

OSGi(Open Service Gateway Initiative)기반의 Smart Home Service Bundle 연구

손민우^o, 최중화, 김건희, 박병철, 신동일, 신동규
 세종대학교 대학원 컴퓨터공학과 멀티미디어 인터넷 연구실
 {minwoo15^o, com97, ghkim, leon, dshin, shindk}@gce.sejong.ac.kr

A Study on Smart Home Service Bundle Architecture based on OSGi(Open Service Gateway Initiative)

Minwoo Son^o, Jonghwa Choi, Gunhee Kim, Byungchul Park, Dongil Shin, Dongkyoo Shin
 Dept of Computer Engineering, Sejong University

요 약

가정 내의 가전기기들이 디지털화 되고 있고, 이것들을 네트워크로 연결하여 원격지에서 제어하는 스마트 홈 관련 기술들이 현실화 되고 있다. 이러한 환경에서의 컴퓨팅을 위해서는 응용 프로그램간의 호환성 뿐 아니라 데이터의 호환성, 그리고 이종의 프로토콜간의 연동이 필수적이다. 따라서 이를 위한 소프트웨어가 바로 스마트 홈 미들웨어이다. OSGi(Open Service Gateway Initiative)[8]는 장비 연결 및 제어로 얻을 수 있는 유효한 서비스나 여러 미들웨어들 간의 통합을 위한 제반 환경 제공을 목표로 하는 비 표준화 단체로서 OSGi 프레임워크 환경에서 서비스들은 서비스 게이트웨이와 서비스 번들의 life-cycle을 따라 동적으로 전개된다. 본 저자들은 SMF(Service Management Framework)[9]를 사용하여 OSGi를 기반으로 한 Smart Home Service Bundle에 관해서 연구하였다.

위한 아키텍처 설계를 서술하고 결론을 맺을 것이다.

1. 소개

스마트 홈(Smart Home)[1]이란 가정에 있는 모든 장비들을 연결하여 서로 통신을 할 수 있도록 함으로써, 가족 구성원이 자신의 현재 위치에 관계없이(가정 내의 위치 또는 가정 밖의 위치에 무관하게) 가정 내 장비를 제어하거나 통신할 수 있도록 해주는 기술이다. 예를 들어, 거실에 앉아서 잠자리에 든 아이들 방의 불을 끈다거나, 외출하다가 도시가스 밸브를 열어놓은 것이 생각나면 휴대폰으로 가스차단기를 작동시키거나, 외부의 기온을 체크하여 자동으로 실내온도를 맞춘다는 등 일상 생활에 즉시 적용할 수 있는 기술인 Home Theater나 Home Office, Home Security와 같은 복합적 기능을 포함하는 진일보한 개념이다.

스마트 홈은 최초로 1979년 X-10 이라는 전력선을 이용한 통신 방법[2]이 등장하면서 구체화되기 시작하였다. 비록 60bps라는 저속의 통신방법이지만 전력선을 이용함으로써 추가적인 배선 없이도 통신이 가능한 혁신적인 방법이었으며, 가전 장비를 제어하기에는 고속의 통신이 필요 없었기 때문에 가전 기기의 자동 및 원격제어, 가정에 쓰이는 에너지(전기·가스등)의 효율적 이용, 각종 방법 등의 목적에 이용되었다. 사람들은 이러한 기술을 가정자동화 (Home Automation)[3]라고 불렀다. 그러나 X-10 기술은 워낙 느린 통신이었기 때문에 가전기기의 세부적인 제어보다는 열고 닫거나 끄고 켜는 정도의 제어 정도로 쓸 수밖에 없었고, 전문가의 개입 없이는 설치나 변경이 불가능하였으며 너무 많은 비용을 요구하였기 때문에 일부 부유층의 전용물로서만 사용되고 일반인들 속으로는 파고들지 못하였다.

이후 가전 장비들의 기능이 다양화/고급화되면서 거의 모든 가전 장비에서 컴퓨터 칩을 이용하고 있다. 자연스럽게 가전 장비자체를 통신망에 붙여 넣을 수 있는 기본을 형성하였고 가정용 컴퓨터 보급과 초고속 인터넷의 확산, 가전 장비의 지능화가 바로 스마트 홈을 위한 기본을 갖춰준 셈이다.

