유비쿼터스 홈 네트워크를 위한 JMX 기반 관리 프레임워크의 설계 및 구현

정회원 김 대 영*, 이 종 언**°, 차 시 호***, 종신회원 조 국 현*

Design and Implementation of a Management Framework for Ubiquitous Home Networks

Dae-Young Kim*, Jong-Eon Lee**°, Si-Ho Cha*** *Regular Members*
Kuk-Hyun Cho* *Lifelong Member*

요 약

본 논문에서는 유비쿼터스 홈 네트워크의 제어 및 관리를 위해 JMX기반 관리 프레임워크를 설계하고 구현한다. 이를 위해 정보가전 기기와 센서 기기들로 구성되는 유비쿼터스 홈 네트워크 테스트 베드를 구축하고, 테스트 베드 상에서 관리 프레임 워크를 구현한다. 프레임워크는 홈 네트워크 환경에서 다양한 애플리케이션과 장비를 관리 및 제어할 수 있으며, 새로운 서비스를 쉽게 정의하고 추가 할 수 있다. 또한, 프레임워크는 멀티프로토콜을 기반으로 다양한 관리 인터페이스를 제공할 수 있을 뿐 아니라, 기존의 관리 시스템과의 통합이 용이한 구조를 가진다.

Key Words : Home Network, Ubiquitous Sensor Network, JMX(Java Management eXtension), OSGi(Open Service Gateway initiative), Network Management

ABSTRACT

In this paper we design and implement a management framework base on JMX to control and manage ubiquitous home networks. To do this, we organize the ubiquitous home network test-bed that consists of information electronics and sensor devices. In the test-bed network, we implement the management framework which can control and manage various applications and devices in home network environment. In addition, it can also define and add new services easily. Moreover, it is possible to provide various management interfaces with multi-protocols. The framework is formed to integrate legacy management systems readily.

I. 서 론

유비쿼터스란 언제 어디서나 사용자가 기기나 미디어에 구애 받지 않고 정보를 교환할 수 있는 환경을 말한다. 유비쿼터스 시대를 열어가는 홈 네트워크는 이동통신, 초고속 인터넷 등의 유/무선 인프라를 기반으로 A/V 가전, 생활가전, 정보기기 등이 연결되어 다양한 서비스가 제공되는 미래의 가정환경이다[1]. 홈 네트워크는 통신, 방송, 건설, 가전 및 솔루션 등이 결합되어 연관 산업에 대한 신규 수요

※ 이 논문은 2008년도 광운대학교 연구년에 의해 연구되었음.
* 광운대학교 컴퓨터과학과({dykim, khcho}@cs.kw.ac.kr)
** 삼성탈레스 기술연구소(jong-eon.lee@samsung.com) (° : 교신저자)
*** 세종대학교 정보통신공학과 (sihoc@sejong.ac.kr)
논문번호 : KICS2007-12-541, 접수일자 : 2007년 12월 5일, 최종논문접수일자 : 2008년 5월 22일

창출 효과가 매우 큰 서비스 산업이다. 미국과 유럽 등의 기술 선진국들은 미래 IT 산업을 주도할 아이템으로 홈 네트워크에 주목하고 관련 기술 개발에 적극 나서고 있는 추세이다. 이에 정부도 홈 네트워크를 IT 839 정책의 8대 신규서비스 이면서 9대 신성장 동력에 선정하여 로드맵을 제시하고 있다.

홈 네트워크 장치들은 네트워크를 통한 통신에 의해서 제어되고, 또한 여러 제조사들의 제품이 혼재하여 상호 동작하여야 하므로, 상호 호환성과 상호 운용성을 가지고 동작하는 것이 매우 중요하다. 즉, 홈 네트워크에서 다양한 애플리케이션과 장비를 관리할 수 있는 프레임워크는 필수적인 구성요소이다[2]. 이를 위해 다양한 미들웨어와 개방형 플랫폼 기술들이 제시되고 있다.

JMX(Java Management eXtension) 기술은 홈 네트워크 관리 프레임워크 구현에 필요한 통합 인터페이스를 제공한다. JMX는 홈 네트워크에서 제어를 위한 미들웨어로 사용될 수 있을 뿐 아니라 다양한 프로토콜과 호환될 수 있는 장점을 가지고 있어 기존의 관리 인프라와의 통합 및 확장이 용이한 구조를 가지고 있다. 또한, JMX는 표준 플랫폼인 OSGi(Open Service Gateway initiative) 번들 형태로 홈 게이트웨이에 탑재 되는 방식과 일반적인 서버 데몬 형태로도 구현이 가능한 유연한 구조를 가지고 있다.

본 논문에서는 기존의 정보가전 네트워크와 센서 기기들로 구성되는 유비쿼터스 홈 네트워크 테스트 베드를 구축하고, 테스트 베드 상에서 관리 프레임 워크를 JMX 기반으로 설계하고 구현한다. 구현되는 프레임워크는 홈 네트워크 환경에서 다양한 애플리케이션과 장비를 관리 및 제어할 수 있다. 그리고 홈 네트워크에 새로운 서비스를 쉽게 정의하고 추가 할 수 있다. 또한, 프레임워크는 멀티프로토콜을 기반으로 다양한 관리 인터페이스를 제공할 수 있을 뿐 아니라, 기존의 관리 시스템과의 통합이 용이한 구조를 가진다.

논문의 구성은 Ⅱ장에서 홈 네크워크와 연관된 미들웨어와 플랫폼 그리고 JMX에 대하여 고찰하고, Ⅲ장에서는 JMX 기반의 관리 프레임워크를 설계한다. Ⅳ장에서는 세부적인 구현에 대하여 기술하고 Ⅴ장에서 결론을 맺는다.

Ⅱ. 관련 연구

2.1 유비쿼터스 홈네트워크

홈네트워크를 구성하는 네트워크는 일반적으로 멀티미디어 컨텐츠를 서비스를 위한 AV 네트워크, 여러 대의 PC 및 장비간의 통신을 위한 정보 네트워크 그리고 생활가전을 중심으로 제어서비스를 제공하는 하는 생활가전 네트워크로 구성된다. 여기에 유비쿼터스 센서 네트워크(USN)등이 구성 요소에 첨가될 수 있다.

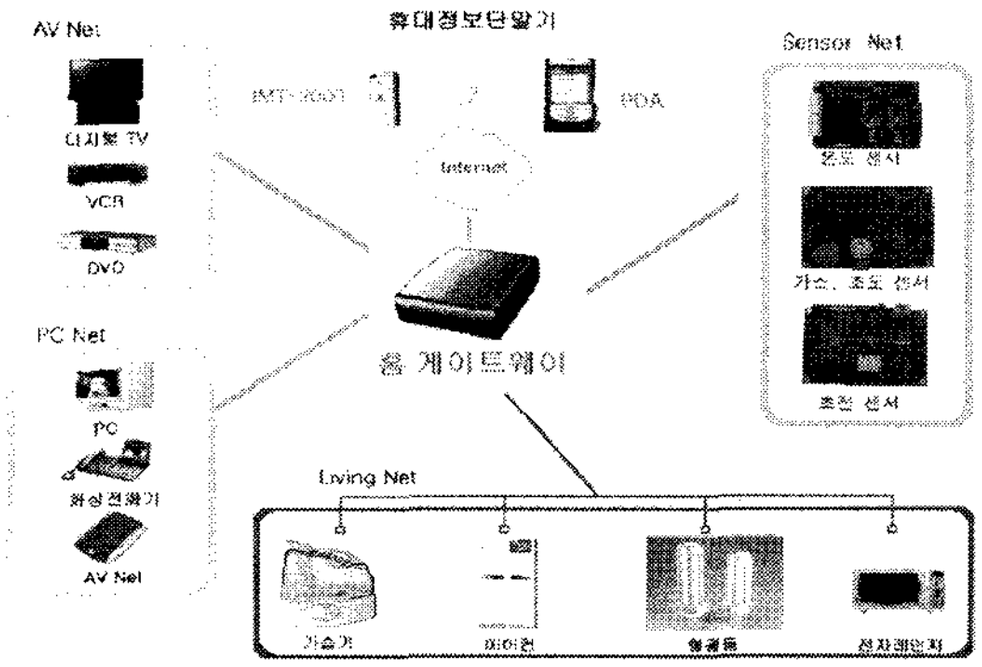


그림 1. 홈 네트워크 구조

이러한 이기종의 네트워크들은 홈 게이트웨이를 통해 하나의 네트워크로 묶이고 사용자에게 다양한 서비스를 제공 할 수 있다. 사용자는 홈 게이트웨이에 접속하여 가전기기를 작동시키거나, 댁내 전력 사용량 검침에 있어서도 PLC(Power Line Communication)를 통해 자동 집계, 통보하는 것이 가능하며, 원하는 시간에 원하는 드라마나 영화를 디지 털 TV를 통해 시청 할 수 있게 된다. 다음의 그림 1은 유비쿼터스 홈네트워크의 일반적인 구조를 보여준다.

