

# UPnP 미들웨어기반의 홈 게이트웨이와 지능형 가전기기 간 연동 시스템 설계 및 구현

한왕원<sup>0</sup> 김혜영, 이동주, 김영만  
국민대학교 컴퓨터학부  
(wwhan<sup>0</sup>, totoro, pocsin, ymkim)@kookmin.ac.kr

## Design and Implementation of Convergence System between Home Gateway and Intelligent Home Appliance Using UPnP Middleware

Wang Won Han<sup>0\*</sup> Hye-young Kim, Dong-ju Lee, Young Man Kim  
Communication Lab, School of Computer Science, Kookmin University

### 요 약

최근 정보통신 기술의 발전으로 인하여 어느 장소에서든 컴퓨터 또는 무선장치를 사용하여 집안의 지능형 가전기기들을 점검하고 제어 할 수 있는 수준까지 이르렀다. 이와 같은 서비스를 제공하기 위한 요소기술 중에 마이크로소프트사에서 발표한 UPnP(Universal Plug & Play)는 TCP/IP 프로토콜을 기반으로하여 각 지능형 가전기기에 IP주소를 할당하여 어느 곳에서나 누구나 편리하게 지능형 가전기기의 서비스를 이용할 수 있도록 해준다. 본 논문에서는 UPnP 미들웨어 기반의 홈 네트워크에서 웹 기반의 인터페이스를 통해 지능형 가전기기를 쉽게 관리할 수 있는 시스템을 설계하고 구현한다.

### 1. 서론

우리나라의 컴퓨터 보급대수가 늘어나고 초고속 인터넷이 대부분의 가정에 보급됨에 따라 가정내 정보화도 가속화되고 있다. 또한 한대 이상의 컴퓨터를 보유한 가구가 증가하고, 스캐너, 프린터 등의 컴퓨터관련 장비뿐만 아니라 인터넷이 가능한 냉장고와 같은 네트워크에 접속이 가능한 가전기기들이 출현함에 따라 홈 네트워크를 구축할 수 있는 기반이 빠르게 조성되고 있다. 이에 따라 홈 오토메이션에 대한 욕구가 증가하고, 홈 네트워크 기술에 대한 관심이 증가되고 있으며, 홈

네트워크 기술 분야의 표준 경쟁 또한 치열해지고 있다.

지능형 가전기기들을 서로 네트워크로 연결하기 위해서는 지능형 가전기기들간의 연결기술(Ethernet, Bluetooth, 무선 랜, IEEE 1394, PLC 기술 등)이 필요하다. 이와 함께 홈 네트워크상의 장치들에 대한 상세한 제어과 지능형 가전기기간의 상호작용을 원활히 하기 위해서는 홈 네트워크에 연결된 지능형 가전기기들을 통합하여 관리할 수 있는 미들웨어가 필요하다. 이러한 미들웨어는 홈 네트워크에 연결된 지능형 가전기기들을

발견하고 관리하며, 사용자가 이들을 제어 할 수 있도록 인터페이스를 제공하는 역할을 수행 한다. 현재 이러한 미들웨어 기술로는 UPnP[1][2], Jini[3], HAVI[4] 등이 있으며, 이중 UPnP는 기존 컴퓨터상에서 주변기기들을 관리하기 위해 사용되던 기술인 Plug and Play 기능을 홈 네트워크에 확대 적용한 기술로서, 컴퓨터나 홈 게이트웨이와 같은 컨트롤 포인트의 역할을 수행할 수 있는 장치를 중심으로 가정 내 각종 지능형 가전기기를 제어한다.

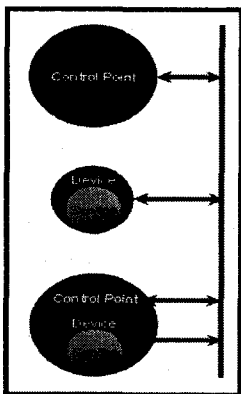
본 논문에서는 UPnP 기술의 기본개념과 동작원리에 대해 우선 살펴보고, 인텔사에서 UPnP 명세서에 맞춰서 개발한 "Intel UPnP SDK for Linux"[5]를 소개한다. 마지막으로 인텔사의 UPnP SDK를 사용하여 웹 기반의 인터페이스를 통해 지능형 가전기기를 쉽게 관리할 수 있는 홈 네트워크 시스템을 설계하고 구현한다.

## 2. UPnP 구조 및 프로토콜 소개

이번 절에서는 본 논문에서 설계하고, 구현하고자 하는 홈 네트워크 시스템의 근간을 이루는 UPnP 기술에 대한 구조 및 프로토콜의 흐름에 대해 설명한다.

### 2.1 UPnP의 구성요소

UPnP를 이루는 구성요소는 [그림1]에 나타난 바와 같이 크게 컨트롤 포인트, 디바이스, 서비스로 이루어지며 각각에 대한 자세한 설명은 다음과 같다.



