

# OSGi(Open Service Gateway Initiative) 기반의 효율적인 서비스 번들 관리를 위한 SBM(Service Bundle Manager)의 설계 및 구현

김 남 훈<sup>†</sup> · 손 민 우<sup>\*\*</sup> · 신 동 규<sup>\*\*\*</sup> · 신 동 일<sup>\*\*\*\*</sup>

## 요 약

현재 여러 종류의 홈 네트워크(home network) 기술들이 존재하고 있으며, 단일화된 홈 컴퓨팅(home computing) 환경을 위해서는 서로 다른 네트워크 프로토콜 간의 연동이 필수적이다. 이를 위하여 여러 종류의 홈 네트워크 미들웨어(home network middleware)가 개발되었고, 특히 OSGi(Open Service Gateway Initiative)는 다양한 미들웨어 환경 사이의 서비스 상호연동을 제공하기 위해 제안되었다. 그러나 OSGi의 실행환경인 프레임워크/framework)는 수동적으로 서비스 번들(service bundle), 사용자 및 장치 관리 등을 수행하는 취약점을 가지고 있다. 본 논문에서는 사용자들의 편의성을 위해 수동적으로 관리되는 부분들의 문제점을 개선하고자 한다. 이를 위하여 자동화 서비스와 효율적인 서비스 번들의 관리를 위한 SBM(Service Bundle Manager)의 구조를 제안하고, SBM을 구현하여 SBM의 실효성을 검증하였다.

키워드 : 홈 네트워크, OSGi, 홈 네트워크 미들웨어, 서비스 플랫폼, 서비스 번들

## Design and Implementation of the SBM(Service Bundle Manager) for the Efficient Service Bundles Management based on OSGi(Open Service Gateway Initiative)

Namhoon Kim<sup>†</sup> · Minwoo Son<sup>\*\*</sup> · Dongkyoo Shin<sup>\*\*\*</sup> · Dongil Shin<sup>\*\*\*\*</sup>

## ABSTRACT

Since different kinds of home network technologies coexist, seamless connections between different kinds of network protocols are essential for home computing environments. Several kinds of home network middleware have been developed and OSGi(Open Service Gateway Initiative) was initiated mainly for interoperability support with service distribution between various middleware environments. Framework, OSGi service platform, has weaknesses of passive service bundle management, user management and device management. Therefore, this paper resolves passive management parts which are critical weaknesses of OSGi Specification Version 3 and proposes SBM(Service Bundle Manager) to efficiently manage various service bundles. Additionally, we verify practical effects of SBM through experiments.

Key Words : Home Network, OSGi, Home Network Middleware, Service Platform, Service Bundle

### 1. 서 론

유비쿼터스 컴퓨팅(ubiquitous computing)[1]은 '어느 곳에서나 존재한다'라는 의미의 유비쿼터스(ubiquitous)와 컴퓨팅(computing)의 합성어로 장소나 시간에 구애 받지 않고, 생활 속에서 자연스럽게 편리하게 컴퓨터를 사용할 수 있는

환경을 의미한다. 즉, 이것은 도처에 편재한 컴퓨터가 장소나 시간에 따라 변하는 정보 서비스를 센싱(sensing)과 트래킹(tracking)을 통해 사용할 수 있다는 것을 의미한다. 유비쿼터스 컴퓨팅의 의미를 보다 포괄적으로 해석하는 경향이 있는데, 대표적인 예가 홈 네트워크(home network)이다[2]. 이는 유무선 네트워크를 통해 인터넷에 접근할 수 있는 오늘날의 정보기반 기반의 컴퓨팅 환경을 의미한다.

홈 네트워크는 가정과 정보 네트워크를 연결함으로써 정보를 효율적으로 유통시킬 수 있도록 지원하며, 각종 공공 서비스 및 사회 서비스(금융·의료 등)에 직접적으로 연결

<sup>†</sup> 정 회 원 : 동양공업전문대학 모바일인터넷과 부교수  
<sup>\*\*</sup> 준 회 원 : 세종대학교 컴퓨터공학부 박사과정  
<sup>\*\*\*</sup> 종신회원 : 세종대학교 컴퓨터공학과 부교수  
<sup>\*\*\*\*</sup> 종신회원 : 세종대학교 컴퓨터공학과 조교수  
논문접수 : 2006년 7월 19일, 심사완료 : 2006년 9월 7일

하여 서비스를 제공할 수 있도록 하였다. 또한 홈 게이트웨이[3]나 홈 서버[4]를 이용하여 정보통신기기, 디지털 AV(digital audio&video)기기 및 기존 가전기기 등을 통합적으로 제어함으로써 가정 생활의 편리함과 효율성을 극대화하였다.