2.2 미들웨어와 플랫폼 기술

2.2.1 홈 네트워크 미들웨어

단말간의 상호 발견, 구성/관리를 위한 네트워크 미들웨어 기술은 홈 네트워크 서비스 환경 구축을 위해서 매우 중요하다. UPnP, HAVi, Jini 등 홈 네트워크 제어 미들웨어 등이 있다.

UPnP(Universal Plug and Play)[3]는 MS사의 Plug and Play를 보다 다양한 장비에 적용할 수 있도록 확장한 기술이다. 각 장비들은 언제나 쉽게 네트워크에 접속을 하여 다른 장비들에게 자신의 기능을 알리고 통신을 하거나 제어를 할 수 있도록 해주며, 더 이상 사용하지 않을 경우에는 네트워크에서 쉽게 제거 시킬 수 있도록 해준다. UPnP는 장비간의 일대일 통신을 기반으로 하고 있으며 TCP/IP를 이용하여 그 구조를 정의하고 있다. 그러나, 이로 인해 장비마다 IP를 부여하고 동적으로 IP를 할당 받아야 하기 때문에 DHCP와 같이 가정내의 장비의 동적인 IP 할당방안을 제공하거나 IPv6를 필요로

한다는 점은 UPnP 확산에 부담으로 작용하고 있다. UPnP 모임은 MS, Intel을 주축으로 하여 정보, 가전중심의 250개 이상의 회사들이 회원으로 가입이 되어 있다.

HAVi(Home Audio Video interoperability)[4]는 가정 내의 오디오 및 비디오 가전 기기간의 통신 및 제어 등을 위한 홈 네트워크용 표준으로 정의된다. 즉, 네트워크로 연결된 모든 오디오/비디오 가전 장비들은 네트워크의 연결 순서나 위치, 장비 생산업체와 관계없이 서로 다른 장비의 기능을 제어할 수 있도록 해주는 것이다. HAVi는 특히 오디오와 비디오에 최적화되어 있는 홈 네트워크 프로토콜로서, 가장 큰 특징은 고품질의 디지털 비디오와 오디오 신호를 고속으로 전송할 수 있도록 하기 위하여 고속의 IEEE 1394 네트워크를 기반으로 하고 있다. HAVi는 적용 분야에서 나타나듯이 주로 오디오 및 비디오 장비를 생산하는 8개 주요 회사 (Grunding AG, Hitachi, Matushita(Panasonic), Royal Philips Electrics, Sharp, Sony, Thomson Multimedia, Toshiba) 에 의해서 시작 되어 현재는 국내의 삼성, LG 전자를 포함하여 50여 개 이상의 업체들이 참여하고 있다.

Jini(Java Intelligent Network Infra-structure)[5]는 선 마이크로시스템사에서 제안한 접속기술로써, 자바를 기반으로 네트워크에 접속되어 있는 지능형 기기들이나 소프트웨어들이 동적으로 상호 작용할 수 있는 홈 미들웨어 기술이다. Jini는 분산 네트워크 접속기술을 이용하여 분산 네트워크를 구성하고 장치들 간 정보 공유가 가능해지고 동일 네트워크 안의 장치들은 상이한 하드웨어, 운영체제 하에서도 상호접근이 가능하다. 따라서 각 기기가 지녀야 하는 프로세서, 메모리, 하드 디스크 등 부담을 낮출 수 있다. 하지만, 시스템에 JVM의 도입으로 수행속도 저하와 높은 메모리 점유율로 시스템의 단가가 높아질 수 있다. 또한 Client와 연결되는 Lookup Server의 의존도가 높아 Lookup Server를 제거할 경우 전체 네트워크 마비의 우려가 있다.

2.2.2 OSGi (Open Service Gateway initiative)

OSGi[6]는 홈 네트워크에서 WAN과 LAN의 망들과 장치들 사이에 여러 가지 서비스들을 전달하기 위한 표준 스펙을 정하기 위해 설립되었다. 스펙의 주된 목적은 서비스 기반의 여러 가지 통신방법을 제공하기 위한 플랫폼의 구실을 하는 개방형 서비스 게이트웨이를 정의하는 것이다. OSGi의 주된 특징은 다음과 같다.

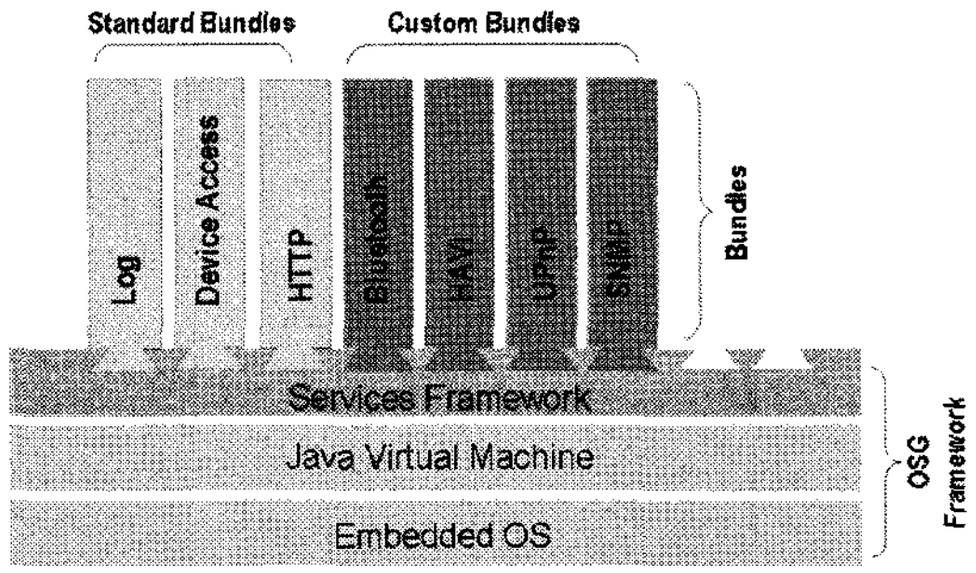


그림 2. OSGi 구조

- 플랫폼과 응용 프로그램의 독립성
- 보안 서비스
- 다양한 로컬 네트워크 기술(블루투스, 무선랜 등)
- 다양한 장치 접근 기술
- 다른 표준들과의 공존 (Upnp, HAVi, Jini, 등)

그림 2는 OSGi의 구조를 나타내고 있는 것으로써, OSGi의 구조는 서비스 프레임워크와 번들, 이 두 가지 주요 컴포넌트들로 이루어져 있다.

- 프레임워크 : 응용 개발 시 일관된 프로그래밍 모델을 제공하는 것이 목적이다. 이 컴포넌트들은 번들이라고 불리고, 요청 시 다운로드 가능하고 제거도 가능하다.
- 서비스 : 정의된 인터페이스를 통해 접근 가능한 컴포넌트로, OSGi 모델에서 응용은 서비스들의 집합으로 만들어진다. 실제 서비스를 하는 부분인 서비스는 프레임워크에 등록된 후에 각각의 역할을 수행한다.
- 번들 : 서비스들과 자바 클래스 그리고 다른 자원들을 포함하는 집합을 가리킨다. jar파일로 되어있고, 프레임워크에서 설치된 후 활성화되면 포함하고 있던 서비스들을 수행한다. 번들은 여섯 가지 상태 중 하나가 된다.

