

OSGi Service Gateway Specification 1.0에 기반을 둔 홈 오토메이션 컨트롤러용 개방형 서비스 게이트웨이의 구현

임동찬^o 김태형^o
한양대학교 컴퓨터공학과
{dcim, tkim}@cse.hanyang.ac.kr

The implementation of Open Service Gateway for Home Automation Controller based on OSGi Service Gateway Specification 1.0

Dong-Chan Im^o Tae-Hyung Kim^o
Dept. of Computer Science & Engineering, Hanyang University

요 약

홈 네트워크에 대한 관심이 높아지며 이를 실현하는데 필요한 Residential Home Gateway를 구현하기 위해 2000년 5월 OSGi에서 Open Service Gateway Specification을 발표했다. 본 논문에서는 이에 따른 Home Automation Controller를 구현하기 위한 방법을 논하고 있다. 먼저 Home Automation Controller의 개념에 대해서 알아보고 OSGi의 배경과 Open Service Gateway Specification의 내용을 살펴본다. 그리고 specification에 따른 Home Automation Controller용 개방형 서비스 게이트웨이를 구현하는 방법에 대해 고찰해 본다.

1 서론

인터넷 환경이 성장함에 따라 최근 들어 홈네트워킹에 대한 관심이 높아지고 있다. 이에 따라 여러 가지 홈 네트워킹을 위한 프로토콜이 개발되고 있고 다양한 연구가 시도되고 있다. 1999년 홈 네트워킹에 관심을 보이는 세계 여러 기업들이 모여 조직된 비영리 단체인 OSGi(Open Service Gateway Initiative)는 service gateway에 대한 specification을 발표하였다. 이는 자바의 플랫폼 독립성과 실행코드의 네트워크 이동성을 이용하여 소용량 메모리 디바이스를 위한 동적인 서비스의 제공을 목표로 제정된 표준이다. 이 specification에 정의된 인터페이스와 클래스를 바탕으로 Home Automation Controller용 개방형 서비스 게이트웨이를 구현하는 방향에 대한 연구가 본 논문의 목적이다.

2 관련 연구

2.1 Home Automation Controller

Home Automation Controller는 가정 내부의 각종 기기들을 제어하기 위한 제어 장치이다. 홈 네트워킹의 전체적인 구조에서 Residential Home Gateway라고 정의되는 service gateway는 외부 네트워크망과 가정 내에 내부 네트워크망 사이에 위치하여 중개자로서 두 네트워크를 연결해주고 서비스를 가능하도록 해주는 역할을 한다. 이는 기존의 ADSL 모뎀이나 전용 라우터처럼 단순한 인터넷 액세스만을 담당하는 것이 아닌 가정 내에 기기들에 대해 기존에 서비스를 제공하거나 앞으로 나올 서비스에 대해 확장 가능한 기능을 제공하는 역할을 담당한다. 이는 그림 1에서 Home Server에 해당한다. Home

Server는 외부 네트워크망으로부터 가정 내의 기기들에 각종 서비스를 전달해주는 기능을 한다. 여기서

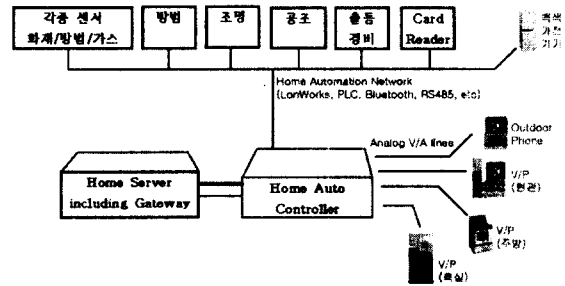


그림 1 Home Automation Controller의 인터페이스

가정 내에 기기들을 제어하는 기능은 Home Automation Controller가 담당하게 된다. 본 논문에서 구현하고자 하는 Home Automation Controller는 홈 네트워킹을 구현에 있어 Home Server와 연결되어 각종 센서 및 방범, 조명, 공조 등과 같이 가정 내의 기기들을 제어하기 위한 서비스를 제공한다.