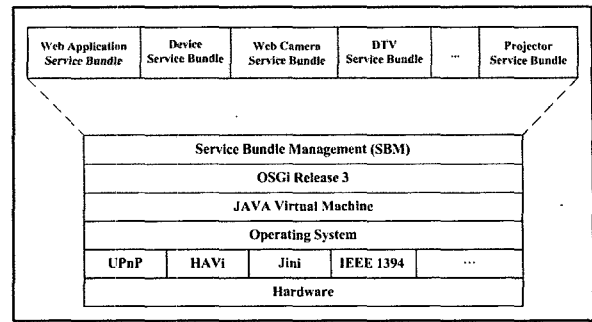
홈 네트워크를 위한 미들웨어(middleware)는 가전기기를 제어하고 전기제품들 사이에 상호작용과 여러 서비스들을 제공한다. 여러 종류의 홈 네트워크를 위한 미들웨어 기술(UPnP(Universal Plug and Play)[5, 6], Jini[7, 8], HAVi(Home Audio Video Interoperability)[9, 10], IEEE 1394[11, 12], 전력선 통신(Power Line Communication) 기술[13] 등)들이 개발되었다. 그러나 이 같은 홈 네트워크 미들웨어는 서로 다른 미들웨어 간의 상호작용을 할 수 없고, 새로운 미들웨어 기반의 서비스 배포 등의 문제점들을 가지고 있다. OSGi(Open Service Gateway Initiative)[14, 15, 16, 17]는 로컬(local) 홈 네트워크를 위한 서비스들을 쉽게 개발할 수 있게 함으로써 위와 같은 문제점들을 극복하기 위해 개발되었다.

OSGi는 점차적으로 홈 네트워크 미들웨어 시장에 세력을 확장해가고 있으며 OSGi 기반의 가전 기기들이 실생활 속에서 사용되고 있다. 또한 이러한 가전 기기들을 제어하기 위해 OSGi 기반의 서비스 번들이 개발되어 실생활에 사용되고 있다. 따라서 사용자는 급격하게 늘어나고 있는 수많은 서비스 번들의 효율적인 관리가 필요하다.

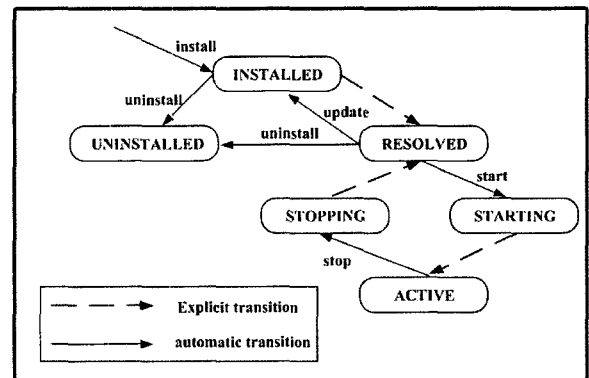
OSGi는 서비스 번들 관리 및 이벤트 처리를 위한 프레임워크(framework)와 업무 처리 관리를 위해 로그 데이터(log data)를 저장하고 관리하는 Log Service, 가전기기 관리자를 정의하고 가전기기 관리자가 새로운 가전기기를 인식해서 그에 맞는 드라이버를 관리하는 Device Access 등의 서비스로 구성된다. 그러나 OSGi Spec. 3 기반의 서비스 플랫폼에서는 여전히 서비스 번들이 사용자에 의해 수동적으로 업데이트, 설치 및 제거 등이 이루어지고 있다. 또한 자동으로 장치(device)의 상태 체크 및 드라이버(driver) 업데이트가 안되고 분산 프레임워크(distributed framework)를 지원하지 않는 등의 문제를 가지고 있다. 따라서 본 논문에서는 서비스 번들의 효율적이며 자동적인 관리가 가능하도록 기존 OSGi Spec. 3 플랫폼 문제점으로 지적된 수동적인 관리 문제를 해결할 수 있는 OSGi 기반의 SBM (Service Bundle Manager)를 제안하고자 한다.

## 2. 관련연구

OSGi는 로컬 네트워크와 집, 차와 같은 서로 다른 환경에 있는 장치들에게 다양한 서비스들을 제공하고 여러 미들웨어의 상호작용 및 서비스 배포 등의 문제를 해결하기 위해 만들어졌다. (그림 1)은 OSGi의 기본 구조를 보여주고 있다.



(그림 1) OSGi 기본 구조



(그림 2) 번들 라이프 사이클

외부 네트워크와 맥내 네트워크를 연결해 주는 OSGi 서비스 플랫폼(service platform)은 OSGi 프레임워크와 OSGi 서비스의 두 부분으로 구분할 수 있다. OSGi 프레임워크는 확장된 자바 런타임(Java Runtime) 환경하에 레지스트리 서비스(registry service), 서비스 라이프 사이클 관리(service life cycle management) 등을 관리하고 OSGi 서비스를 위한 실행환경을 제공한다.

OSGi 서비스는 자바 인터페이스에 의해 정의된다. OSGi 서비스들은 HTTP, Logging, Device Access등을 포함하고 있다. 서비스는 모두 번들(bundle)이라 불리는 물리적 묶음에 포함된다. 번들은 프레임워크를 구성하는 가장 작은 단위이며, (그림 2)에서 보여지는 것처럼 번들은 설치(install), 제거(uninstall), 갱신(resolved), 정지(stopping), 실행(starting) 등의 라이프 사이클을 갖는다. 번들은 자바 클래스 코드 파일(java class code file), 리소스(resource) 등을 포함하고 있으며, 메니페스트 파일(Manifest file)을 포함하고 있어 번들에 대한 메타(meta) 정보나 번들의 설치(install), 시작(start) 등을 사용하기 위해 프레임워크가 필요로 하는 정보 등을 가지고 있다. 복수개의 OSGi 서비스가 하나의 번들에 포함될 수도 있으며, 번들은 배포와 관리의 기본 단위를 형성한다.

