

OSGi 기반의 스마트 홈 시스템 아키텍처 설계

손민우*, 임철호, 박병철, 신동일, 신동규

*세종대학교 대학원 컴퓨터공학과

e-mail: {minwoo15*, for1soul, leon}@gce.sejong.ac.kr ,
{dshin, shindk}@sejong.ac.kr

Design of a Smart Home System Architecture based on OSGi

Min-Woo Son*, Chul-Ho Lim, Byung-Chul Park, Dong-il Shin,
Dong-kyoo Shin

*Dept of Computer Engineering, Sejong University

요 약

인터넷 사용인구가 전 국민의 3분의 2가 넘었고, 초고속 인터넷 설치가구가 벌써 1,100만 이상에 이르게 되었다. 점점 더 사람들은 디지털 생활에 익숙해져 가고 있고, 디지털을 떠나서 살아갈 수 없게 되고 있다. 이러한 분위기에 더하여 가정내의 가전기기들도 디지털화 되고 있고, 이것들을 네트워크로 연결하여 원격지에서 제어하는 홈 네트워크 관련 기술들이 현실화 되고 있다.

그러나, 여러 가전회사들이 각자의 미들웨어를 만들고 이것을 이용해서 제품들을 만들고 있다. 따라서 스마트 홈(Smart Home)[1]을 구성하기 위한 여러 가지 미들웨어 제안들이 나오게 되었고, 이것들은 계속 혼재할 것으로 예상되어 실제로 홈 네트워크를 구성하는데 많은 혼란이 예상되고 있다. 본 논문에서는 여러 가지 미들웨어를 통합하는 OSGi(Open Service Gateway Initiative)[2,3]를 기반으로 구성된 스마트 홈 시스템 아키텍처를 제안하여 홈 네트워크 구성에 대한 위험 부담이 적고 신뢰할 수 있는 방법을 제시하고자 한다.

1. 소개

최근 스마트 홈에 대한 개념이 발표되어 많은 기업들과 학회에서의 관심이 되어오고 있다. 스마트 홈이란 맥내의 전자기기들과 각종 컴퓨터 시스템이 자동으로 컨트롤 되도록 건설된 집 혹은 사무실을 의미한다. 스마트 홈은 사용자의 다양한 취향에 맞게 모든 기능이 자동으로 조정 가능하다. 예를 들면 자동으로 실내 온도가 조절되거나 출입문 보안 등급의 선택 등이 사용자의 자유로운 선택에 의해서 조정 가능하다.

이러한 스마트 홈은 아래에 명시된 홈 게이트웨이를 포함하는 기술들을 망라한다.

- ① 다양한 종류의 홈 네트워크 접속
- ② 홈 네트워크와 외부 인터넷의 접속
- ③ 가전기기에 대한 무선 컨트롤 및 진단
- ④ 소프트웨어의 확장과 업데이트에 대한 융통성

있는 기술

⑤ 무선 관리의 신뢰성과 안정성

본 논문에서는 먼저 홈 네트워크를 위한 각종 미들웨어 기술 동향에 대해 소개하고, 본 논문에서 언급된 OSGi의 구조에 대해서 서술한다. 이후에 OSGi(Open Service Gateway Initiative)를 기반으로 하는 스마트 홈 시스템 아키텍처의 설계에 대해 논한다.

2. 홈 네트워크 미들웨어

2.1 홈 네트워크 미들웨어의 필요성

현재 초고속망의 폭발적 확산과 다양한 이동 단말뿐 아니라 지능형 가전기기들이 출현함에 따라 가정내에서도 홈 네트워크 구축에의 요구가 늘어가고 있다. 이는 단순히 가전기기를 제어하는 홈오토메이션을 넘어 정보 교환 및 엔터테인먼트의 요구를 충족시킬 수 있어야 한다. 이를 위해서는 다양한 디지털

정보 가전의 개발과 함께, 이 기기들을 연결하여 이를 통제하고, 기기들 간의 데이터를 교환할 수 있는 미들웨어 기술이 필요하게 되었다. 즉, 디지털 가전 제품이 손과 발, 그리고 눈과 귀라 하면, 이를 연결해 주는 미들웨어는 신경조직과 모세혈관이라 할 수 있다.

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.

2.2 홈 네트워크 미들웨어의 기술 동향

2.2.1 UPnP(Universal Plug and Play)

마이크로소프트사가 제안한 미들웨어로서 기존의 IP 네트워크와 HTTP 등의 프로토콜을 사용하여 홈 네트워크 상의 기기들이 제공하는 서비스를 자동으로 발견하고, 사용할 수 있도록 한다.