2.2 OSGi

최근 들어 인터넷 환경이 급성장하고 홈 네트워킹에 대한 관심이 높아짐에 따라 service gateway에 대한 표준

이 요구되었다. 이에 Sun, IBM 등의 기업들을 중심으로 1999년 OSGi가 조직되었고 2000년 5월에는 OSGi에서 Open Service Gateway Specification 1.0을 발표하였다. 이는 Java의 platform에 독립적인 실행특성과 보안등의 이유로 Java에 기반을 두고 있다. specification에는 홈네트워킹을 실현하기 위해 요구되는 service gateway에 대한 specification으로서 service gateway로서 만족해야 하는 인터페이스와 클래스에 대해 명시하고 있다. 버전이 1.0이므로 모든 것이 확실하고 완벽하게 정의되지는 않았지만 이미 여러 기업에서 specification을 만족하는 솔루션을 개발한 상태이고 홈네트워킹의 표준으로서 자리잡고 있다.

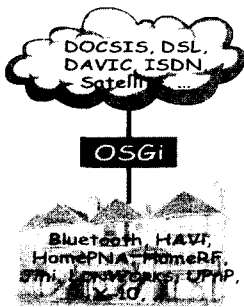
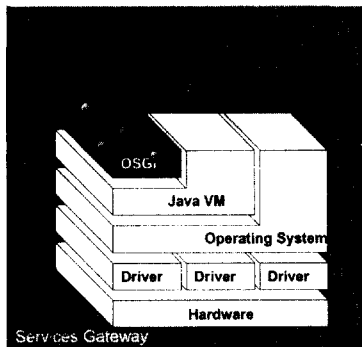


그림 2 OSGi의 개념적 위치

그림 2에서 보는 것처럼 OSGi는 외부 네트워크망과 내부 네트워크망에 사용되는 다양한 프로토콜간의 bridge 역할을 하며 서비스의 delivery 기능을 담당한다. 가정 내에서 Bluetooth, Jini, UPnP, HAVI, HomePNA 등 어떤 프로토콜이라도 관계없이 OSGi Service Gateway Specification을 만족하는 service gateway라면 지원이 가능한 것이다. 그러므로 OSGi는 기존의 홈네트워킹을 위한 프로토콜들을 지원하고 외부 네트워크망으로부터 service를 제공하는 미들웨어로서 Bluetooth나 HAVI와는 다른 레벨의 미들웨어라고 할 수 있을 것이다.



● = service interface exported and imported by bundles

그림 3 Service gateway의 구조

그림 3에서는 OSGi Service Gateway Specification에서 정의하는 service gateway의 구조에 대해 나타내고 있는

데 OSGi framework은 운영체제 상위의 자바 가상 머신을 기반으로 위치하게 된다.

OSGi에서 새로 정의된 개념을 살펴보면 서비스(service)는 특정 기능을 수행하는 자바 인터페이스와 실제 구현 객체를 말하고, 번들(bundle)은 그러한 서비스를 제공하기 위한 기능적 배포단위이다. 번들 컨텍스트(bundle context)란 번들의 life cycle을 관리하는 번들의 실행환경을 의미한다.

OSGi framework 패키지에 따르면 번들의 life cycle에서 나타나는 번들의 상태는 모두 6가지로 나뉘어 지는데 installed, uninstalled, resolved, starting, stopping, active의 상태가 있다. 그림 4에서 번들의 life cycle에 따른 번들의 상태 변화를 설명하고 있다.