현재까지 OSGi Spec. 3을 기반으로 구현된 Oscar의 Module Loader를 통해 앞서 언급한 OSGi 서비스 플랫폼의 문제점을 해결하려는 연구가 진행되었다[15]. IBM에서는 OSGi Spec. 3기반의 번들 관리를 위해 SMF(Service

Management Framework)[16]를 개발하였으며, 그 외에 LONWORKS기반의 기기들을 제어하기 위한 OSGi Spec. 3의 서비스 플랫폼 기반의 LNS(LONWORKS Network Service) [17] 등이 활발하게 연구되고 있다.

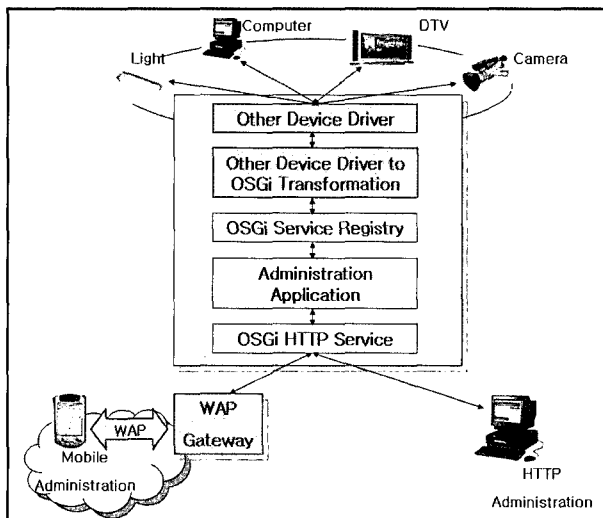
그러나 현재 Oscar나 SMF는 사용자를 위한 번들 관리의 자동 시스템화와 효율적인 장치 및 사용자 관리 부분을 제공하지 못하고 있으며, 필요성을 역설하고 향후 연구 부분으로 밝히고 있다[15, 16].

본 논문에서는 제안할 SBM을 통해 OSGi의 서비스 번들 관리를 자동 시스템화하고 동시에 효율적인 장치 및 사용자 관리 등의 문제를 해결하고자 한다.

### 3. OSGi 기반의 홈 네트워크 게이트웨이 구조 설계

서로 다른 미들웨어 기반의 장치들간 상호연동 및 제어 등의 문제점들을 해결하기 위해 OSGi 기반의 홈 네트워크는 점차 활성화 될 것이다.

(그림 3)은 타 미들웨어 기반의 장치를 제어하기 위한 홈 네트워크 게이트웨이 구조를 설계한 것이다. 이 게이트웨이는 OSGi Service Registry, Administration Application과 OSGi HTTP 서비스 등을 지원하고 있어 홈 네트워크내에 새 장치가 감지 되었을 때, 감지된 장치에 적절한 미들웨어를 선택하고 OSGi Service Registry에 기록된 기존 장치들의 어태치먼트(attachment)를 조정하여 서비스를 제공하도록 설계하였다. 예를 들어 DTV가 플러그인(plug-in) 되었을때 OSGi Service Registry에 등록이 되어있는지 확인해보고 필요에 따라 드라이버를 다운로드하여 서비스를 지원한다. 타 미들웨어 (Jini, UPnP, HAVi 등) 기반의 장치를 제어하기 위해서, 타 미들웨어 기반의 드라이버 모듈을 JAVA 패키지(package) 형태로 서비스 번들을 구현하고 이



(그림 3) 타 미들웨어 기반의 장치(Device)를 OSGi 기반의 서비스로 변환하는 홈 네트워크 게이트웨이 구조

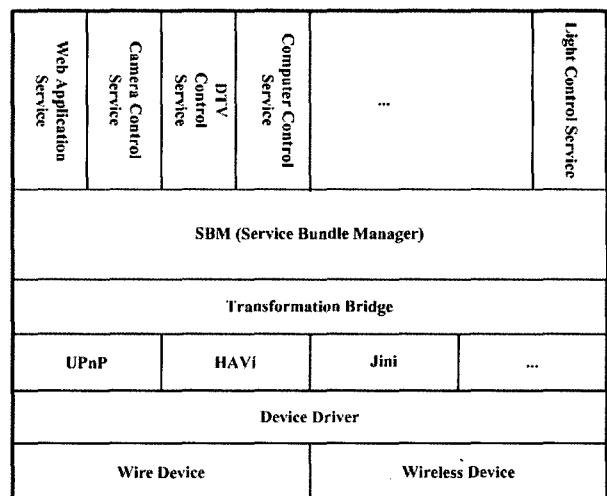
것을 통해 각 장치에 접근한다. Other Device Driver to OSGi Transformation 단계는 여러 종류의 서비스들을 제공하는 타 미들웨어들을 위한 장치 드라이버를 체크한다. 서비스 번들은 OSGi Service Registry 단계를 통해 기록된다. 또한 타 미들웨어 기반의 장치가 게이트웨이와 통신을 요구 시 익스포트(export) 서비스를 제공한다.