일반적인 UPnP 구조에서는 Control Point와 UPnP 장치들 간의 제어 명령과 통신만 존재하지만, UPnP A/V 구조에서는 A/V 장치 사이의 직접 데이터 교환이 이루어진다. Control Point는 두 장치 사이의 동기화와 조정만을 담당하고, 실제 A/V 데이터는 HTTP, MPEG 등의 전송 프로토콜을 이용하여 소스(source)와 싱크(sink) 장치 사이에서 직접 발생한다. 일단 데이터의 전송이 시작되면 Control Point를 제거하여도 전송에는 지장을 주지 않는다.

2.2.2 HAVi(Home Audio Video Interoperability)

하부 네트워크 모듈이 IEEE 1394로 제한되어 있다는 것이다. IEEE 1394는 A/V정보를 전송하기 위해 제안되었고, HAVi 역시 A/V기기들을 하나로 묶는 홈 네트워크 기능을 제안하고 있다. IP기반이 아닌 것도 HAVi의 특징이 될 수 있을 것이다. HAVi는 디바이스들을 IEEE 1394에서 제공하는 노드 ID를 사용하여 관리하기 때문에 IP를 지원하지 않는다. 따라서 인터넷과의 연동을 위해서는 다른 방법을 사용해야 한다[5]. HAVi에서는 IEEE 1394에서 지원하는 버스 리셋과 노드 어드레싱을 사용하여

hot plugging과 Plug and Play를 지원하고 있다.

3. OSGi(Open Service Gateway Initiative)

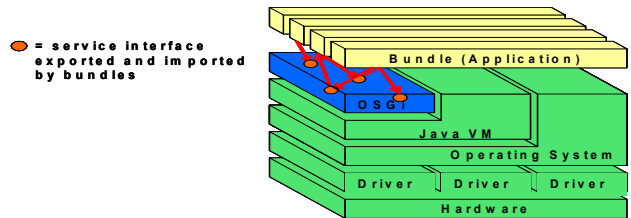
3.1 OSGi 설립 배경

각각의 네트워크에 속한 단말을 접근하기 위해서는 장비에 대한 연결기술(Ethernet, Bluetooth, 무선랜, IEEE 1394 등)이 필요하다. 좀더 섬세한 조절과 장비간의 상호작용을 원활히 하기 위해서는 미들웨어가 필요한데 이것이 UPnP, HAVi, JINI등인 것이다. 이러한 장비 연결 및 제어로 얻을 수 있는 유효한 서비스의 배포문제와 서비스가 작동하기 위한 제반 환경 제공을 목표로 가전/전자, 자동차, 통신, IT등의 다양한 분야의 회원사들이 1999년 3월에 설립된 OSGi는 서비스를 로컬 네트워크나 장비에게 전달하고 전달된 서비스가 운용되는 개방적 표준을 만드는 비영리 표준화 단체라고 할 수 있다.

3.2 OSGi 구조

OSGi Service Platform은 실행할 수 있고 다운로드 할 수 있는 번들로 알려진 서비스 애플리케이션의 배치를 지원하는 안전하고 관리적인 자바 소프트웨어 프레임워크이다 [1,2].

OSGi-compliant gateway는 번들의 필요시에 다운로드와 인스톨을 불필요시에 언인스톨을 할 수 있다. OSGi의 구조는 아래와 같이 사용 가능하다.



(그림 1) OSGi Architecture

- ① 원격 제어와 진단
- ② 동적 소프트웨어 업데이트
- ③ 원격 관리

또한 타 미들웨어의 단점들은 보완과 통합을 통해서 여러 가지 장점을 가지고 있다.

구분	특징	단점
UPnP	<ul style="list-style-type: none"> ■ Scalability ■ Plug and Play ■ 작은 개발량과 적은 Resource 차지 ■ Legacy Systems, Non-IP Devices 통합지원 	<ul style="list-style-type: none"> ■ Windows기반으로 OS 및 Driver들로 인해 실제 Resource 사용량 큼 ■ 서비스 제어를 위한 단계의 복잡
HAVi	<ul style="list-style-type: none"> ■ Peer-to-Peer 방식 ■ 유연성 ■ JRE 방식 제공 	<ul style="list-style-type: none"> ■ 하부 네트워크 제한 ■ IP 지원 안함 ■ 버스의 연결 노드수 제한

<표 1> UPnP와 HAVi의 특징 및 단점

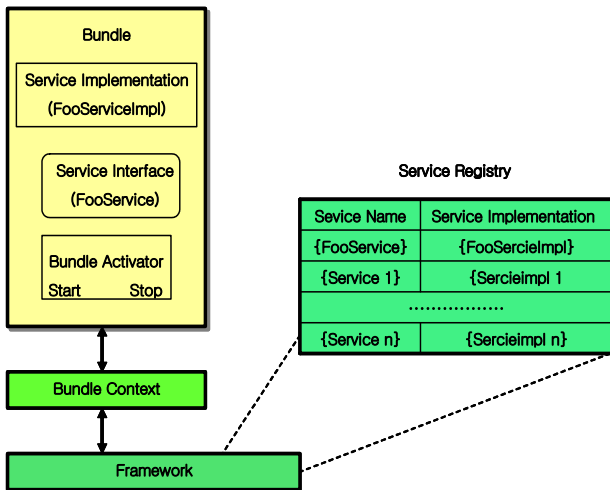
대표적인 장점들은 아래와 같다.