2.2.3 JMX (Java Management eXtention)

JMX[8]는 각종 자원을 MBean(Managed Bean)으로 추상화 하여 MBean 서버에 등록시킴으로써 원격에서 그 자원을 관리하는 것을 가능하게 해주는 기술을 말한다. MBean 서버는 JMX 에이전트에 포함되며 에이전트에는 각종 프로토콜에 대한 인터페이스가 제공되므로 그 프로토콜에 따른 Adapter나 Connector만 구현해 준다면 어떤 프로토콜을 이용하던 접근과 관리가 가능하다는 커다란 장점을 가진다. JMX 구조는 그림 3과 같이 Instrumentation

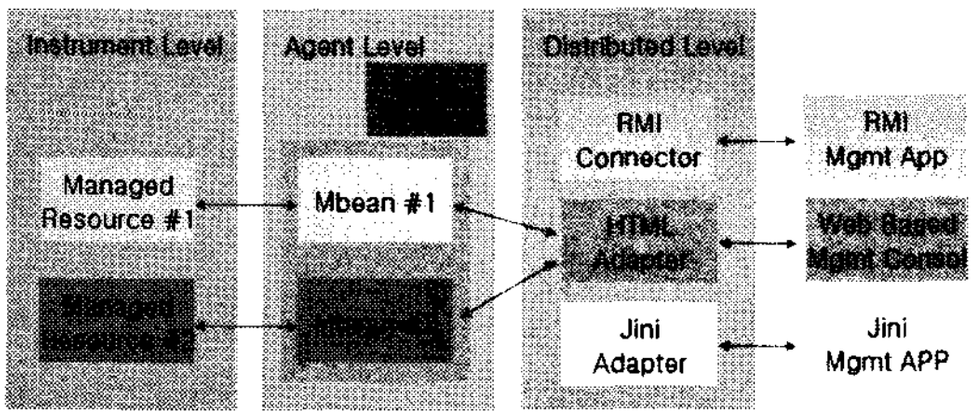


그림 3. JMX 구조

Layer, Agent Layer, Distributed Layer로 나뉜다.

- Instrumentation level : 자원과 직접적으로 상호작용하는 계층으로, 자원을 JMX로 관리 가능하게 구현하기 위한 스펙을 제공한다. 자원들은 하나 이상의 MBean(Managed Bean) 형태로 다루어진다. MBean을 통해 리소스(애플리케이션, 서비스의 구현과 디바이스 등)를 관리할 수 있게 하며 이것들은 자바로 구현되어 있거나 자바 래퍼(Java Wrapper) 클래스가 있어야 한다.
- Agent level : 자원을 관리하고 원격의 관리 툴을 사용가능하게 하고 MBean Server와 MBean들, 시스템에 부가적인 정보를 제공하는 서비스들의 집합으로 구성되며, MBean Server에 서비스들을 등록함으로써 관리 시스템의 요구사항을 동적으로 설정할 수 있게 한다.
- Distributed services level : JMX 매니저를 구현하기 위한 인터페이스를 제공한다. 상위 수준의 프로토콜을 제공하거나 여러 에이전트의 관리 정보를 통합해 사용자의 비즈니스 로직과 관련된 논리적인 뷰를 제공한다. 이와 같이 분산 서비스 레벨은 에이전트와 MBean을 좀 더 쉽게 접근하고 추상화된 뷰를 제공하는 인터페이스를 포함한다.

Ⅲ. 관리 프레임워크 설계

본 장에서는 유비쿼터스 홈 네트워크 관리 프레임워크를 설계하기 위한 고려 사항과 이를 바탕으로 관리 프레임워크를 설계한다. 또한, 프레임워크의 관리 프로세스를 정의하고 PDA 원격 관리자를 설계한다.

3.1 설계 요구 사항

유비쿼터스 홈 네트워크 관리를 위해서는 유연하고 확장성 있는 관리 프레임워크가 제공되어야 한다. 다음은 관리 프레임워크에서 요구되는 일반적인 요구 사항을 나열한 것이다.

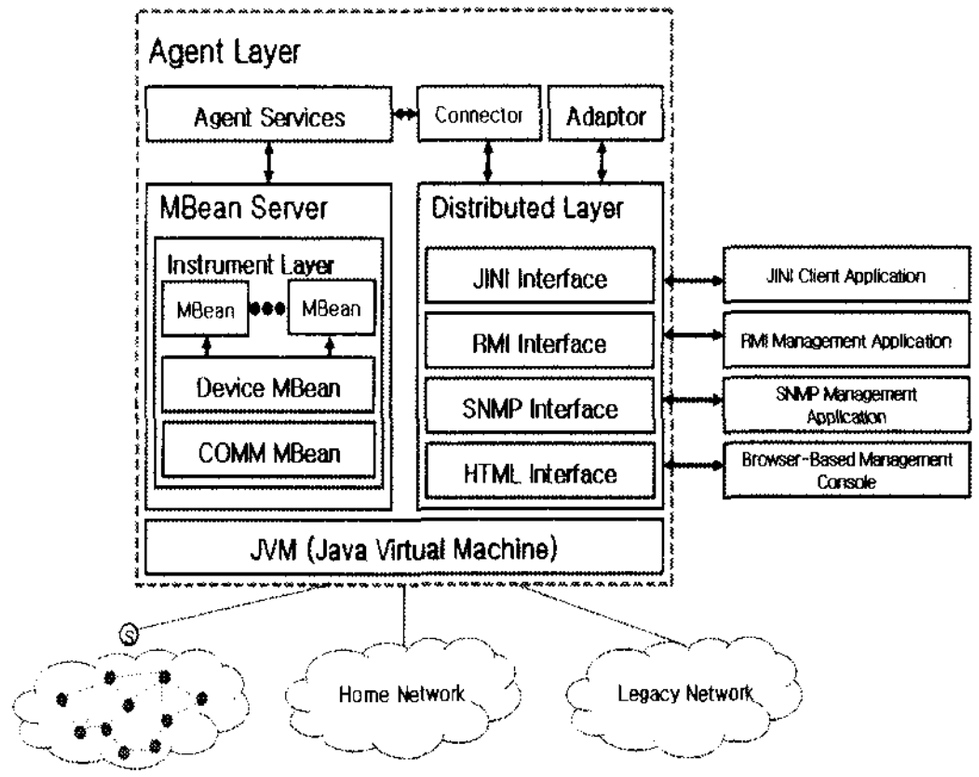


그림 4. 관리 프레임워크 구조

- 관리 프레임워크는 다양한 장치를 지원 할 수 있어야 한다.
- 홈 네트워크를 기반으로 다양한 서비스를 쉽게 정의하고 추가할 수 있어야 한다.
- 기존 네트워크 관리 프레임워크와의 통합이나 확장이 용이하여야 한다.
- 사용자나 관리자에게 다양한 관리 환경을 제공할 수 있어야 한다.

본 논문에서는 위와 같은 고려사항을 해결하기 위해 JMX 기술을 이용한다. JMX 기술은 관리 대상을 MBean으로 추상화 할 수 있고, JMX의 서비스 에이전트를 통해 쉽게 서비스를 정의 할 수 있다. 또한, 멀티프로토콜을 지원하기 때문에 기존 시스템과 통합이 용이하고 사용자 및 관리자에게 다양한 관리 환경을 제공할 수 있다.

3.2 설계

그림 4는 위의 설계 요구 사항에 따라 JMX 프레임워크를 확장한 관리 프레임워크의 구조를 보여준다. 센서 네트워크 연결을 위해 COMM MBean을 설계하고, 네트워크의 디바이스들을 관리하기 위해서 Device MBean을 설계하였다.

또한, Agent Layer에서 다양한 서비스들을 쉽게 추가 및 지원을 할 수 있도록 되어 있으며, Distributed Layer를 통해 기존 관리 시스템간의 연동이 가능하도록 확장성 있는 관리 프레임워크를 설계하였다. 본 논문에서는 설계한 프레임워크를 적용하여 홈 게이트웨이와 센서 및 홈 네트워크 장치들에 대한 정

보를 MBean으로 정의하여 효율적인 관리 환경을 제공하고, 분산 계층에서는 외부 매니저 층에서 다양한 원격 매니저들을 지원할 수 있도록 SNMP, HTML, RMI 등의 다양한 프로토콜을 제공할 수 있다.

3.3 관리 프로세스

그림 5는 관리 자원과 홈 게이트웨이 그리고 관리자 간의 관리 프로세스를 순차 다이어그램으로 보여 주고 있다. 홈 게이트웨이는 관리 할 대상을 찾은 후 그 자원을 관리하는데 사용 할 MBean을 생성한다. 그러고 나서 관리자는 MBean을 통해 원하는 메소드를 호출함으로써 자원을 관리할 수 있게 된다. 이 관리 프로세스는 일반적인 홈 네트워크 디바이스를 관리/제어 하는데 사용된다.