### 4. SBM (Service Bundle Manger)의 설계

#### 4.1 SBM 기반의 홈 게이트웨이 구조 설계

홈 네트워크가 널리 확산되어가면서 점차적으로 서비스 번들이 증가하게 될 것이다. 따라서 사용자들은 좀 더 사용하기 편하고 효율적인 서비스 번들 관리를 필요로 하게 된다. OSGi Spec. 3에서 제공하는 서비스 플랫폼으로는 여러 서비스 번들을 관리하기 어렵고, 많은 취약점 들이 존재한다. 따라서 본 논문에서는 서비스 플랫폼의 수동적인 서비스 번들 관리, 사용자 관리 및 장치 관리 지원이 불가능한 취약점들을 보완해서 Web Application Service, Camera Control Service 등과 같은 여러 서비스 번들들을 사용자가 효율적으로 관리하기 위한 SBM을 설계하였다.

(그림 4)는 SBM기반의 홈 게이트웨이 구조를 보여주고 있다. 제일 아래에 위치한 Device Driver와 Wire/Wireless Device는 주로 장비간의 연결방식을 결정짓는 부분으로 표준화가 필수적인 부분이다. 운영체제는 대부분 WinCE나 내장형 Linux, 실시간 OS등을 이용하고 있기 때문에 별도의 표준화에 대한 요구가 적은 부분이다. 현재 장비간의 연결 방식으로는 무선 장치(wireless device)로는 무선랜, RFID(Radio Frequency Identification)[18, 19], Bluetooth[20, 21] 등과 유선 장치(wire device)로는 USB, IEEE 1394, Ethernet 등이 있다. 물리적으로 장치가 연결되면 새로운 장치를 연결하여 사용할 수 있도록 홈 네트워크를 자동으로 재구성하여 통신규약에 따라 장치간의 정보나 제어신호를



(그림 4) 홈 게이트웨이 구조

전달할 수 있도록 해주는 기능이 UPnP, HAVi, Jini 등의 미들웨어이다.

Transformation Bridge부분은 서로 다른 미들웨어 기반의 가전기기간 통신을 위해 사용한다. 이 부분에선 홈 게이트웨이 플랫폼이 지원하는 미들웨어(UPnP, HAVi, Jini 등)가 결정되면 적절한 브릿지(bridge)가 결정된다.

컴퓨터에서 Windows XP나 Linux, Mac 등의 OS와 같이 컴퓨터 사용방식을 결정하는 운영체제가 있듯이 SBM은 홈 게이트웨이의 위치에 놓여 가정의 내부 또는 외부에서 홈 네트워크에 접속하여 다양한 홈 네트워크 서비스를 제공한다.

Web Application Service, Camera Control Service, Device Manager Service 등의 서비스 번들을 제어하기 위해 OSGi기반의 SBM를 사용하게 된다. SBM은 OSGi Spec. 3에 포함된 서비스 플랫폼의 수동적인 서비스부분과 그 외에 사용자/장치 관리 부분의 취약점들을 보완하여 서비스 번들을 제어한다.

### 5. SBM의 구조 설계

(그림 5)는 SBM의 구조를 보여주고 있다. 가전기기를 제어하기 위해서 사용자는 두 가지 방법을 사용할 수 있다.

첫 번째는 웹 브라우저(web browser)를 통해 사용자 인증을 거쳐 서비스 번들 제어가 가능하다. 두 번째는 PDA나 Web PAD와 같은 모바일 장치(mobile device)를 사용하여 보내져 오는 UIML(User Interface Markup Language)[22] 문서를 Network Service를 사용하여 Service Bundle Manager Server에 접근하게 되고 UIML문서는 Service Using History Storage에 저장되고 문서를 분석하게 된다.

여기서 UIML 문서의 데이터(data) 형식은 (그림 6)에서와 같이 가전기기를 크게 영상과 음향으로 나눠 DataSet을 만들고, 이 형식에 맞게 UIML 문서 패턴을 작성한다.

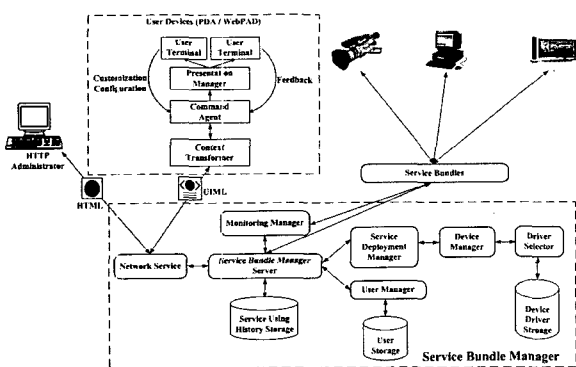
웹이나 모바일 장치를 사용해 가전기기를 제어하기 위해 사용자들이 접근을 하면, 사용자 ID를 통해 User Manager로부터 사용자의 장치별 접근 권한에 관한 자료를 받게 된다. 이것을 통해 Service Bundle Manager Server는 Device Manager에 가전기기 ID 및 나머지 장치의 기능에 관한 자료를 보내 가장 적절한 장치에 접근해서 기능 값을 변경하