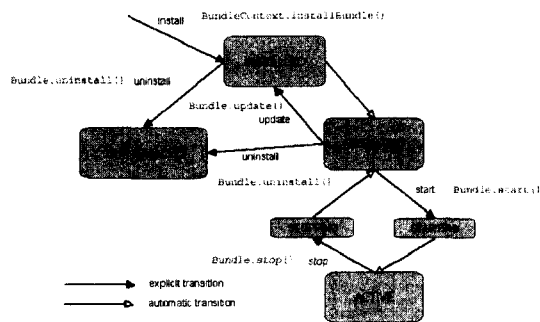


그림 4 번들의 life cycle

OSGi framework에서 동작하는 특정 기능을 수행하는 서비스 번들을 만들기 위해서는 서비스를 디자인하고 그에 맞는 interface를 작성한다. 그리고 작성한 interface에 맞게 자바 코드로서 이를 구현하고, framework package의 BundleActivator 인터페이스를 구현하게 되는데 이 단계에서 서비스 번들이 시작될 때와 중지될 때 실행되어야 할 것들을 start, stop method에 구현해 넣어야 한다. 그 후 번들에 대한 manifest 파일을 작성해야 하는데, manifest 파일에는 해당 번들이 제공하는 서비스와 이 번들의 실행을 위해 필요한 다른 서비스 등을 명시한다. 그런 후 작성한 결과물을 jar 파일로 묶어 하나의 번들로 배포하게 된다.

2.3 OSGi Framework의 실제 구현

OSGi framework은 번들간의 관계를 관리하고 번들의 install, uninstall, update 등의 기능을 수행한다. 이에 대한 세부사항은 OSGi Service Gateway Specification에 명시되어 있다. 그림 5에서 framework과 설치된 번들간의 상호작용을 설명하고 있다. 번들이 framework에 설치되면 그림 5의 오른쪽에 보이는 것처럼 Service Registry에 해당 번들이 설치되었다는 정보가 기록되고 해당 번들에서 제공하는 서비스를 등록시키게 된다. 이렇게 등록된 서비스들 중 다른 번들에서 필요로 하는 서

비스가 있다면 framework은 해당서비스를 탐색할 수 있도록 LDAP(Lightweight Directory Access Protocol) query 기능을 제공한다.

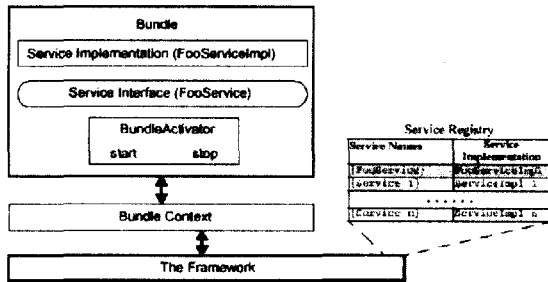


그림 5 Framework과 서비스 번들의 상호작용

이와 같은 구조로 OSGi Service Gateway Specification에 명시된 인터페이스나 클래스들을 기능에 맞게 코딩함으로써 framework을 구현할 수 있다. SUN의 Java Embedded Server와 같이 이미 구현된 framework을 작동시켜 보면 각각의 기능을 확인할 수 있고, Service Tango와 같은 Open Source의 구조를 참조 연구함으로써 service gateway의 framework'부분의 구현을 완성할 수 있다. 본 논문에서 제시하는 방법에 따라 현재 service gateway 구현 프로젝트가 진행 중에 있다. OS로는 내장형 리눅스(Embedded Linux)를 사용하고, JVM은 JDK 1.3을 환경으로 하고 있다. OSGi는 자바 가상 머신 위에서 platform에 관계없이 구동되어야 하므로 OS는 크게 고려하지 않아도 된다. 다만 device manager 구현에 있어 디바이스의 드라이버들은 platform에 종속적이므로 이에 대한 처리가 필요할 것이다. 진행중인 OSGi Service Gateway Specification 1.0에 따른 service gateway 구현 연구의 framework의 구조를 살펴보면, 먼저 처음 framework이 기동되면 이전에 framework이 종료되기 전 상태에서 인스톨되었거나 실행 중이었던 번들의 상태를 복구하기 위해 캐쉬된 정보를 읽어들이어 framework을 이전 상태로 복구시킨다. 그 후 번들을 install, uninstall, update, start, stop시키거나 framework을 종료하거나 현재 framework의 상태를 나타내주는 기능을 수행하는 명령들을 사용자에게 제시해 주고 특정 명령이 들어오기까지 대기한다. 각각의 명령이 사용자로부터 입력되었을 때 이를 처리하기 위해, framework을 구현을 위한 핵심부분에는 번들의 install, start, stop과 같은 기능을 담당하는 method가 OSGi Specification에서 명시된 interface를 바탕으로 구현되어 있다. OSGi Specification에서 명시된 인터페이스의 구현은 프로그래머에 따라 자유롭게 할 수 있다. 그리고 framework에는 번들에서 제공하는 서비스를 얻기 위한 method와

