 1.0.4 버전



〈그림 5〉 Intel UPnP SDK의 구조

까지 공개되어 있다.

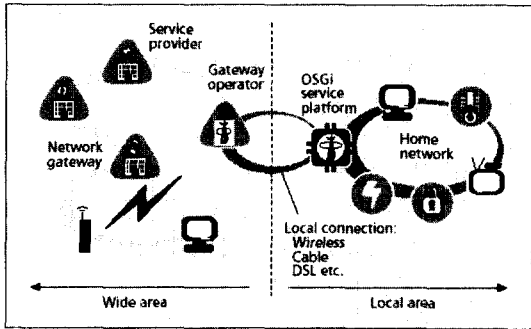
〈그림 5〉에서 보듯이 이 개발도구는 3개의 계층으로 구성되어 있다. 상위 계층은 개발자들이 구현하게 될 컨트롤 포인트나 장치 등의 응용 프로그램이다. 중간 계층은 8개의 컴포넌트로 이루어진 개발도구의 핵심 부분이다. 이 계층에는 UPnP 프로토콜과 함께 장치에서 사용할 수 있는 미니 웹 서버 모듈, 그리고 XML 파서가 내장되어 있다. 하위 계층은 OS에서 지원하는 TCP/IP 프로토콜 계층이다.

이 개발도구는 네트워크의 특정 이벤트가 발생하였을 때, 개발자가 프로그램에서 등록된 콜백 함수를 호출함으로써 개발자에게 개발도구를 사용하기에 편리함을 제공한다. 개발자들은 홈 네트워크에 장치가 연결되어 광고 메시지를 전송하는 것, 장치가 이벤트 메시지를 전송하는 것, 그리고 컨트롤 포인트가 장치검색 메시지를 전송하는 것 등을 모두 이벤트로 처리할 수 있다^[8].

III. OSGi 기술

1. OSGi의 개요

OSGi는 외부 인터넷에 존재하는 다양한 서비스를 가정, 자동차 또는 작은 사무실과 같은 내부 홈 네트워크 망으로 전달하기 위한 표준화된 API의 제공을 목적으로 썬 마이크로시스템사를 중심으로 시스코, 컴팩, 에릭슨 등을 포함한 15개



〈그림 6〉 동적인 서비스 관리에 대한 OSGi의 개념

업체가 참여한 가운데 1999년 3월 설립되었다. 이는 썬 마이크로시스템사의 자바기술을 기반으로 2000년 5월 서비스의 생명주기 관리, 디바이스 액세스, 로그 및 HTTP 서비스를 포함한 OSGi 서비스 게이트웨이 1.0을 발표하였으며, 2001년 10월에는 서비스의 동적인 발견, 서비스 레지스트리의 관리, 서비스 실행에 대한 보안 및 데이터 관리 등을 추가한 서비스 플랫폼 2.0을 발표하여 개방형 서비스 게이트웨이의 표준으로 부각되고 있다.

현재 홈 네트워킹 시장은 다양한 유·무선 네트워크 기술을 기반으로 다양한 홈 게이트웨이, 지능형 정보가전기 및 홈 네트워킹 미들웨어들이 난립하고 있으며, 서로 다른 H/W 플랫폼, OS 및 네트워크 프로토콜에 따른 수많은 서비스 개발환경이 존재한다. 이와 같은 홈 네트워크의 이질성과 복잡성을 해결하기 위하여 OSGi는 동일한 형태의 API를 통하여 서비스의 동적인 관리에 대한 규격을 정의하고, 플랫폼 독립적인 서비스 실행환경인 프레임워크를 제공함으로써 임베디드 시스템 상에서 서비스를 개발하는 프로그래머로 하여금 최적의 개발환경을 제공하는 임베디드 서버 솔루션으로 떠오르고 있다. 동적인 서비스 관리에 대한 개념은 〈그림 6〉과 같다.

2. OSGi 기술 개발 사례

1) Java Embedded Server

Java Embedded Server(이하 JES)는 2000

년 5월에 발표한 OSGi Service Gateway 1.0을 기준으로 구현된 것으로 소용량의 메모리 풋프린트와 저 수준의 컴퓨팅 파워를 가지는 임베디드 장치를 위한 소용량의 응용 서버이다¹⁴⁾.