고, Service Bundle Manager Server를 통해 서비스 번들을 제어하게 된다. 특히 사용자들의 서비스 번들 관리를 효율적이면서 편리한 사용을 위해 Service Bundle Manager Server는 항상 Monitoring Manager를 통해 보내져 오는 SBM에 설치된 각 서비스 번들의 상태 정보를 받아 Service Using History Storage에 저장을 하고 일정 서비스 번들(냉장고, 보안 서비스 등)같은 경우 항상 작동하도록 관리를 한다. 또한 서비스 번들이 설치되면 사용자의 별도 제어없이도 실행시켜서 사용자가 편리하게 사용할 수 있도록 한다. Monitoring Manager는 일정 간격으로 SBM에 연결된 각 서비스 번들의 상황을 체크하고 Service Bundle Manager Server에 알려준다. 예를 들어 사람이 없는 방 안에 전등이 켜져있다거나, 집안 온도가 적정 온도에 맞추어져 있는데도 보일러 서비스 번들이 가동되는 등의 일들이 발생한다면, Monitoring Manager는 Service Bundle Manager Server에 서비스 번들의 상태를 알려주어서 불필요한 서비스 번들의 사용을 중지시키고, 필요한 서비스 번들의 경우는 실행시킨다. 특히 냉장고, 보안 서비스, 위치 인식 서비스 등과 같은 경우의 서비스는 꺼지면 그 즉시 Service Bundle Manager Server에 알려 실행을 시키며, 서비스 번들의 변동이 있을 시 SBM 관리자의 모바일 장치에 서비스 번들의 상태 변동 사항을 통보해 준다.

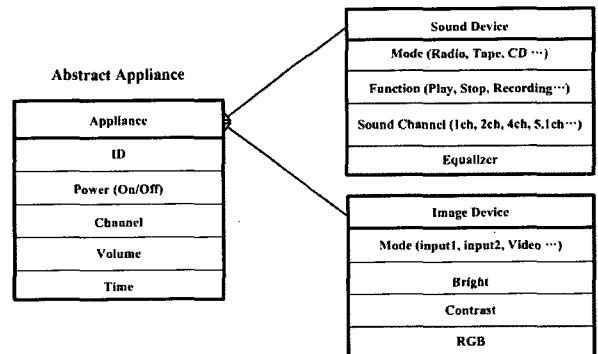
SBM의 저장소(Storage)부분은 3개의 저장소를 가지는데, Service Using History Storage, User Storage, Device Driver Storage이다. Service Using History Storage는 각 디바이스를 제어하기 위해 PDA에서 보내온 UIML을 저장한다. 디바이스 별로 테이블을 만들어 저장된 자료는 추후 상황인식(context awareness) 데이터로 접목되어 활용할 수 있다.

User Storage는 각 사용자의 인적 사항을 저장한다. 각 사용자별로 ID를 부여해서 디바이스 제어 접근 권한을 갖도록 하기위한 정보를 가진다. 사용자별 알맞은 디바이스 서비스(device service)를 제공하는데 필요한 정보이다.

Device Driver Storage는 각 디바이스별 드라이버를 저장하고 있다. 필요한 디바이스의 드라이버를 항상 최신 버전을 기준으로 사용할 수 있도록 제공한다.



(그림 5) SBM(Service Bundle Manager)의 구조



(그림 6) DataSet

SBM을 구성하는 세부 서비스 관리자(service manager)의 기능을 간단히 정리해 보면, Network서비스는 사용자가 웹환경을 이용하여 SBM에 접근하도록 인터페이스를 제공하며, 서비스 번들을 제어 가능하도록 사용자 서비스를 제공한다.

Service Bundle Manager Server는 서비스 번들들의 실행 라이프 사이클인 번들의 설치, 실행, 정지, 갱신 그리고 제거 기능을 관리한다. Network Service를 통해서 사용자들이 접근하여 서비스 번들 관리를 실행하며, 그 외의 서비스 관리자로부터 정보를 수집하여 이용한다. SBM에 서비스 번들이 설치되면 사용자의 별도 제어없이 실행이 되며, Monitoring Manager로부터 각 서비스 번들의 상태 정보를 받아 Service Using History Storage에 저장한다. 일정 서비스 번들(냉장고, 보안 서비스, 위치 인식 카메라, 등)같은 경우는 항상 작동하도록 Monitoring Manager로부터 정보를 받아서 필요한 경우 번들의 자동 재실행 등의 관리기능을 수행한다.

Service Deployment Manager는 Service Bundle Manager Server에 사용자가 제어할 디바이스의 드라이버를 제공하여, 사용자가 요구한 서비스 번들에 적절한 드라이버가 연결되어 동작할 수 있도록 관리한다.

Device Manager는 디바이스별 드라이버를 추가/수정/제거 관리하는 기능을 수행한다. 디바이스별 서비스를 추가/수정/제거 관리하며, 드라이버를 정기적으로 업데이트를 하여 최적의 드라이버가 제공될 수 있게 Driver Storage에 저장하는 기능을 수행한다.

User Manager는 사용자 정보를 추가/수정/제거 관리한다. 사용자의 기본적인 인적사항도 저장하지만, 사용자 별 등급을 이용하여 사용자별로 디바이스 접근권한을 제한하는 기능을 수행한다.

Monitoring Manager는 일정 간격으로 SBM에 연결된 각

서비스 번들의 상황을 체크하고 Service Bundle Manager Server에 해당 정보를 통보한다. 냉장고, 보안 서비스, 위치 인식 카메라와 같은 기기의 서비스 번들이 꺼져 있으면, Service Bundle Manager Server를 통해 필요하면 자동으로 재실행시키도록 한다. 그 외 서비스 번들이 꺼져 있거나 변동이 있을시, SBM 관리자의 모바일 장치에 서비스 번들의 상태 변동을 통보하는 기능을 수행한다.