- ① 플랫폼의 독립성
- ② 애플리케이션의 독립성
- ③ 멀티 서비스 지원
- ④ 협동 서비스 지원
- ⑤ 믿을수 있는 보안성
- ⑥ 멀티 네트워크 기술 지원
- ⑦ 편의성

이러한 이유에서 본 논문에서는 스마트 홈 시스템 아키텍처의 기본 프레임 워크로 OSGi를 선택하였다.

3.1 OSGi 프레임워크 및 번들

3.1.1 프레임워크 (Framework)

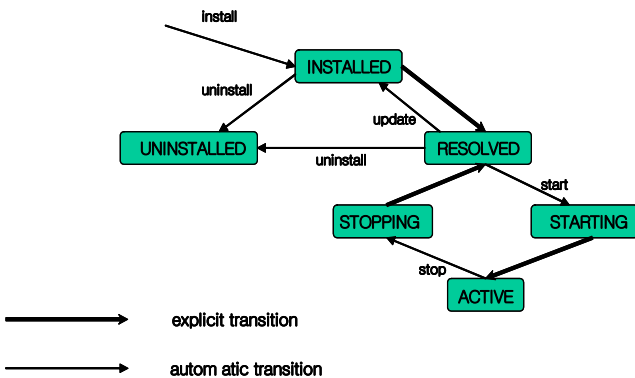


(그림 2) Relationship of the Bundle and Framework

OSGi 환경에서 번들의 설치와 업데이트를 관리하고 번들과 서비스사이의 의존성들을 관리

3.1.2 번들(Bundle)

번들은 프레임워크를 구성하는 가장 작은 단위이다. 헤더들은 SMF에 bundle을 묘사하고 bundle의 의존성을 말한다. 번들은 다른 번들에 의해서 사용할수 있는 SMF에 서비스들을 기록할 수 있다.



(그림 3) Bundle Life Cycle

3.2 Device 관리

홈 네트워크에 새로운 device를 연결하는 예를 생각해보자. device 관리자는 새로운 device를 위해 드라이버를 찾거나 다운로드하는 부분을 담당하나, 정확한 진행은 사용하는 프로토콜에 많이 의존을 한다.

새로운 device가 홈 네트워크에 설치되었을 때, 네트워크는 device를 찾아야만 한다. 새로운 device를 찾아 다루는 두가지 방식을 따른다.

- ① discovery protocol
- ② 게이트웨이의 수동 구성에 대한 틀

Device가 찾아지게 되면 device 드라이버는 device manager를 통해서 게이트웨이에 설치된다. 드라이버는 게이트웨이나 인터넷을 통한 어디서든지 저장될 수 있다.

3.3 SMF 툴킷

SMF(Service Manageme Framework)[6]는 telcos, ISPS등과 같은 서비스 제공자들과 네트워크를 통한 애플리케이션에 대한 소프트웨어 기반 관리 프레임워크이다.

서비스들과 리소스들은 애플리케이션들에 대한 delivery unit처럼 사용되는 파일들인 번들에 패키지화된다. 번들은 패키지 레벨에서 공유하고 있는 클래스와 서비스들을 사용하는 특정 header들과 함께 manifest들을 가지고 있다.

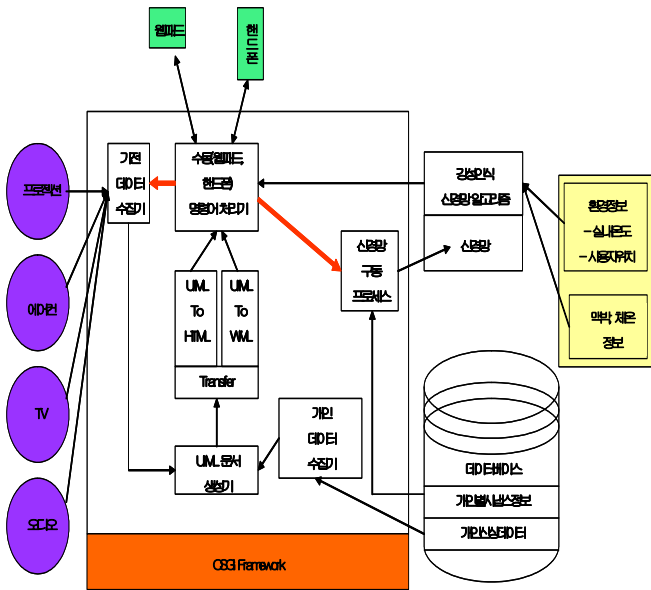
OSGi 스펙은 또한 프레임워크위에서 사용하기 위해 실행되는 애플리케이션들에 대한 HTTP, log permission administration 등 서비스들의 설정을 정의하고 있다[2]. SMF implements들은 아래 표와 같다