JES는 크게 두 가지 시스템 컴포넌트인 프레임워크와 서비스 번들로 구성되어 있다. 프레임워크는 서비스에 대한 컨테이너로서 네트워크를 통하여 전달된 서비스를 동적으로 설치하고 관리하기 위한 개방형 서비스 게이트웨이의 역할뿐만 아니라, 다수의 서비스가 동시에 실행되기 위한 서비스 실행환경을 제공한다. 서비스 번들은 OSGi에서 규정하는 코어 서비스 번들과 확장 서비스 번들로 분류할 수 있으며, 각 번들은 컴포넌트화된 서비스를 JAR(Java ARchive)의 형태로 패키징한 것이다.

◆ 코어 서비스

- ① HTTP 서비스
 - 표준 HTTP 서비스
- ② 연결 관리 서비스
 - OSGi의 디바이스 액세스 서비스
- ③ 로그 서비스
 - 이벤트와 에러에 대한 원격 로깅 서비스
- ④ 쓰레드 관리자 서비스
 - 쓰레드 풀 관리를 위한 서비스
- ⑤ 스케줄러 서비스 번들
 - 이벤트 스케줄링에 대한 서비스

◆ 확장 서비스

- ① 날짜 서비스
- ② 원격관리 서비스
 - 웹을 통한 JES의 원격 관리 서비스
- ③ Servlet 서비스
 - 자바 Servlet API
- ④ RMI 서비스
 - 자바 RMI API
- ⑤ SSL 서비스
 - 보안채널을 위한 서비스

위에서 나열한 서비스들은 서비스 제공자에 의

하여 외부 네트워크로부터 홈 내부의 네트워크로 동적으로 전달되어 JES 프레임워크 상에서 실행된다. 궁극적인 JES의 목적은 인터넷으로부터의 다양한 서비스를 동적으로 설치하여 보다 안정적이고 신뢰적인 실행환경을 제공하는 것이지만 아직까지는 프레임워크에서 서비스 실행에 대한 보안사항이 고려되지 않아 해결해야 할 많은 문제점을 지니고 있다.

2) Gamespace's Distributed Service Platform

Gamespace's Distributed Service Platform (이하 GDSP)은 OSGi 프레임워크 규격 1.1을 기준으로 구현된 분산형 서비스 플랫폼으로서 서비스 추적, 사용자 및 패키지 관리 서비스와 같은 OSGi 서비스 플랫폼 2.0의 API를 다수 포함하고 있다.

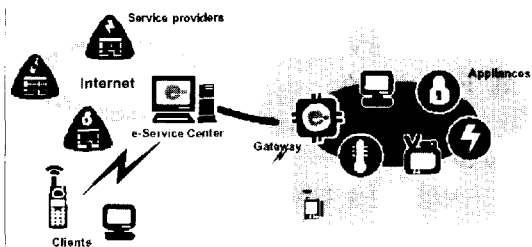
GDSP는 다양한 소프트웨어 및 서비스를 동적으로 관리하고 조합할 수 있는 게이트스페이스사의 telematic-system 및 에릭슨사의 e-service system의 서비스 플랫폼으로 장착되어 상업화를 시도하고 있다^[6]. E-service system 환경은 <그림 7>과 같다.

에릭슨사는 분산되어 인터넷에 존재하는 다수의 서비스를 집적한 e-service center가 존재하며 서비스 게이트웨이는 이를 통하여 다양한 서비스를 제공받는 e-service system이라는 사업 모델을 제시하고 있다.^[6] 또한 게이트스페이스사는 GDSP를 기반으로 다양한 어플리케이션을 개발함에 있어 필요한 서비스들을 번들의 형태로 개발하여 탑재한 Service Gateway Applica-

tion Development Kit(이하 SGADK)을 제공하고 있으며 다음과 같은 컴포넌트로 구성되어 있다.