## 6. SBM을 기반으로 한 서비스 번들

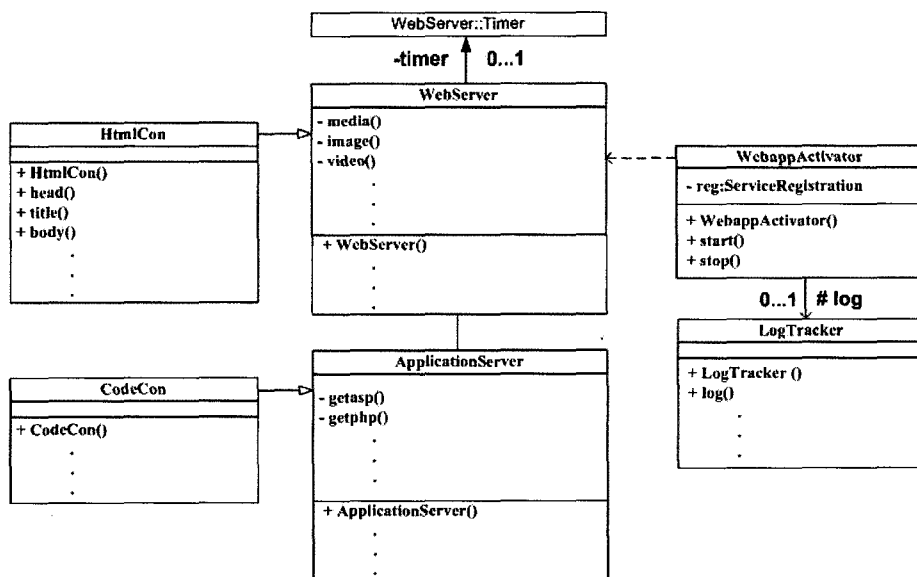
### 6.1 WebApplication 서비스 번들

어플리케이션의 사용자들은 웹 어플리케이션의 기능을 사용하기 위해 웹 브라우저를 사용한다. 사용자들은 웹 브라우저를 이용하여 서비스의 상태정보를 확인하여, 데이터 값들을 채우거나 링크들을 클릭하여 서비스 상태 정보를 보거나 변경할 수 있다.

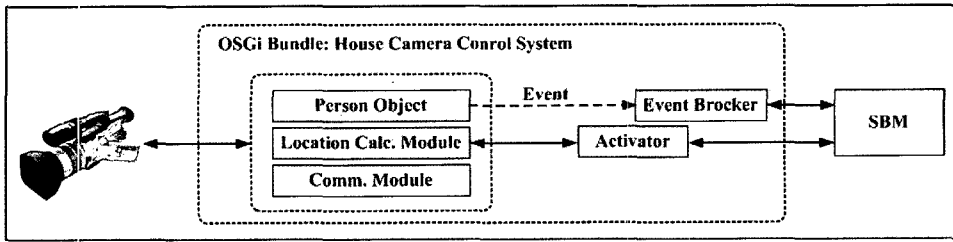
(그림 7)은 WebApplication 서비스 번들의 다양한 컴포넌트(component)들 사이의 데이터 흐름을 보여준다. 웹 어플리케이션(web application)들은 어플리케이션(application)의 기능 전송과 연결된 많은 컴포넌트들을 얻는다. WebServer 클래스는 이미지, 비디오 혹은 오디오 파일들과 같은 멀티 미디어 콘텐츠와 HtmlCon 클래스를 통해 HTML태그(Tag) 페이지(page)를 서비스한다. ApplicationServer 클래스는 HTML 태그들을 WebServer 클래스에 넘겨 서비스하고 xml, asp, php 등의 서비스들을 CodeCon 클래스를 통해 처리한다.

### 6.2 Camera Control 서비스 번들

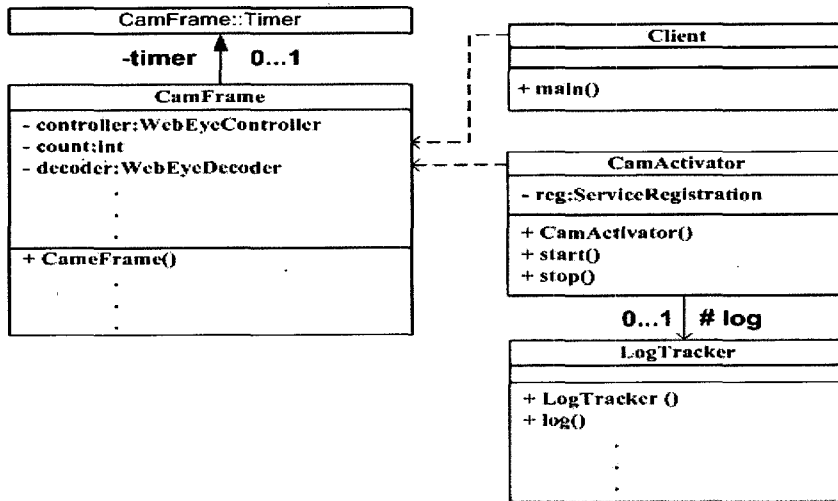
(그림 8)에서 보여지듯이 Activator 컴포넌트와 Event Broker 컴포넌트가 SBM 프레임워크에 접근하며, SBM 프