본 논문에서는 스마트 홈 미들웨어인 OSGi를 기반으로 하는 Smart Home Middleware와 Smart Home Service Bundle

2. 스마트 홈 미들웨어의 필요성

가정 뿐 아니라 다시 가정과 가정을 연결하는 네트워크를 하나의 시스템으로 보면, 이 시스템은 이러한 통신망에 연동되어 다양하게 산재되어 있는 자원들을 이용할 수 있도록 분산 컴퓨팅으로 이용될 수 있다. 그러나 현대 시스템 개발은 애플리케이션 및 클라이언트가 모두 객체를 통하여 구현되고, 따로 저장되는 일이 많아졌다. 이러한 환경에서의 분산 컴퓨팅을 위해서는 응용 프로그램간의 호환성 뿐 아니라 데이터의 호환성, 그리고 이종의 프로토콜간의 연동이 필수적이다. 따라서 이를 위한 소프트웨어가 바로 미들웨어이다.

여러 종류의 스마트 홈을 위한 미들웨어 기술(UPnP(Universal Plug and Play)[4], JINI[5], HAVI(Home Audio Video Interoperability)[6], IEEE 1394[7], 전력선 통신기술[2] 등)이 생겨나면서 장비들을 조정하고 장비간의 상호작용을 원활히 하는 등의 유효한 서비스들이 지원되었으나 각 미들웨어 간의 상호작용을 할 수 없어 UPnP 가전기기는 UPnP 가전기기간에 JINI 가전기기는 JINI 가전기기간에 상호 지원되며 새로운 서비스의 배포가 어려운 점 등의 문제들을 해결하고 서비스가 작동하기 위한 제반 환경 제공을 목표로 하는 것이 바로 OSGi[8]라고 할 수 있다.

OSGi는 서비스가 작동하고 관리되어지는 서비스 환경에 관한 표준이다. 현재와 같이 초기단계의 스마트 홈 시장에서는 아직까지 장비제어 정도의 기술이 주로 사용되고 있지만 많은 장비들이 상호 작용할 시점에 이르면 미들웨어 시장이 활성화될 것이다. 또한 더 나아가 서비스가 점차적으로 고도화되어 필수적 OSGi와 같은 서비스 환경이 될 것이다. 오디오/비디오쪽 미들웨어의 표준인 HAVI나 전력선 제어 쪽의 미들웨어와 OSGi가 연동되어진다면 OSGi의 활용성이 더욱 높아질 것이다. 스마트 홈의 UPnP, HAVI JINI와 같은 미들웨어가 주로 장비의 제어나 데이터 전달 등을 처리하게 되므로 OSGi는 상호 보환적인 위치에 있는 셈이다[그림 1].

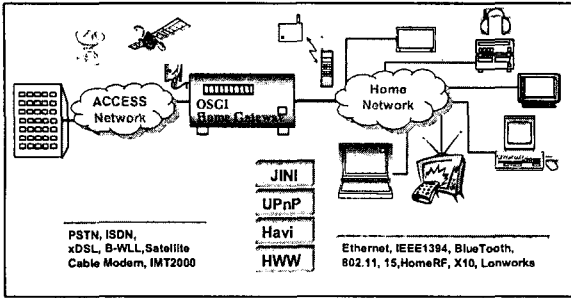


그림 1 OSGi 개관

따라서 본 저자들은 OSGi를 사용하거나 구현하는 쪽에서는 이러한 미들웨어의 지원이 또 다른 고려 사항을 유의하여 연구를 하였다.

2. 관련연구

2.1 IEEE 1394

IEEE 1394 기술은 1995년에 IEEE 표준화기구에 의해 처음으로 확정되었으며, 이의 보완 표준인 IEEE 1394-2000을 통해 400Mbps의 전송속도를 안정적으로 지원하는 고속 직렬 버스 통신 기술이다.

IEEE 1394 같은 경우는 오래 전부터 연구되어 오던 것으로 홈 네트워크 미들웨어로서 사용되어 구현된 경우들을 볼 수 있다.