그림 6은 그림 5의 기본 관리 프로세스를 통해서 확장된 다이어그램으로써 기존 SNMP로 관리되던 자원들을 관리하는 시퀀스 다이어그램으로 보여주고 있다. 1번과 2번 과정은 JMX 기반 에이전트에서 생성한 MBean을 MBean서버에 등록시키고 관리자가 MBean 서버에 등록된 객체에 접근하여 실제 가전기기에 접근하는 모습을 보여주고 있으며, 3번 과정은 관리자가 장비로 값을 설정하거나 읽어오는 request 과정이다. JMX 에이전트와 SNMP 에이전트와의 연동을 위하여 MBean의 오퍼레이션에서 SNMP 에이전트에게 3.1.1 과정과 같이 snmpGet, snmpSet 오퍼레이션을 호출함으로써 MBean에서의 정보와 SNMP 에이전트와의 정보를 일치 시키고, 3.1.2 과정과 같이 실제 가전기기에 정보를 보낸다. 본 논문에서는 기존 시스템과의 확장성을 위하여 TCP, SNMP, HTTP, RMI를 위한 어댑터를 구현하여 등록된 MBean에 관리 프로토콜에 상관없이 접근하여 기존 시스템의 변경 없이 확장 가능한 구조로 설계하였다.

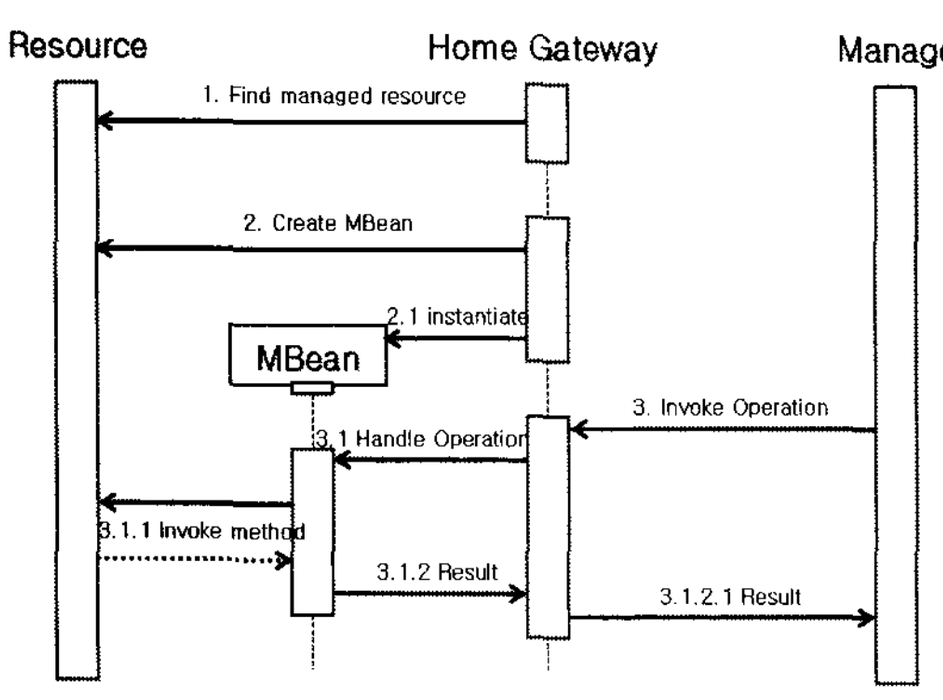


그림 5. 자원 관리 순차 다이어그램

3.4 PDA 원격 관리자

JMX 기반 관리 프레임워크에서 원격 관리자는 다양한 형태로 설계가 가능하지만 본 논문에서는 웹 기반과 PDA 기반의 관리자 형태로 설계하였다. PDA 관리자는 그림 7과 같이 설계하였으며, 클래스 구조는 MVC 모델을 채택하여 설계하였다.

MVC 모델은 각 기능별로 클래스들이 생성되어 있기 때문에, 이벤트 발생에 따르는 구현의 중복성이 줄어들고, 기능의 분할을 통해 클래스들의 상호 의존성을 약하게 만들 수 있는 장점이 있다. 또한 Event driven 방식이 아닌 이벤트를 상위 객체로 전달하는 Event Passing 방식이기 때문에, 윈도우들만의 디자인 구현, PDA에 들어갈 기능별 구현, 서버와 접속하며 받아들이는 정보 저장 및 내부 명령어 구현을 분담하여 할 수 있게 되므로 상세 구현의 개별화가 가능하게 되며, 각 내부 클래스들의 정보 보안이 가능하게 된다.

Ⅳ. 구현 및 테스트

4.1 구현 환경 및 테스트 베드

홈 네트워크 관리 시스템의 구현 환경은 다음과 같다.