(그림 7) WebApplication 서비스 번들 계층도



(그림 8) Camera Control 서비스 번들 구조



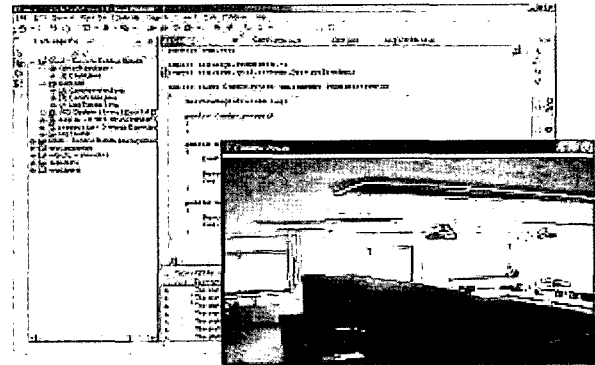
(그림 9) Camera Control 서비스 번들 계층도

레이워크와의 상호작용에 의해서 Activator 컴포넌트는 프레임워크의 관리 인터페이스로부터 Camera Control 서비스 번들의 시작과 정지를 하기 위한 기술을 제공한다. Camera Control 서비스 번들 이벤트(Event)에 관련된 클라이언트는 Event Broker에 기록할 수 있다. 이벤트를 받으면 Event Broker는 모든 기록 클라이언트들에 이벤트 객체(event object)를 보낸다. 이 방법으로 Camera Control 정보는 SBM 프레임워크 내에 있는 멀티 애플리케이션에 의해 동시에 공유된다. 카메라(camera)와 서비스 번들은 통신을 통해서 여러 서비스 제공과 카메라 제어가 가능하게 된다.

Camera Control 서비스는 가정 내에서 일어나고 있는 상황을 실시간으로 웹을 통해서 보여주는 카메라를 움직이거나, 렌즈의 Zoom-in/out 등의 기능들을 제어한다.

(그림 9)는 Camera Control 서비스 번들내의 각 클래스의 관계를 보여주고 있다. CamActivator 클래스에서 번들의 실행과 정지를 맡고 있으며, start()를 호출하면 서비스 번들이 시작되면서 CamFrame 클래스가 수행된다. CamFrame 클래스는 connect() 메소드를 호출해서 카메라에 접속하여 실시간으로 이미지의 정보나 모드를 설정하고 useZoom() 메소드를 호출하여 줌렌즈를 설정하며, 그 외의 메소드들은 팬(pan), 틸트(tilt), 줌(zoom)이 카메라에 장착이 되었는지 확인하는 설정을 하도록 구현되었다.

LogTracker 클래스에서는 주로 이벤트들과 에러 상황을 보고하는데 쓰인다. log() 메소드들은 주어진 로그 레벨로 예



(그림 10) Camera Control 서비스 번들 채널 변경

외와 함께 메시지를 로그로 남기며 그 외에 명시된 서비스들도 제공한다. 이 클래스는 서비스 번들이 시작과 동시에 시작하여 정지되면 log.close() 메소드를 호출하여 정지한다. (그림 10)은 Camera Control 서비스 번들이 실행되는 동안 Client 클래스를 통해서 카메라의 채널을 제어하기 위해 CamFrame 클래스와 데이터를 주고받는 것을 보여주고 있다.

### 7. 결론 및 향후 과제

본 논문은 홈 네트워크상에서 OSGi를 기반으로 하는 여러 서비스 번들들을 효율적으로 관리하기 위한 OSGi기반의

SBM을 제안하고, SBM에서의 서비스 실험 결과를 서술하였다.

수동적으로 서비스 번들/사용자/장치 관리를 하는 OSGi 서비스 플랫폼의 취약점을 해결한 Monitoring Manager를 통해 자동화 서비스를 지원한 SBM은 인증된 사용자에게 각각의 장치 제어를 허용하고, 필요한 경우 사용자의 별도 제어 없이도 각 서비스 번들의 제어가 가능하도록 설계되었다. 사용자가 장치 제어를 하기 위해 웹 브라우저와 모바일 장치를 사용하여 접근하면 SBM은 User Manager와 Device Manager를 통해 사용자 접근 권한을 분석한 장치의 서비스 정보를 서버에 보내어 Service Bundle Manager Server를 통하여 효율적으로 서비스 번들을 제어한다. SBM은 자동적으로 서비스 번들을 업데이트하고 사용자의 권한과 각 장치의 제어를 통해서 서비스 번들을 관리한다. 이러한 관리 기능은 점점 복잡해지는 홈 네트워크 구성에 매우 효과적으로 사용될 수 있을 것으로 기대된다.

향후 가전기기들과 서비스 번들들의 효율적 관리를 위한 연구가 계속 되어야 할 것으로 사료되며, 패턴 정보를 기반으로 좀 더 향상된 상황 인식 기능, 보안 인증 그리고 분산 처리와 같은 SBM의 서비스들이 더 확장된 형태로 기능을 가져야 할 것이며, OSGi와 다른 미들웨어간의 상호운용을 위한 연구도 좀 더 개선되어 적용될 필요가 있다고 보여진다.