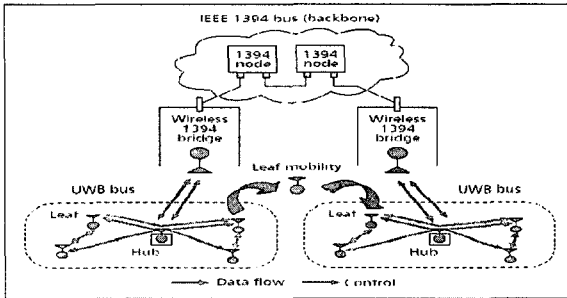


그림 2 IEEE 1394기반의 UWB 홈 엔터테인먼트와 컴퓨팅 네트워크를

위한 계층적 구조[10]

[그림 2]에서는 UWB Bus는 중앙 제어기인 hub station working을 사용한 star topology를 보여주고 있다. 이 hub station은 UWB Bus의 관리와 유지를 기대할 수 있다. 또한 UWB 버스는 무선 1394 Bridge를 사용하여 케이블 1394 백본 네트워크에 접속할 수 있다.

2.2 UPnP

UPnP는 스마트 홈에서 intelligent appliances, wireless devices, PC와 같은 다양한 home device들 사이의 인터넷 워킹을 보다 쉽게 지원하기 위하여 고안된 것이다. 1995년 Microsoft사에서 발표한 Plug and Play의 확장된 형태로서 진일보하여 UPnP는 다양한 device의 networking에 대하여 zero-configuration을 지원한다. 즉, device를 발견하고 device를 통하여 사용 가능한 서비스를 탐색한 후 UCP(Universal Control Point)를 사용하여 각종 home appliance들을 제어하는 목적으로 고안된 것이다.

[그림 3]는 UPnP를 활용한 홈 네트워크 시스템으로서 홈 서버는 네트워크 데이터베이스 같은 서비스들을 제공하고 홈 서버 스케줄 시간테이블을 제공한다. 홈 서버 스케줄을 사용하여 집안의 모든 가전기기를 사용자가 관리할 수 있게 한다. 좀 더 자세하게 홈 서버 프로그램을 보면 각각의 가전제품은 서로 다른 가전제품에 메시지를 보내고 데이터 스트림을 제어할 수 있다 [11].

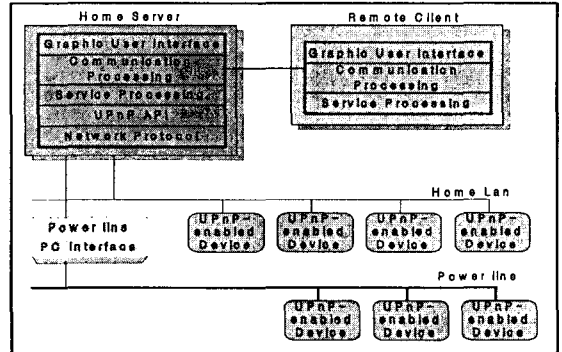


그림 3 Home Network System 구조[11]

표 1 홈 네트워크 서버 프로그램과 클라이언트 프로그램상의 data class[11]

Server Program CLASS NAME	내용
ClogInfo	log 정보를 관리
CuserInfo	사용자 관리
CscheduleInfo	가전제품의 스케줄 서비스 관리
CUHome	가전제품의 위치정보와 집안의 정보 관리
CdevInfo	가전제품 각각의 정보를 관리
CsvcInfo	가전제품 각각의 서비스를 관리
Client Program CLASS NAME	내용
Udevice	냉장고, TV 등과 같은 UPnP용 가전제품 정보
Property	각각의 가전제품의 property의 구성
service	각각의 가전제품의 다양한 서비스들의 구성
Variable	다양한 가전제품 정보

3. OSGi를 이용한 Smart Home Service Bundle Architecture

OSGi 디바이스 접근 기술은 산업 표준 인지 메서드 통합을 위한 강력한 메서드를 제공한다. 자바 인터페이스와 자바의 함수 전체를 표현한 APIs를 제공하여 OSGi는 이 인지 메서드들을 쉽게 접근할 수 있도록 만든다. OSGi 프레임워크와 인지 기술의 통합은 import/export 모델에 기반하고 있다. 등록된 OSGi 디바이스들과 서비스들은 프레임워크 밖으로 전파되고 그들은 특정 응용 프로그램에 내재된 인지 기술을 사용하여 찾아진다. 예를 들어 OSGi Location Service Bundle은 OSGi 프레임워크를 안에서 OSGi Device Driver에 접근하여 Location Service Bundle이 제공하는 Camera On/Off, Image Frame, Movement, Zoom-in/Out 등 서비스를 Mobile 장치 등을 통해서 제어 및 확인 할 수 있다[그림 4].