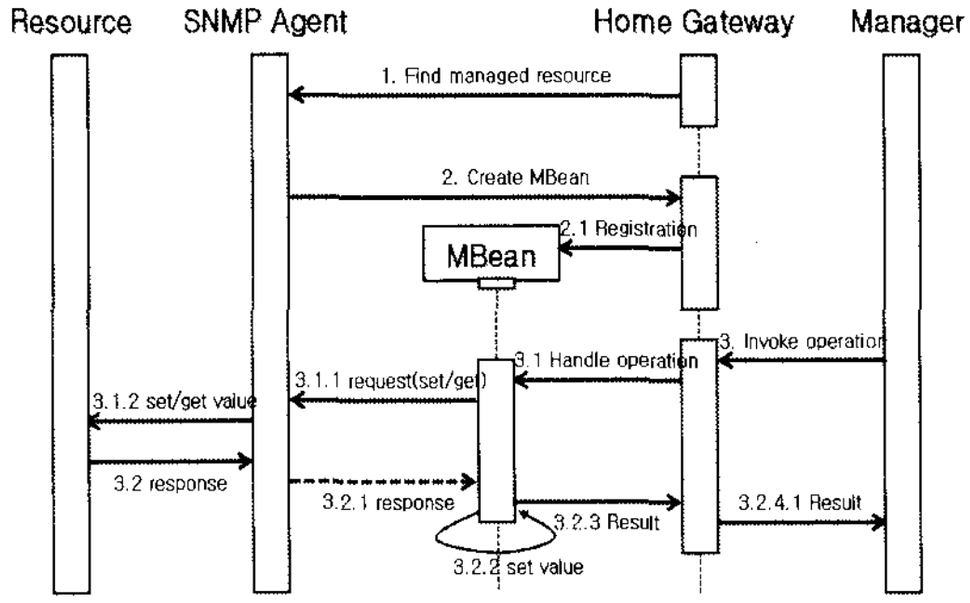


그림 6. SNMP 관리를 위한 확장된 순차 다이어그램

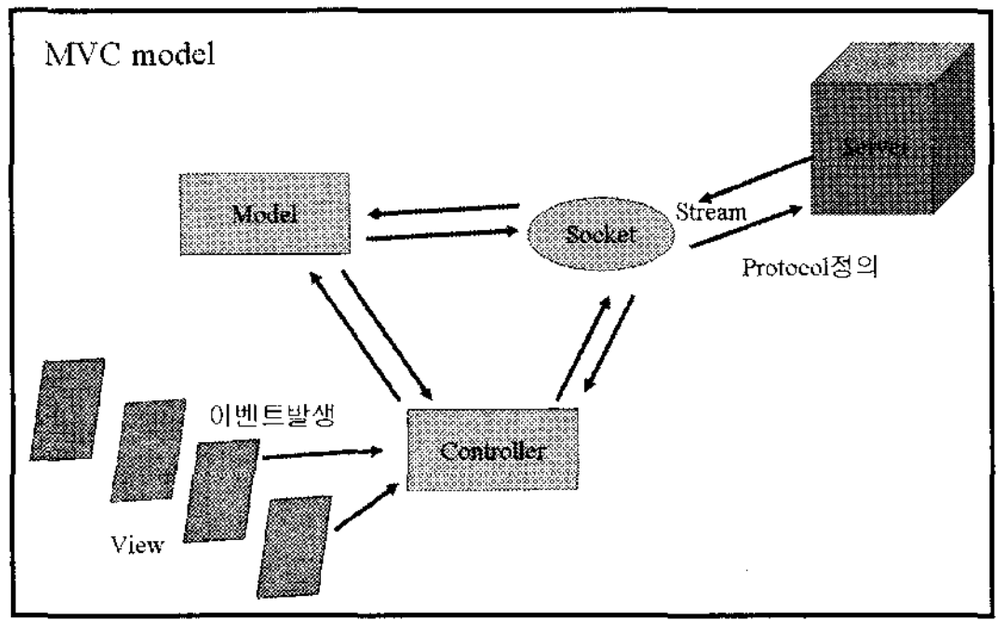


그림 7. PDA Manager Software 설계

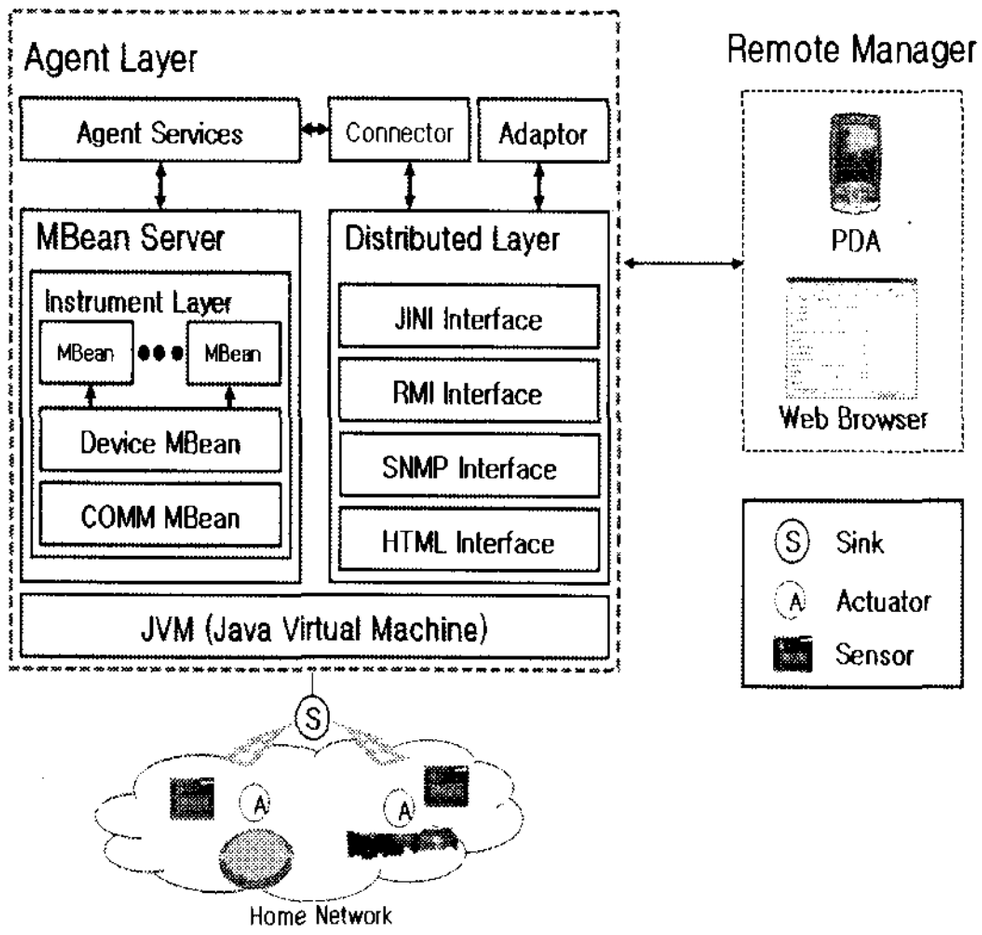


그림 8. 유비쿼터스 홈 네트워크 테스트 베드

- Home Gateway : Window XP, 펜티엄 M4 1.6GHz, J2SE 1.5(JMX)
- PDA : Window Mobile 2003 SE, 400MHz 삼성 S3C 2440, 128MB Flash ROM, 64MB RAM, Micorsoft Visual Studio.Net 2003, ActiveSync 3.7.1, Microsoft Pocket PC 2003 SDK
- Sensor Network : Nano-Qplus[11], Nano-24 kit[12], C(Nano-Qplus API)

그림 8과 같이 Remote Manager는 PAD와 웹 브라우저를 통해서 원격으로 홈 네트워크를 관리하며, 홈 네트워크에는 엑츄레이터와 센서들로 구성된다. 센서는 가스센서와 조명센서가 있으며, 센싱되는 값들은 싱크노드를 통해서 홈 게이트웨이로 전달된다. 이 때, 홈 게이트웨이에서 일정 임계치 값이 넘어가는 값이 오면 원격 관리자로 통보도 가능하며, 원격 관리자가 값을 설정, 변경, 모니터링이 되어 전체 네트워크를 관리하는 구조로 구성하였다.

4.2 구현

그림 9는 본 논문에서 제시한 시스템의 센서 데이터 프로세스 과정이다. 센서에서 수집된 환경정보는 RF 통신을 이용하여 싱크노드로 전달되고, 그 데이터는 COM 포트를 이용한 통신을 이용 하여 게이트웨이의 COMM MBean으로 전달된다. 그 다음 센싱 데이터 정보는 JMX의 Notification 이라는 기능을 이용하여 센서 정보를 관리하는 관리MBean으로 전달되어 관리된다. 이후 외부 매니저가 이 인터페이스에 접속하여 MBean센싱 정보를 얻어간다. 이와 같은 방식으로 센서의 센싱 데이터가 전달되

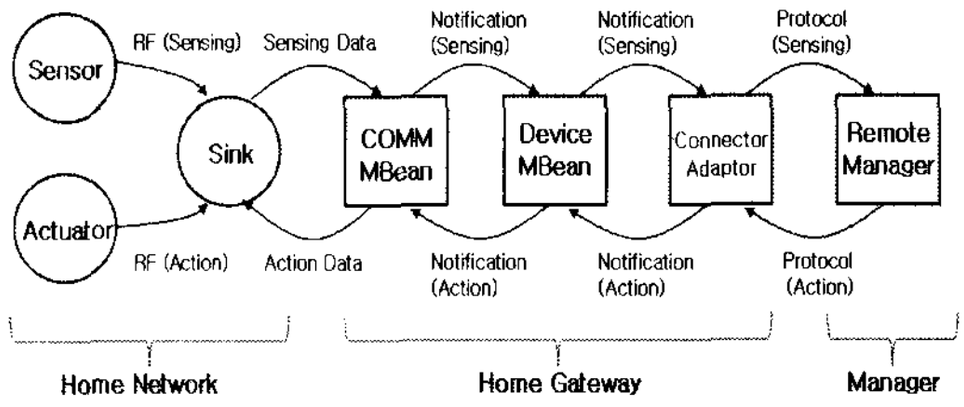


그림 9. 데이터 프로세스

고, 역방향으로는 센서 네트워크로 명령이 수행된다.

4.2.1 PDA 관리자

PDA 관리자는 이벤트 패싱 방식으로 구현 하였다. 이벤트 패싱 방식의 구현은 각 이벤트를 리스너로 감싸기 때문에 발생한 이벤트가 어떻게 처리되는지 모르게 되어, 보안성 향상 및 코드의 간결화가 이루어지게 된다. 다음의 표 1의 코드에서는 윈도우에서 버튼 클릭 이벤트 발생시 Action Factory의 인스턴스가 하나만 생성되어 사용자가 임의로 조작하지 못하도록 한다. 그리고 객체화된 Action들을 switch문으로 판별하여 생성하게 되는데, 이는 추후 생성될 다른 홈 네트워크의 기능을 쉽게 확장할 수 있도록 고려된 것이다.