## 참 고 문 헌

- [1] H. Schulzrinne, Wu Xiaotao, S. Sidiroglou, S. Berger, "Ubiquitous computing in home networks", *Communications Magazine, IEEE, Vol.40, Issue:11, pp.128-135, 2002.*
- [2] B. Rose, "Home Networks: a standards perspective" *Communications Magazine, IEEE, Vol.39, Issue:12, pp.78-85, 2001.*
- [3] T. Saito, I. Tomoda, Y. Tokabatake, J. Arni, K. Teramoto, "Home gateway architecture and its implementation", *Consumer Electronics, 2003. ICCE 2003 IEEE International Conference on, pp.386-387, 2003.*
- [4] Bea Changseok, Yoo Jinho, Kang Kyuchang, Choe Yoonsik, Lee Jeunwoo, "Home server for home digital service environments," *Consumer Electronics, IEEE Transactions on, Vol.49, Issue:4, pp.1129-1135, 2003.*
- [5] UPnP Specification Version 1.0, <http://www.upnp.org>
- [6] Kim Dong-Sung, Lee Jea-Min, Kwon Wook Hyun, Yuh In Kwan, "Design and implementation of home network systems using UPnP middleware for networked appliances" *Consumer Electronics, IEEE Transactions on, Vol.498, Issue:4, pp.963-972, 2002.*
- [7] Jini Specification Version 1.0, <http://www.jini.org>
- [8] S. Landis, V. Vasudevan, "Reaching out to the cell phone with Jini", *System Sciences, 2002. HICSS Proceedings of the 35<sup>th</sup> Annual Hawaii International Conference on, pp.3821-3830, 2002.*
- [9] HAVi Specification Version 1.1, <http://www.havi.org>.
- [10] R. Lea, S. Gibbs, A. Dara-Abrams, E. Eychison
- [11] IEEE 1394 Specification Version 1.0, <http://www.1394ta.org>
- [12] M. Nakagawa, Zhang Honggang, H. Sato, "Ubiquitous homelinks based on IEEE 1394 and ultra wideband solutions," *Communications Magazine, IEEE, Vol.41, Issue:4, pp.74-82, 2003.*
- [13] H.C. Ferreira, H.M. Grove, O. Hooijen, A.J. Han Vinck, "Power line communications: an overview," *AFRICON, IEEE ARFICON 4<sup>th</sup>, Vol.2, pp.558-563, 1996.*
- [14] OSGi Specification Version 3.0 March 2003, <http://www.osgi.org>.
- [15] R.S. Hall, H. Cervantes, "An OSGi implementation and experience report," *Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE pp.394-399, 5-8 Jan., 2004.*
- [16] IBM Service Manager Framework(SMF), <http://www-306.ibm.com/software/wireless/smf/index.html>
- [17] S. Chemichian, "Building smart services for smart home," *Networked Appliances, 2002. Gaithersburg. Proceedings, 2002 IEEE 4<sup>th</sup> International Workshop on, 2002, pp.215-224.*
- [18] Radio Frequency Identification(RFID), <http://www.aimglobal.org/technologies/rfid>
- [19] R. Want, "Enabling ubiquitous sensing with RFID," *Computer, Vol.37, Issue:4, pp.84-86, 2004.*
- [20] Bluetooth, <http://www.bluetooth.com>
- [21] Lee Kwang Yeol, Choi Jea Weon, "Remote-controlled home automation system via Bluetooth home network," *SICE 2003 Annual Conference, Vol.3, pp.2824-2829, 2003*
- [22] UIML, <http://www.uiml.org>

## 김 남 훈

e-mail : nhkim@dongyang.ac.kr

1988년 연세대학교 전산학과(이학사)

1996년 KAIST 정보및통신공학과(석사)

2006년 세종대학교 컴퓨터공학과(박사)

1988년 1월~1998년2월 삼성SDS 정보

기술연구소 선임연구원



1998년 3월~현재 동양공업전문대학 모바일인터넷과 부교수

관심분야: 무선인터넷, 멀티미디어DB, 홈네트워크 컴퓨팅,

지식기반시스템



**손민우**

e-mail : minwoo15@gce.sejong.ac.kr  
2003년 상지대학교 전자계산공학부  
(공학사)  
2005년 세종대학교 컴퓨터공학부(석사)  
2005년~현재 세종대학교 컴퓨터공학부  
박사과정

관심분야: 홈네트워크, 미들웨어, HCI, Data Mining



**신동일**

e-mail : dshin@sejong.ac.kr  
1988년 연세대학교 전산과학과(이학사)  
1993년 M.S. in Computer Science,  
Washington State University  
1997년 Ph.D in Computer Science,  
University of North Texas

1997년 9월~1998년 2월 시스템공학연구소 선임연구원  
1998년 3월~현재 세종대학교 컴퓨터공학과 조교수  
관심분야: 무선인터넷, HCI, 게임엔진, CSCW



**신동규**

e-mail : shindk@sejong.ac.kr  
1986년 서울대학교 계산통계학과(학사)  
1992년 M.S. in Computer Science,  
Illinois Institute of Technology  
1997년 Ph.D in Computer Science,  
Texas A&M University

1986년 2월~1991년 8월 한국국방연구원, 연구원  
1997년 8월~1998년 2월 현대전자 멀티미디어연구소,  
차장(책임연구원)  
1998년 3월~현재 세종대학교 컴퓨터공학과 부교수  
관심분야: XML 보안, 전자상거래, MPEG