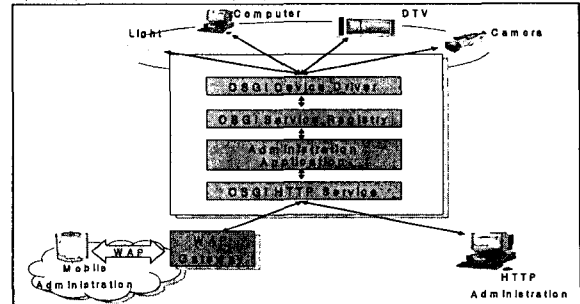


그림 4 OSGi기반의 Smart Home Service Bundle Architecture

그리고 [그림 5]를 보게 되면 OSGi Device Driver를 제공하지 않는 Device를 제어하기 위한 Smart Home Service Bundle Architecture를 설계한 것이다. 이 같은 경우 Registry 부분이나 Administration Application, OSGi HTTP Service은 [그림 4]와 같은 서비스를 지원하도록 설계하였고 그림에서 봐도 알 수 있듯이 서로 다른 부분은 Device를 제어하기 위해서는 각 Device가 지원하는 미들웨어 기반의 Driver에 접근하여 OSGi Driver Transformation 단계를 거쳐서 마치 타 미들웨어의 서비스들을 OSGi 서비스들처럼 사용할 수 있게 한다.

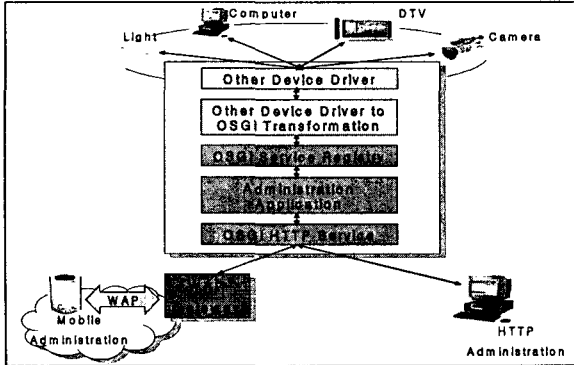


그림 5 타 미들웨어 기반의 Driver를 OSGi기반 Driver로 변환과정의 Smart Home Service Bundle Architecture

3.1 Device Access Services

서비스 플랫폼은 여러 다른 벤더들로부터의 서비스들과 디바이스들이사용자가 어떻게 예약된 서비스들을 등록하고 취소 할 것인지, 새로운 디바이스 설치서비스들을 디바이스들과 조화롭게 임플리케이션 할 것인지, 하드웨어에 디바이스 부분을 어떻게 연결할지를 보는 부분이다.

디바이스 접근 서비스는 디바이스의 몇몇 형식을 표현한다. 이것은 하드웨어 장치를 표현할 수 있다. 그러나 이것을 요구하는 것이 아니다. 디바이스 서비스는 많이 다르다. 예를 들어 어떤 것은 각각의 물리적 장치를 표현하고 다른 것은 통합 네트워크를 표현한다. 몇몇 디바이스 서비스들은 심지어 추상의 다른 단계에 물리적 디바이스를 유사하게 표현한다.

디바이스 서비스는 프레임워크에 등록되어있는 일반적인 서비스로 정의된다. 디바이스 서비스가 등록될 때, 부가적 특성은 디바이스 매니저와 잠재적 유저에 의해 디바이스 기술이 설정될 것이다.

표 2 디바이스 서비스 등록에 필요한 부가적 특성

Device 기술	내용
DEVICE_CATEGORY	<ul style="list-style-type: none"> 이 서비스는 디바이스 매니저에 의해 디바이스 서비스 String[] 타입
DEVICE_DESCRIPTION	<ul style="list-style-type: none"> 사용자 디바이스 String 타입
DEVICE_SERIAL	<ul style="list-style-type: none"> 디바이스를 위한 독특한 시리얼 넘버 디바이스 하드웨어가 시리얼 넘버를 포함한다면, 드라이버 번들은 이 속성으로서 지정 String 타입
service.pid	<ul style="list-style-type: none"> org.osgi.framework.Constants.Device 서비스에 정의된 Service PID는 이 속성으로 설정 메시간 동일한 하드웨어는 동일 PID를 사용하여 플러그인.

3.2 Driver Service

드라이버 서비스는 디바이스 매니저의 제어 아래 정확한 디바이스 서비스 연결에 책임이 있다. 그러나 디바이스 서비스에 연결할 수 있기 전에, 제어를 위해 다른 드라이버 서비스들과 경쟁 해야만 한다.