4.2.2 DeviceMBean

네트워크들의 다양한 디바이스들을 관리하기 위해서 많은 고려사항 및 프로그램의 복잡성이 야기되었다. 하지만, DeviceMBean을 정의함으로써 다양한 디바이스들을 쉽게 관리할 수 있다. 표 2에서

표 1. PDA 관리자 코드

```
private void btnOption_Click(object sender,
System.EventArgs e) {
    OptionAction action = new OptionAction();
    this.lsnr.actionPerformed(action);  }
...
public static  ActionFactory instance() {
    if ( factory == null)
        factory = new  ActionFactory();
    return factory;
}
public Action createAction( int type ) {
    Action action = null;
    switch(type) {
        case LOGIN_ACTION :
            action = new LoginAction();
            break;
        case Exit_ACTION :
            action = new ExitAction();
            break;
    }
}
```

표 2. DeviceBean 코드

```
import device.DeviceMBean;
public interface LightMBean extends DeviceMBean {
    //operation
    public void ManualMode();
    public void AutoMode();
    public void TimeMode();
    // 1. scheduling mode (0:수동,1:자동,2:시간);
    public String getMode();
    //2. auto 모드용 Threshold
    public void setMaxThreshold(int Max);
    public void setMinThreshold(int Min);
    //3. time 모드용 시간  //켜짐
    public void setOnTime(String ontime); //꺼짐
    public void setOffTime(String offtime);
    //3. scheduling 작동 상황
    public boolean getSchedOperation();
}
```

표 3. LightBean 코드

```
import device.DeviceMBean;
public interface LightMBean extends
DeviceMBean{
    //operation
    public void ManualMode();
    public void AutoMode();
    public void TimeMode();
    // 1. scheduling mode (0:수동,1:자동,2:시간);
    public String getMode();
    //2. auto 모드용 Threshold
    public void setMaxThreshold(int Max);
    public void setMinThreshold(int Min);
    //3. time 모드용 시간 //켜짐
    public void setOnTime(String ontime); //꺼짐
    public void setOffTime(String offtime);
    //3. scheduling 작동 상황
    public boolean getSchedOperation();
}
```

DeviceMBean의 코드는 홈 네트워크를 위하여 센서 및 액츄레이터들을 관리하기 위한 코드의 일부분을 보여주고 있다. 이 DeviceMBean은 장치 관리에 필요한 최소한의 정보를 가지고 인터페이스 형식으로 제공함으로써, 관리되는 디바이스들은 단순히 이 인터페이스를 확장하여 사용하면 된다.

표 3은 표 2의 DeviceMBean 인터페이스를 통하여 홈 네트워크에서 관리되어지는 전등 디바이스를 관리하기 위한 LightMBean 구현 코드로써, 관리에 필요한 값들만 단순히 설정하면 되는 하나의 예를 보여주고 있다. 경고성(가스, 적외선 센서) 이든 관리성(조도, 온도, 습도 등)이든 상관없이 센서 장치들을 새롭게 추가하기 위해서는 이와 같이 DeviceMBean만 상속받아 구현함으로써 유연한 구조를 가질 수 있도록 MBean을 구성하였다.

표 4. Sensor Node 코드

```
void rf_recv_data(ADDRESS *srcAddr, INT8
nbyte, BYTE *data)
{
    BYTE protocolId = 0, ommand = 0;
    BYTE actuator = 0, operation = 0;
    LED1_BLINKING();
    unsigned char buff[7];
    protocolId = data[0]; command = data[2];
    if(protocolId == 0x55) puts("Protocol : Star
      Network Application\n");
    else puts("Protocol : Unknown\n");
    if(command == ACTION_COMMAND)
    puts("Command : Action\n");
    else puts("Command : Unknown\n");
    switch(command) {
      case ACTION_COMMAND:
        LED2_BLINKING();
          actuator = data[3];
        operation = data[4];
        if(actuator == LIGHT_SENSOR)
            puts("Actuator : LIGHT\n");
        else puts("Actuator : Unknown\n");
        if(operation == STATE_ON)
            puts("Operation : Turn Off\n");
        else {
            itoa(operation, buff, 16);
          puts("Operation : Unknown [");
          puts(buff); puts("]\n\n");}
        if(operation == STATE_ON) {
            ACTUATOR0_ON();
            ACTUATOR1_ON();
        //currentState = STATE_ON;
        //transmit_act_state(); }
        else if(operation == STATE_OFF) {
            ACTUATOR0_OFF();
            ACTUATOR1_OFF();
        } break;
      default:   break;
    }
}
```

4.2.3 센서 네트워크

원격 관리자에 의해 전달된 명령은 구현된 홈 게이트웨이의 Mean을 거쳐 센서 네트워크로 전달된다. 다음의 표 4는 전달된 명령에 따라 구동기를 작동하여 조명의 전원을 ON/OFF 시키는 소스의 일부이다.

4.3 구현의 확장

JMX는 홈 네트워크 표준 플랫폼인 OSGi 번들 형태의 구현이 가능하다. 즉 JMX 단독으로 서버 프로그램 형태로도 구현이 가능하고, OSGi 플랫폼에 임베디드 되는 형태로도 가능하다. 그림 10과 같이 OSGi 프레임워크에서 본 논문의 관리 프레임워크를 JMX 번들 형태로 구현할 수 있다. 이렇게 구현하게 되면 UPnP나 HAVi와 같은 이기종 미들웨어 기반의 디바이스나 네트워크를 동시에 관리할 수 있는 통합 관리 환경을 쉽게 구현할 수 있다.

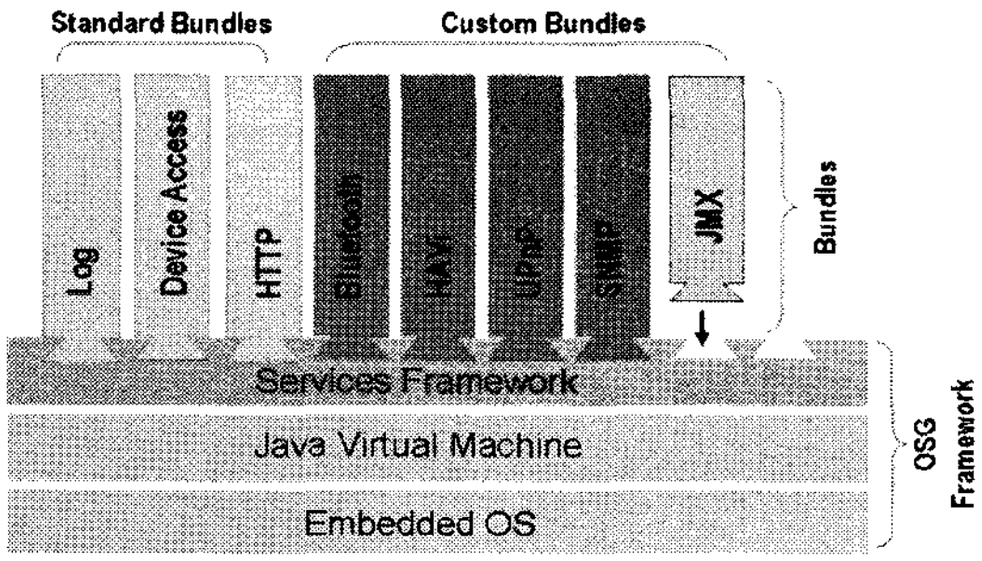


그림 10. OSGi 상의 JMX 번들

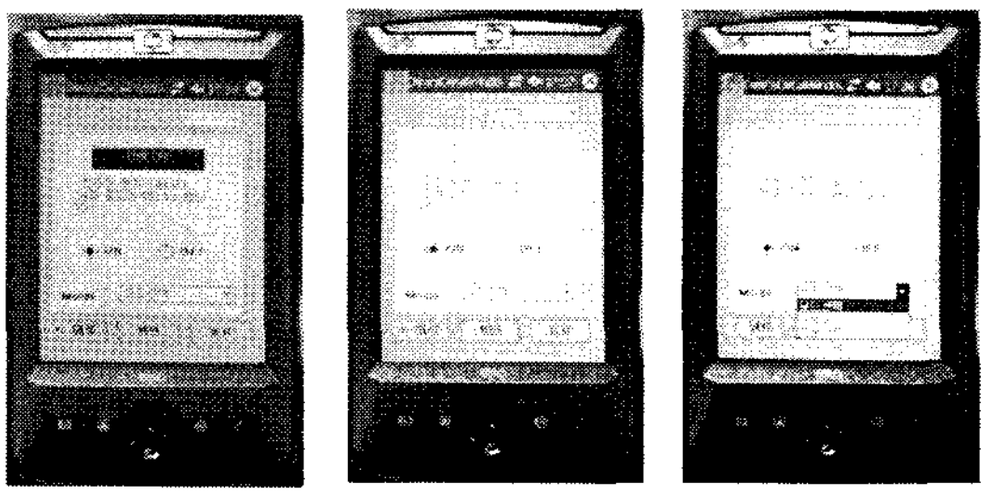


그림 11. PDA 관리자의 전등 제어 화면

4.4. 테스트

그림 11은 PDA를 사용하여 전등을 On/Off 시키거나 Mode(자동/수동) 설정을 수행하는 화면이다. 첫 번째 화면은 전등 정보가 Off이며 조도 값을 모니터링하고 있다. 두 번째 화면은 모드 정보가 자동 모드로 설정되어 있어서 어두워지면 저절로 전등이 On되는 화면을 보여주고 있다. 마지막 화면은 자동/수동모드를 선택할 수 있는 화면이다.