[그림1] UPnP의 구성요소

#### 2.1.1 컨트롤 포인트

UPnP 네트워크의 컨트롤 포인트는 다른 UPnP 디바이스를 검색하여 제어하는 능력을 가진 컨트롤러이다. 컨트롤 포인트가 UPnP 디바이스들을 찾아낸 이후에는 디바이스 명세서를 검색하여 관련된 서비스 목록을 얻고, 관련되는 서비스의 서비스 명세서를 확보한다. 확보한 서비스 명세서를 바탕으로 컨트롤 포인트는 UPnP 디바이스를 제어하거나, UPnP 디바이스의 상태가 변경될 때 마다 생성되는 이벤트를 받기위해 자신을 UPnP 디바이스의 이벤트 서버에 등록한다.

#### 2.1.2 UPnP 디바이스

UPnP 디바이스는 사용자에게 제공하기 위한 서비스 및 부속 장치들을 내포하고 있다. 예를 들어 VCR 장치는 테이프 전송 서비스, 튜너 서비스 및 시계 서비스로 구성되어 있을 수 있고, TV/VCR 콤보 장치는 서비스 뿐만 아니라 다른 내장형 장치들로 구성되어 있을 수 있다.

각 UPnP 디바이스는 고유의 서비스 그룹과 내장형 장치를 가지고 있다. 예를 들어, VCR의 서비스는 프린터가 제공하는 서비스와는 다르다. 이러한 정보들은 해당 UPnP 디바이스가 호스팅하는 XML 장치 명세서에 포함되어 있다. 또한 명세서는 해당 디바이스와 관련된 디바이스의 속성정보도 포함하고 있다.

#### 2.1.3 서비스

UPnP 네트워크의 최소 제어 단위인 서비스는 UPnP 디바이스의 동작을 나타내고 있으며, 상태 변수를 통하여 자체의 상태를 모델화 한다. 이들 정보는 장치 명세서와 비슷하게 UPnP 포럼에 의하여 표준화된 XML 서비스 명세서의 일부를 구성한다.

UPnP 디바이스가 제공하는 서비스는 상태 테이블, 제어 서버 및 이벤트 서버로 구성된다. 상태 테이블은 상태 변수를 활용하여 서비스의 상태를 모델화 하며, 제어 서버는 동작 요청을 수신하여 실행하고, 상태 테이블을 업데이트하며, 그 결과를 반환한다. 마지막으로 이벤트 서버는 서비스 상태가 변경될 때 마다 이벤트에 관계된 컨트롤 포인트들에게 상태의 변경을 알려준다.

## 2.2 UPnP 네트워크 동작 단계 및 프로토콜

UPnP는 IP, TCP, UDP, HTTP, XML과 같은 기존의 프로토콜들을 기반으로 하여 동작한다. 디바이스간에 교환하는 데이터는 XML로 표현되고, HTTP를 통해서 통신한다. UPnP를 구성하는 핵심 프로토콜에는 SSDP, GENA, SOAP가 있다. SSDP(Simple Service Discovery Protocol)는 네트워크상의 서비스를 찾기 위한 프로토콜이며, GENA(General Event Notification Architecture)는 한 디바이스의 상태가 변했을 때 이를 다른 디바이스에게 알리기 위한 프로토콜이며, SOAP(Simple Object Access Protocol)은 디바이스가 다른 디바이스에게 제어 명령을 보내기 위해 사용하는 프로토콜이다.

UPnP를 통한 디바이스 간의 통신은 발견 단계, 기술 단계, 제어 단계, 이벤팅 단계, 프리젠테이션 단계로 나누어 진다.[6]

### 2.2.1 발견(Discovery) 단계

발견 단계에서 컨트롤 포인트는 서비스들을 찾고, 서비스들은 컨트롤 포인트에게 자신의 존재를 알린다. UPnP는 SSDP 프로토콜을 사용하며 이때 서비스 발견을 위해서 멀티캐스트 UDP 상에서 동작하는 HTTP와 발견에 대한 응답을 위한 유니캐스트 UDP상에서 동작하는 HTTP로 이루어진다.

### 2.2.2 기술(Description) 단계

일단 컨트롤 포인트가 서비스를 찾으면 TCP 상의 HTTP를 통해서 UPnP 디바이스 명세서를 요청한다. 이는 표준 HTTP GET 명령을 통해서 수행된다. UPnP 디바이스 명세서는 XML로 표현되며, 모델명, 모델 번호, 시리얼 번호, 제조자명, 제조자의 정보도 포함한다. 또한 UPnP 디바이스 명세서는 제어, 이벤팅, 프리젠테이션을 위한 URL이 내장된 디바이스 또는 서비스의 리스트도 포함한다.

### 2.2.3 제어(Control) 단계

컨트롤 포인트가 UPnP 디바이스의 명세서를 가져온 후, 컨트롤 포인트는 디바이스의 서비스 모듈에게 명령을 보낼 수 있다. 이를 위해서 컨트롤 포인트는 서비스

모듈의 제어 URL로 적합한 제어 메시지를 보낸다. 제어 메시지는 XML로 표현되며, SOAP 프로토콜을 사용한다.