- ① HTTP 및 HTTPS 서비스
 - OSGi 규격 1.0에서 제정된 HTTP 서비스 및 보안을 적용한 HTTPS 서비스
- ② 보안 메시징 서비스
 - 서비스와 디바이스간의 암호화된 메시지 전송을 위한 서비스
- ③ 원격 관리 서비스
 - GDSP에 대한 원격 관리 서비스
- ④ 플랫폼 관리 서비스
 - 게이트웨이 오퍼레이터를 위한 서비스 관리를 위한 서비스
- ⑤ 요금부과 서비스
 - 서비스 사용자에게 대한 요금부과를 위한 서비스
- ⑥ 사용자 관리 서비스
 - GDSP 사용자에게 대한 인증 및 권한 부여 서비스
- ⑦ 디바이스 액세스 서비스
 - 다양한 디바이스 드라이버 및 디바이스 관리 서비스

이처럼 SGADK는 GDSP상에 OSGi 규격 2.0에서 요구하는 기본 서비스 외에 서비스 개발 시 필요한 다양한 서비스를 탑재하고 있으며 JES와는 달리 보안에 대한 사항을 충분히 고려하였다. 또한, GDSP를 기반으로 한 비즈니스 모델을 제시하여 상업화에 주력하고 있다.

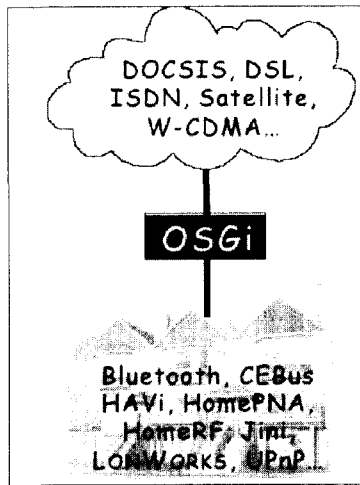


<그림 7> e-service system 환경

IV. UPnP over OSGi

1. 홈 네트워킹 미들웨어의 연동

앞서 언급한 바와 같이 OSGi는 WAN상에서 다양한 서비스 제공자에 의하여 분산되어 존재하는 서비스들을 홈 네트워크 상에 동적으로 전달



〈그림 8〉 OSGi기반의 홈네트워킹 미들웨어

하는 게이트웨이로 동작하며, 홈 네트워킹 미들웨어는 홈 내부의 지능형 정보가전기기간의 통신 및 제어의 목적을 지니고 있다. 동적인 서비스 관리기능과 홈 네트워킹 미들웨어의 지능형 정보가 전기기에 대한 제어 기능을 연동하기 위하여 홈 네트워킹 미들웨어에 대한 서비스 번들을 개발하여 OSGi상에 장착함으로써 두 가지 기능에 대한 통합형 모델을 제시할 수 있다. OSGi 기반의 홈 네트워킹 미들웨어의 연동 모델에 대한 개념은 〈그림 8〉과 같다.

2. Prosys UPnP Bundle

프로시스트사는 OSGi 프레임워크인 mBedded Server(이하 mBS)상에 Jini, UPnP와 같은 다양한 종류의 홈 네트워킹 미들웨어를 번들의 형태로 개발한 후, mBS상에 탑재하여 집안의 지능형 가전기기의 제어에 대한 방안을 연구하였다. 여러 미들웨어 중에서 특히 UPnP 번들의 경우 UPnP 표준 API, OSGi 서비스와의 연동 API 및 컨트롤 포인트 등의 기능을 내장하고 있다.

프로시스트사에서 제공하는 UPnP 서비스 번들은 다음과 같은 기능을 포함하고 있다.