그림 12는 그림 11에서 보인 것과 같이 원격 PDA 관리자로부터 전달된 명령이 홈 게이트웨이를 통하여 패킷 변환 후 구동기로 명령이 전달된 결과를 보인 것이다. 즉, 조도 센서가 현재의 조도 값을 읽어 보내고 그 값이 적으면(어두우면) 조명 구동기로 패킷을 전송함으로써 전등을 On 시킬 수 있다.

그림 13은 웹 브라우저를 통해서 홈센서 네트워크를 관리하는 관리자 화면이다. 각 장치의 현재 상태를 확인할 수도 있으며, 현재 actuator의 상태, 모드 상태, 위치 등의 상세한 내용을 알 수 있다. 홈 게이트웨이에서 설계된 MBean에 따라 센서 정보와 센서의 ON/OFF 등의 조작 기능이 제공되며, PDA 원격 관리자와 똑같은 기능을 제공한다. 따라서 PDA에서 설정을 변경하면, 웹브라우저의 설정 값 또한 자동으로 바뀌게 된다. 뿐만 아니라 설계된 MBean을 통하여 여러 가지 속성들을 사용자가 요구에 따라 추가 및 설정할 수 있다.

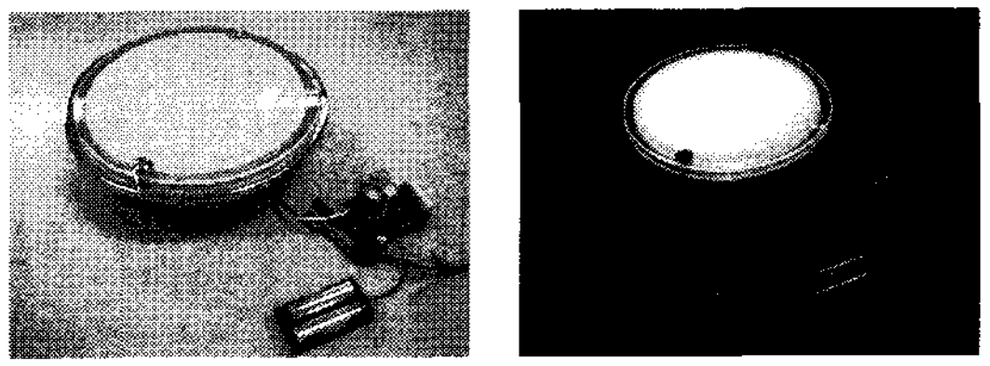


그림 12. 전등 제어 결과

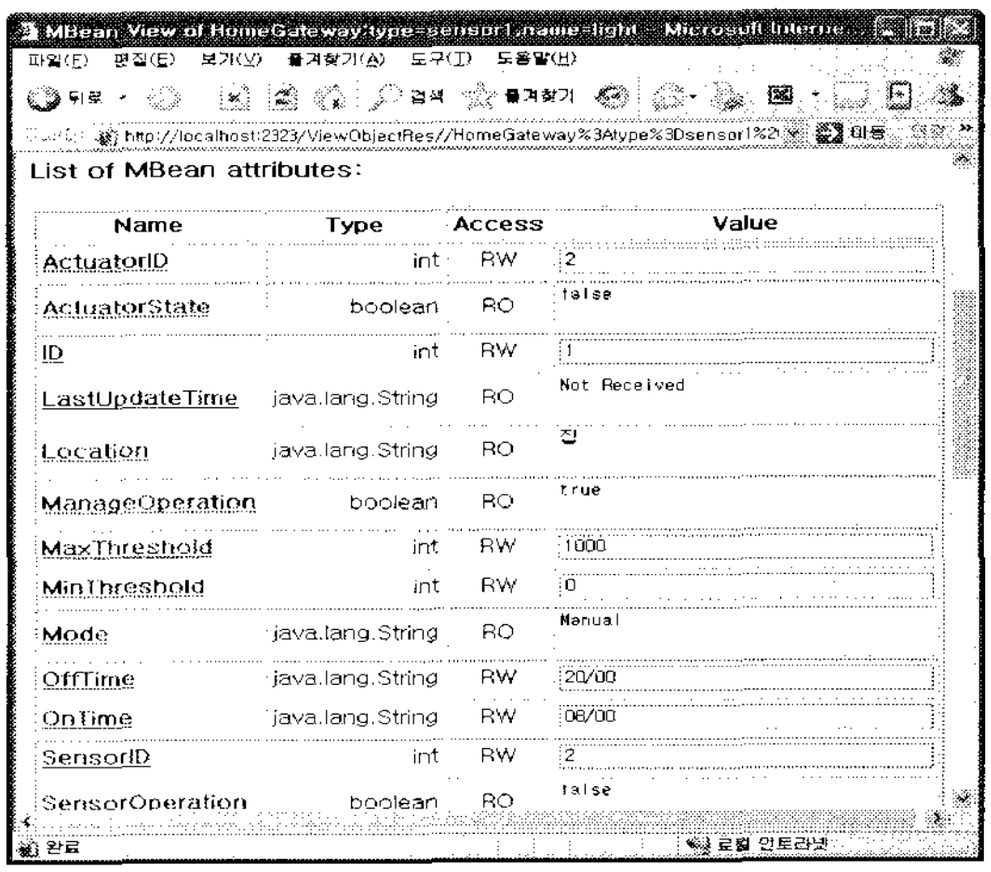


그림 13. 웹브라우저 기반 센서 네트워크 관리 화면

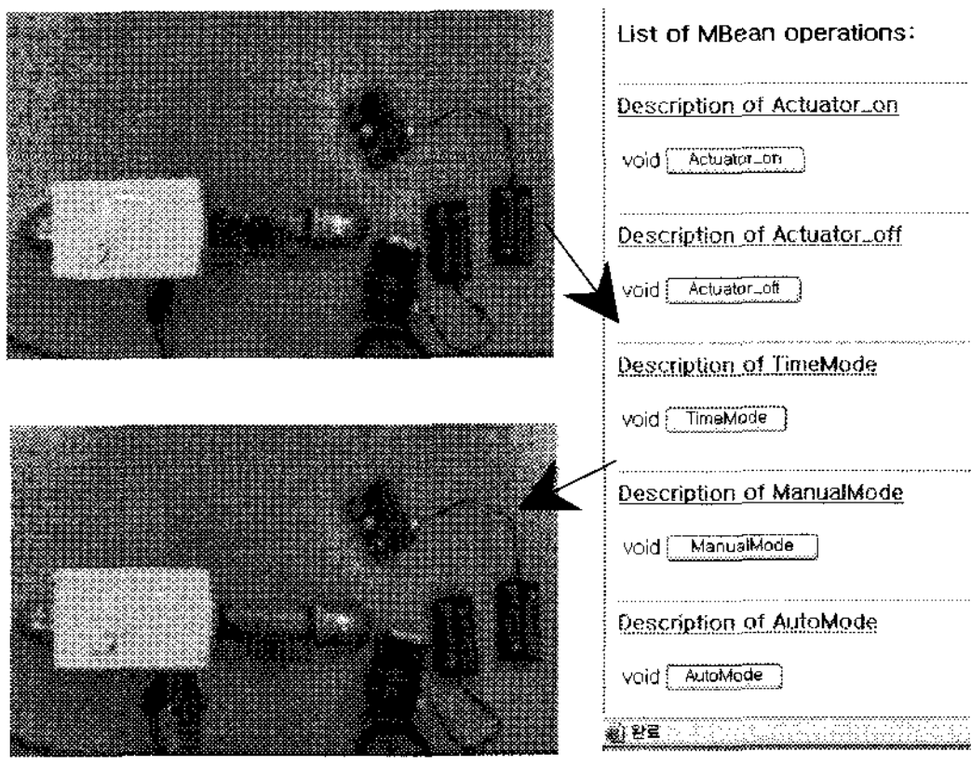


그림 14. 가스밸브 동작 화면

그림 14는 그림 13의 웹브라우저 관리 화면을 통해서 센서 정보를 확인하고 가스밸브 구동기를 동작시킨 화면이다. 홈 게이트웨이에 접속된 웹브라우저를 통해서 열려있는 가스밸브의 상태정보를 확인하고 미리 정의된 MBean 메소드(가스밸브 닫기)를 통해서 가스밸브를 닫는 동작을 수행한다.