서비스	내용
Configuration Admin	실행자가 배정된 번들의 구성정보를 설정하는 것을 허용
Device Access	Device에 대한 드라이버 업데이트나 설치 제거를 제공
HTTP	인터넷이나 다른 네트워크들을 통한 서비스접속
Log	OSGi환경에 대한 메시지 로그서비스
Package Admin	공유하는 패키지들에 대한 정책을 제공하는 것을 management bundle이 허용
Permission Admin	management bundle이 모든 번들에 default들을 제공과 administer 번들의 permission을 허용
Preferences	번들에 대한 영구적인 데이터 저장 제공
User Admin	사용자 인증 제공

<표 2> SMF Implements

4. 스마트 홈 시스템 구축을 위한 OSGi 아키텍처

OSGi를 기반으로 하는스마트 홈 시스템 구축은 아래 그림과 같다.



(그림 4) OSGi 기반의 스마트 홈 시스템 아키텍처 설계

기본적으로 OSGi는 각각의 디바이스들의 정보를 관리하고 디바이스들 간의 의존성 및 정보교환 등을 번들로 만들어서 OSGi에 설치하여 서비스를 지원하며 사용자가 인터넷을 통해서 디바이스를 설치, 제거, 드라이버 업데이트 등이 편할수 있도록 HTTP 서비스와 같은 지원을 할 수 있도록 구현을 할 예정이다.

```

<interface>
package test.service;
public interface EchoService
{
    public String echo (String s);
}
<BundleActivator>
....
public void start (BundleContext context) {
    echoTracker = new ServiceTracker (context,
        EchoService.class.getName(), null );
    echoTracker.open ();
    frame = new EchoFrame(250, 250, echoTracker,
context.getBundle());
}
public void stop (BundleContext context) {
    frame.dispose ();
    echoTracker.close();
} ....
    
```

<표 4> Echo Service Source

5. 결론 및 향후 연구 전망

홈 네트워크 기술은 앞으로 사람들의 삶의 질을 향상시켜줄 기술로서 주목받고 있으며 물리적인 하부 망 기술부터 상위 응용 프로그램까지 다양한 분야에서 활발히 연구되고 있다. 특히 홈 네트워크를 위한 미들웨어 기술 중에서는 여러 가지 미들웨어를 서로 공유 할 수 있는 OSGi를 기반으로한 UPnP, Jini 그리고 LonWorks 기술 등이 활발히 연구되고

있다. 또한 현재 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.

따라서 본 논문에서는 머지않아 모든 사무실이나 집 등에서 상용화될 스마트 홈 시스템 아키텍처를 설계하였다.

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

- ① 핸드폰, PDA, 웹패드 등을 이용한 다양한 방법의 디바이스 상황정보 전달 방법의 구현에 대한 연구
- ② 통합적인 시스템 관리를 위하여 센서, 기기들의 OSGi 미들웨어와의 연계 상용에 대한 연구
- ③ 사용자 인터페이스의 보다 효율적인 가시화 방안

참고문헌

[1] 5. W. Edwards and R. Grinter, "At Home with Ubiquitous Computing: Seven Challenges," Proc. 3rd Int'l Conf. Ubiquitous Computing, Lecture Notes in Computer Science 2201, Springer-Verlag, Berlin, pp. 256~272, 2001

[2] OSGi Specification v. 3.0, March.2003, <http://www.osgi.org>

[3] Condry, M.; Gall, U.; Delisle, P, "Open Service Gateway architecture overview", Industrial Electronics Society, 1999. IECON '99 Proceedings. The 25th Annual Conference of the IEEE , Volume: 2 , 29 Nov.-3 Dec. 1999

[4] Miller, B.A.; Nixon, T.; Tai, C.; Wood, M.D.; "Home networking with Universal Plug and Play", Communications Magazine, IEEE , Volume: 39 , Issue: 12 , Pages:104 - 109, Dec. 2001

[5] Wendorft, R.G.; Udink, R.T.; Bodlaender, M.P.;" Remote execution of HAVi on Internet-exabled devices", Consumer Electronics, IEEE Transactions on , Volume: 47 , Issue: 3 , Pages:485 - 495, Aug. 2001

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