① UPnP-to-OSGi 서비스 변환

- 홈 네트워크 내에 새로운 디바이스가 추가 되었을 때 새로이 장착된 디바이스상의 서비스를 OSGi 서비스로 변환

② OSGi-to-UPnP 서비스 변환

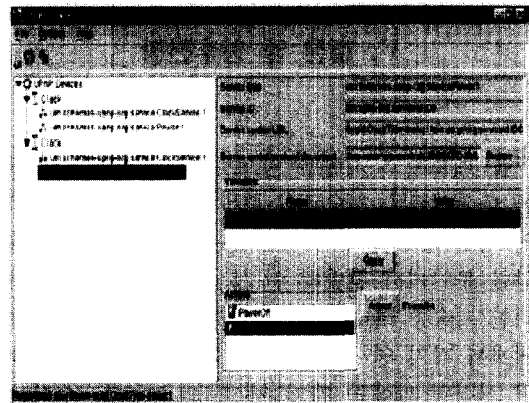
- OSGi에서 제공되는 서비스를 UPnP 기반 서비스로의 변환

③ Base-Driver

- UPnP 컨트롤 포인트

UPnP 서비스 번들은 번들 액티베이터와 서비스로 구성되어 있으며, 서비스는 다시 UPnP 서비스와 UPnP Export Manager 서비스로 구성되어 있다. 개발자는 이 서비스를 사용하기 위하여 Device-Listener 인터페이스를 구현하여야 하며, UPnPDevice 서비스를 통하여 디바이스의 정보를 얻을 수 있으며, 특히 UPnPService 서비스는 컨트롤 포인트의 기능을 내장하고 있어 홈 네트워크 내의 지능형 정보가전기기를 제어할 수 있다. 또한 사용자에게 편리한 인터페이스를 제공하기 위하여 〈그림 9〉처럼 UPnP 브라우저를 제공한다.

OSGi상에 UPnP, Jini 등의 홈 네트워킹 미들웨어를 탑재함으로써 각각의 미들웨어 구조상 핵심적인 컴포넌트들은 번들의 형태로 장착되어 라이프 사이클의 관리가 용이해지며, 컴포넌트들 사이의 연동을 통하여 다양한 홈 네트워킹 미들웨어를 통합할 수 있을 것이다^[7].



〈그림 9〉 mBedded Server의 UPnP 브라우저

3. 기존 구현물의 번들화 방안

기존의 UPnP 구현물의 공간 제한적인 단점을 극복하기 위하여 개발자들은 자신이 선호하는 언어를 이용하여 UPnP 프록시를 구현할 수 있으며, 사용자는 이를 이용하여 외부 인터넷에서 맥 내망의 장치들을 제어할 수 있다.

그러나, OSGi의 번들은 Java로 구현되어야 하므로 Java 이외의 언어로 구현된 개발물을 OSGi상에 탑재하기 위하여 다음과 같은 요구사항을 만족하여야 한다.

- 번들의 형태로 프레임워크에 장착
- 번들의 라이프사이클 관리
- 번들과 UPnP 프록시의 원격 제어/관리
- UPnP 서비스 제공을 위한 API 제공

이러한 요건을 만족시키기 위하여 개발자는 관리 모듈과 UPnP 핵심 모듈로 분리하여 개발할 수 있다. 관리 모듈은 핵심 모듈을 제어하고 관리하는 것으로서 번들의 형태로 프레임워크에 장착될 수 있다. 그리고 핵심 모듈은 관리 모듈과 통신을 통하여 전체 시스템이 번들의 형태로 동작하는 것과 동일한 효과를 얻을 수 있다.

이러한 방법을 통하여 기존의 구현물을 Java를 이용하여 다시 구현할 필요가 없으므로, OSGi 상의 번들의 형태로 개발하는 기간을 단축할 수 있다.

V. 결론 및 향후전망

홈 네트워크 기술은 앞으로 사람들의 삶의 질을 향상시켜줄 기술로서 주목받고 있으며 물리적인 하부 망 기술부터 상위 응용 프로그램까지 다양한 분야에서 활발히 연구되고 있다. 그리고 이러한 기술중 하나인 미들웨어 기술 역시 여러 단계에서 각기 연구되고 있다. 처음 미들웨어 기술이 시작될 때에는 각기 자신들의 기술을 주장하고 따로 발전하였으나, 지금의 추세는 각 기술들

을 각각 발전시키면서 이러한 기술들을 같이 적용하고 연동하는 방향으로 추진되고 있다. Jini와 Havi의 연동 또는 UPnP와 Havi와의 연동, 그리고 OSGi를 기반으로한 UPnP, Jini, 그리고 LonWorks 기술 등이 활발히 연구되고 있다. 본고에서 소개한 프로시스트의 mBedded Server가 이의 예라 할 수 있다.