framework에서 번들과의 관계에서 발생하는 event에 관련하여 event와 event listener에 관련된 method의 기능도 구현되었다. 그 외에 필요에 따라 자유롭게 framework을 구동시키기에 관련된 유틸들을 작성하여 포함시킬 수 있다.

다음으로 service gateway에서 가정 내 device들에 대해 자동으로 인식하고 그에 따른 driver를 동적으로 업데이트 하기 위한 device manager 부분을 구현해야 한다. framework 부분과는 다르게 device manager의 구현은 아직 OSGi service gateway framework에서 작동하게 될 device들의 device service가 발표되지 않았고 기존의 device에 대한 device service와 driver가 제공되지 않은 상태이므로 device manager를 구현한 뒤 다양하게 테스트할 수는 없다. OSGi service gateway framework에서 작동되는 device 번들은 표준 자바 인터페이스로서 정의되어야만 한다. 그러나 현재 상태에서 device manager를 구현하고 기존의 device를 사용해서 device manager의 기능을 테스트 하고자 한다면, 기존의 특정 platform에 맞는 device driver의 native code를 call할 수 있도록 Java의 wrapper로서 device 번들을 구현할 수 있다. 이와 같은 방법으로 구현된 service gateway framework는 윈도우즈 OS에서 동작하게 된다면 윈도우즈용 device driver들도 device manager에서 지원하게 되는 것이다. 이 부분에 대해서 차후 더 많은 연구가 필요할 것이다.

3 결론

본 논문의 연구로부터 최종적으로 만들어질 홈 컨트롤러 box에는 RS-232C, RS-485와 ethernet 등을 위한 포트를 장착하게 되고 Bluetooth나 LonWorks등의 프로토콜도 지원하게된다. 여기에 프로세서와 메모리등이 base board에 탑재된다. 이러한 하드웨어 환경 위에 내장형 Linux가 동작하게 되고, 그 위에서 홈네트워킹을 위한 application이 실행될 수 있도록, 위와 같이 구현된 service gateway가 동작하게 된다.

본 논문에서는 OSGi Service Gateway Specification 1.0에서 명시된 OSGi Service Gateway Framework를 구현하여 Home Automation Controller용 개방형 서비스 게이트웨이를 구현하는 방법을 살펴보았다.

향후 연구 방향으로는 Service Tango와 같은 Open Source 프로젝트를 참고로 개선된 OSGi framework을 개발하고 OSGi Specification에 명시된 framework의 device manager부분의 구현에 대한 연구가 필요하다. 또한 디바이스 서비스이외에 http 서비스와 log 서비스의 구현도 앞으로의 과제로 남아 있다.

4 참고문헌

[1]. Open Service Gateway Initiative. "OSGi Service Gateway Specification 1.0". <http://www.osgi.org>. 2000.
 [2]. Gerard O'Driscoll. "The Essential Guide to Home Networking Technologies". Prentice-Hall. 2001.