4.5 결과 분석

국내외적으로 홈게이트웨이, 홈서버, 미들웨어 등

을 통해 홈 네트워크를 제어하기 위한 연구가 활발히 진행되고 있다. 하지만, 본 논문에서 제시한 홈 네트워크 제어 및 관리를 위한 관리 프레임워크는 홈 네트워크 환경에서 다양한 애플리케이션과 장비를 관리 및 제어할 수 있으며, 새로운 서비스들을 쉽게 정의하고 추가할 수 있다는 점에서 기존 연구와의 차이가 있다. 또한, JMX 기술을 활용함으로써 멀티프로토콜을 기반으로 하여 다양한 관리 인터페이스를 제공할 수 있을 뿐 아니라, 기존의 관리 시스템과의 통합이 용이한 구조를 제시하였다.

본 논문에서 제시한 관리 프레임워크를 실험하기 위해서 유비쿼터스 홈을 위한 가장 기본적인 관리 항목인 전등과 가스밸브를 사용하여 표 5와 같은 항목을 실험하였다. 본 관리 프레임워크는 이 외에 다른 관리 항목도 쉽게 추가할 수 있다.

표 5. 실험 항목

실험 항목
① PDA → 관리프레임워크 → 전등
② PDA → 관리프레임워크 → 가스밸브
③ 가스밸브 → 관리프레임워크 → PDA
④ 웹브라우저 → 관리프레임워크 → 전등
⑤ 웹브라우저 → 관리프레임워크 → 가스밸브

V. 결론 및 향후과제

본 논문에서는 유비쿼터스 홈 네트워크를 위한 JMX기반 관리 프레임워크를 설계하고 구현하였다. 구현을 위해 정보가전과 센서 기기로 구성되는 테스트 베드를 구축하고 구축된 테스트베드 상에 관리 프레임워크를 구현하였다.

관리 프레임워크는 JMX 기반으로 구현되었기 때문에 센서 네트워크를 포함하는 다양한 디바이스의 특성에 적합하게 관리 속성을 정의하고 관리 할 수 있다. 그리고 네트워크 계층과 응용 계층을 서로 분리함으로써 변화하는 네트워크에 상관없이 서비스를 독립적으로 개발하고 이용할 수 있다. 또한, 게이트웨이를 통해 관리자가 SNMP, RMI, TCP등 다양한 형태로 관리가 가능한 확장성 있는 관리 구조를 가진다.

향후 과제로는 테스트베드의 장비와 네트워크를 확장하여 더 많은 장치와 프로토콜을 지원하도록 확장하는 것과 현재 고려되지 않은 보안 관련된 컴포넌트를 추가 구현하는 것이다.

참 고 문 헌

〔1〕 박세현, "차세대 홈네트워크 서비스의 발전 방향 및 요소 기술," ITFIND 주간기술동향, 2005.

〔2〕 강동오, 강규창, 이전우, "홈네트워크 장치 제어를 위한 OSGi 및 UPnP 미들웨어 기술," 한국조명/전기설비학회, 조명/전기설비 제18권 제2호, 2004.

〔3〕 UPnP Forum, http://www.upnp.org.

〔4〕 HAVi, http://www.havi.org.

〔5〕 JINI Network Technology, http://java.sun.com developer/products/jini/index.jsp.

〔6〕 OSGi, "OSGi Service Platform Release 1.0," 2003.

〔7〕 Richard S. Hall, Humberto Cervantes, "An OSGi Implementation and Experience Report," Consumer Communications and Networking Conference, 200.

〔8〕 Java Management eXtention(JMX), http://kr.un.com.

〔9〕 Julio Guijarro, "Framework for managing large scale component-based distributed applications using JMX," 2002.

〔10〕 Benjamin G. Sullins, Mark B. Whipple, "JMX in Action," MANNING, 2003.

〔11〕 Qplus-N, ETRI, http://www.etri.re.kr.

〔12〕 Nano-24: Sensor Network, Octacomm Inc., http://www.octacomm.net.

〔13〕 Richard S. Hall, Humberto Cervantes, "An OSGi Implementation and Experience Report," Consumer Communications and Networking Conference, 2004.

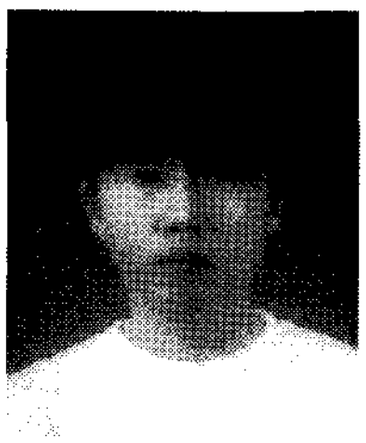
〔14〕 Julio Guijarro, HP Labs. "Framework for managing large scale component based distributed application using JMX," May 2002.

〔15〕 Saito, T.; Tomoda, I.; Takabatake, Y.; Arni, J.; Teramoto, K. "Home gateway architecture and its implementation" IEEE Transactions on Volume 46, Issue 4, Nov. 2000.

〔16〕 B. Tierney, B. Crowley, et al. A Monitoring Sensor Management System for Grid Environments. In High Performance Distributed Computing (HPDC-9, Pittsburgh (Pennsylvania), August 2000.

〔17〕 Abdelkader Lahmadi, Laurent Andrey, Olivier Festor: On the Impact of Management on the Performance of a Managed System: A JMX-Based Management Case Study. DSOM 2005: 24-35

김 대 영 (Dae-Young Kim) 정회원



2002년 남서울대학교 컴퓨터학과 졸업(이학사)
2004년 광운대학교 대학원 컴퓨터과학과 졸업(공학석사)
2004년~현재 광운대학교 대학원 컴퓨터과학과 박사과정
<관심분야> 네트워크 관리, 무선 네트워크, 홈 네트워크, 유비쿼터스 센서 네트워크

이 종 언 (Jong-Eon Lee) 정회원

2001년 광운대학교 전자계 산학과 졸업(이학사)
2003년 광운대학교 대학원 컴퓨터과학과 졸업(공학석사)
2007년 광운대학교 대학원 컴퓨터과학과 졸업(공학박사)
2007년~2008년 세종대학교 유비쿼터스정보통신사업단 전임연구원
2008년~현재 삼성탈레스 기술연구소 책임연구원
<관심분야> 네트워크 관리, 유비쿼터스 센서 네트워크, 무선 메시 네트워크

차 시 호 (Si-Ho Cha) 정회원

1995년 순천대학교 전자계산학과 졸업(이학사)
1997년 광운대학교 대학원 전자계산학과 졸업(이학석사)
1997년~2000년 대우통신 종합연구소 선임연구원
2000년~2003년 광운대학교 대학원 컴퓨터과학과(공학박사)
2004년~2005년 WarePlus, Inc. 책임연구원
2005년~2006년 세종대학교 컴퓨터공학과 초빙교수
2007년~현재 세종대학교 정보통신공학과 연구교수
<관심분야> 네트워크 관리, 무선 센서 네트워크, 유비쿼터스 컴퓨팅, MIMO/Congnitive Networks

조 국 현 (Kuk-Hyun Cho) 종신회원

1977년 한양대학교 전자공학과 졸업(공학사)
1981년 일본 동북대학교 대학원 졸업(공학석사)
1984년 일본 동북대학교 대학원 졸업(공학박사)
1984년~현재 광운대학교 전자정보대학 컴퓨터공학부 교수
개방형컴퓨터통신연구회(OSIA) 회장 역임
<관심분야> 네트워크 관리, 분산처리, 정보통신 분야 표준화 등