현재에는 앞서 언급된 미들웨어 기술들을 적용한 상용 제품들이 많이 출시되지는 않았으나, 많은 가정으로의 인터넷 보급을 바탕으로 몇 년 후에는 많은 제품들이 가정 내에 설치될 것으로 예상된다. 그리고 이러한 제품들은 미들웨어 상호연동 기술을 바탕으로 유기적으로 상호 동작하여 인간에게 편리함을 제공할 것으로 예상된다.

참고 문헌

- [1] Microsoft Corporation, "Understanding Universal Plug and Play," URL: http://www.upnp.org/download/UPnP_UnderstandingUPnP.doc, Jun. 2000.
- [2] Microsoft Corporation, "Understanding Universal Plug and Play Device Architecture," URL: http://www.upnp.org/download/UPnPDA10_20000613.htm, Jun. 8, 2000
- [3] Daniel Baumberger, "Universal Plug and Play Software Development Kit V1.0 for Linux," URL: <http://upnp.sourceforge.net/>, Sep. 14, 2000
- [4] Sun Microsystems, "How to write your first JES program," URL: <http://java.sun.com/j2me/docs/pdf/tutorialwin.pdf>
- [5] Gatespace Corporation, "Gatespace Extending own Distributed Service Platform to Complete Telematics Solution," URL: http://www.gatespace.com/news/archive/20020304_en.shtml
- [6] Ericsson Corporation, "Residential e-

services," URL:http://www.ericsson.com/wireless/products/ebox

(7) Prosysy Corporation, "UPnP Package,"

URL:http://www.prosysy.com/PDF26_10/upnp.pdf

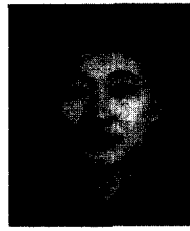
저자 소개



金 東 熙

2000년 2월 경북대학교 컴퓨터과 학과 이학사, 2002년 2월 경북대학교 컴퓨터과학과 이학석사, 2002년 3월 경북대학교 정보통신 학과 박사과정, 2001년 12월~현재 : (주)네이버시스템 주임 연구원, <주관심 분야: WAP, 이동 컴퓨팅, 홈 네트워킹>

구원, <주관심 분야: WAP, 이동 컴퓨팅, 홈 네트워킹>



林 敬 植

1982년 2월 경북대학교 전자공학과 공학사, 1985년 2월 한국과학기술원 전산학과 공학석사, 1994년 University of Florida 전산학과 공학박사, 1985년~1998년 : 한국전자통신연구원 책임연구

원, 실장, 1997년~현재 : Computer Communications 논문심사위원, 1997년~현재 : 한국정보과학회 정보통신연구회 운영위원, 1999년~현재 : 한국정보과학회 논문지 편집위원, 1998년~현재 : 경북대학교 컴퓨터과학과 조교수, <주관심 분야: 이동 컴퓨팅, 무선인터넷, 홈 네트워킹, 컴퓨터통신>



李 和 榮

2002년 2월 경북대학교 컴퓨터과 학과 이학사, 2002년 3월 경북대학교 컴퓨터과학과 석사과정, <주관심 분야: 이동 컴퓨팅, 홈 네트워킹, 액티브 네트워크>

구원, <주관심 분야: 이동 컴퓨팅, 홈 네트워킹, 액티브 네트워크>



趙 忠 來

1994년 2월 부산대학교 전자계산학과 (학사), 1996년 2월 부산대학교 전자계산학과 대학원 (석사), 1996년 3월~2000년 7월 : 한국기계연구원 연구원, 2000년 7월~현재 : 한국전자통신연구원 연구원, <주관심 분야: 홈네트워킹, Ad Hoc 네트워킹, 정보 검색>

구원, <주관심 분야: 홈네트워킹, Ad Hoc 네트워킹, 정보 검색>