하나의 드라이버 서비스가 여러 드라이버 서비스 중에서 선택이 되어지면, 그것은 디바이스 카테고리-spec 방식에 디바이스를 연결 해야만 한다. 드라이버 서비스는 다른 디바이스 서비스 사용을 어떻게 사용하는지 나와 있지 언급되어 있지 않으며 다른 디바이스 카테고리에 새로운 디바이스 서비스들이 하나 또는 그 이상 등록되어 있을 것이다.

모든 서비스들과 같이 드라이버 서비스는 번들에서 구현된다. 그리고 프레임워크에 드라이버 서비스 객체 하나 또는 그 이상 등록된 디바이스 매니저에 의해 인식된다. 드라이버 번들은 BundleActivator.start 구현에 적어도 하나의 드라이버 서비스가 등록되어야만 한다.

[그림 5]에서와 같이 Other Device Driver to OSGi Transformation 부분에서 하는 일은 타 미들웨어를 지원하는 Device Driver 번들이 어떤 서비스가 OSGi 서비스로 등록되었는지를 알릴 수 있게 지원을 해주며 현재 OSGi가 지원을 하고 있는 Jini나 UPnP 같은 미들웨어의 Device Driver 경우는 OSGi 프레임워크에서 Jini와 UPnP 서비스들을 찾아 import해 마치 OSGi의 서비스 번들처럼 사용 가능하다. OSGi가 지원을 하지 않고 있는 미들웨어 같은 경우는 Device Driver를 패키지 형태로 만들어 번들로서 서비스를 지원하도록 변환하도록 설계되어 있다. 또한 OSGi기반의 Device Driver를 타 미들웨어 기반에서도 Device 제어가 가능하도록 서비스를 지원하도록 한다.

4. 결론 및 향후 연구

본 논문에서는 OSGi를 기반으로 하는 Smart Home Middleware와 Smart Home Service Bundle의 아키텍처 설계를 기술하고 있다. 앞으로 Smart Home Service Bundle에 관한 연구를 통해서 다양한 가전기기의 제어하는데 도움을 줄 것으로 기대 된다.

향후 연구로는 보다 나은 스마트 홈 시스템 구축을 위해서 다음 몇 가지 연구의 개선 및 추가적인 연구가 계속 될 것이다.

- ① OSGi 상에서 효과적인 Service Bundle 관리를 위한 연구
- ② 핸드폰, PDA, Web Pad 등을 이용한 다양한 방법의 디바이스 상황정보 전달 방법의 구현에 대한 연구
- ③ 통합적인 시스템 관리를 위하여 센서, 기기들의 OSGi 미들웨어와의 연계 상용에 대한 연구
- ④ 사용자 인터페이스의 보다 효율적인 가시화 방안

참고 문헌

[1] Valtchev, D.; Frankov, I.; "Service gateway architecture for a smart home", Communications Magazine, IEEE , Volume: 40 , Issue: 4 , Pages:126 - 132, April 2002

[2] Ferreira, H.C.; Grove, H.M.; Hooijen, O.; Han Vinck, A.J., "Power line communications: an overview", AFRICON, IEEE AFRICON 4th , Volume: 2 , Pages:558 - 563 vol.2, 24-27 Sept. 1996.

[3] Ryan, J.L.; "Home automation" Electronics & Communication Engineering Journal, Volume: 1, Issue, Pages:185-192, July-Aug. 1989.

[4] UPnP Specification v1.0, <http://www.upnp.org>.

[5] Jini Specification v1.0, <http://www.jini.org>.

[6] HAVi Specification v1.1, <http://www.havi.org>.

[7] IEEE 1394 Specification v1.0, <http://www.1394ta.org>.

[8] OSGi Specification v. 3.0, <http://www.osgi.org>.

[9] SMF Toolkit Specification, <http://www-306.ibm.com/software/wireless/smf/>

[10] Nakagawa, M.; Honggang Zhang; Sato, H.; "Ubiquitous homelinks based on IEEE 1394 and ultra wideband solutions" Communications Magazine, IEEE , Volume: 41 , Issue: 4 , Pages:74 - 82, April 2003

[11] Dong-Sung Kim; Jae-Min Lee; Wook Hyun Kwon; In Kwan Yuh; "Design and implementation of home network systems using UPnP middleware for networked appliances" Consumer Electronics, IEEE Transactions on , Volume: 48 , Issue: 4 , Pages:963 - 972, Nov 2002