### 2.2.4 이벤팅(Eventing) 단계

서비스에 대한 명세서는 서비스가 응답하는 명령의 리스트와 런타임 때 서비스의 상태를 나타내는 변수들의 리스트가 포함되어 있다. 서비스는 변수의 값이 변했을때 이 변수가 변했음을 알리며, 컨트롤 포인트는 이 정보를 받기 위해서 서비스에 이벤트 등록할 수 있다. 이벤트 메시지는 XML로 표현되며 GENA를 사용하여 구성된다.

### 2.2.5 프리젠테이션(Presentation) 단계

디바이스가 프리젠테이션을 위한 URL을 가지고 있다면, 컨트롤 포인트는 이 URL로부터 페이지를 가져와서 브라우저에 로드시켜, 브라우저를 통해 사용자는 디바이스를 제어하거나 디바이스 상태를 볼 수 있다.

## 3. UPnP 홈 오토메이션 시스템 설계 및 구현

이번 절에서는 앞에서 설명한 UPnP 미들웨어를 사용하여 웹기반으로 집안의 지능형 가전기기들을 제어하고 점검할 수 있는 홈 오토메이션 시스템을 설계하고, 구현하는 방법에 대해 설명한다.

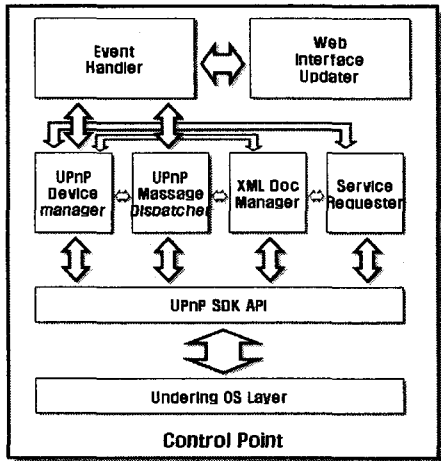
### 3.1 UPnP 홈 오토메이션 시스템 개요

UPnP 기반의 홈 오토메이션 시스템을 구현하기 위해 본 논문에서는 Fedora core 4 리눅스 환경에서 동작하는 인텔사의 "UPnP SDK for Linux 1.0.4"를 사용하였다. UPnP SDK는 개발자들이 리눅스 환경에서 컨트롤 포인트와 UPnP 디바이스를 개발할 수 있도록 프로토콜과 XML DOM 라이브러리, 그리고 기타 웹 관련 API를 제공한다.

본 논문에서는 UPnP SDK를 사용하여 홈 게이트웨이와 UPnP 디바이스를 위한 소프트웨어를 개발하였으며, UPnP 디바이스의 내장형 장치로써 센서 네트워크를 구축하여 검증하였다. 센서 네트워크를 구성하는 센서 노드는 Crossbow사의 Mica2[7] 모드를 사용하여 조도,

습도 값을 측정하여 홈 게이트웨이(컨트롤 포인트)에 보고 하도록 선정 하였다. 홈 게이트웨이는 UPnP 디바이스가 보고한 조도, 습도 값을 사용하여 센서 네트워크에 연결된 가전기기의 전원을 제어한다. 즉 측정된 조도값이 사용자가 이전에 설정한 특정 기준치 이하가 된다면 스탠드의 전원을 On하고, 기준치 이상이 된다면 스탠드의 전원을 자동으로 Off 해준다. 또한 수동으로 가전기기의 전원을 제어하거나 가전기기의 상태를 점검하기 위해 홈 오토메이션 시스템은 웹 인터페이스를 제공한다.

3.2 홈 게이트웨이(컨트롤 포인트) 응용 프로그램 구조

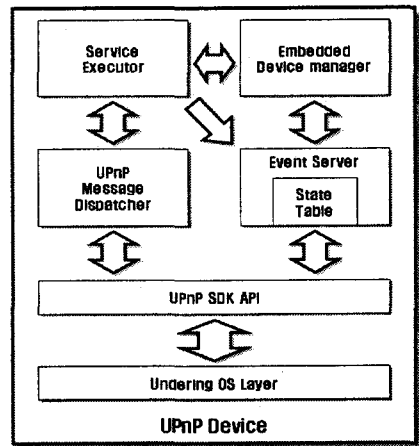


[그림2] 홈 게이트웨이 응용 프로그램 구조

홈 오토메이션 시스템의 홈 게이트웨이는 [그림2]과 같이 XML 문서 관리자, UPnP 메시지 해석기, UPnP 디바이스 관리자, 웹 인터페이스 업데이트, 서비스 요청기, 이벤트 핸들러로 이루어진다. 홈 게이트웨이 응용 프로그램을 구성하는 모듈들은 모두 UPnP SDK API를 사용하여 구현되었다. XML 문서 관리자는 기술(Description)단계에서 사용되는 모듈로써 UPnP 디바이스로부터 얻어온 명세서에 포함된 서비스 목록 및 상태 변수 정보등 UPnP 디바이스와 관련된 중요 정보들을 관리하고, 다른 모듈들이 이러한 정보를 요구하면 제공하는 역할을 담당한다. UPnP 메시지 해석기는 UPnP

디바이스가 보내온 UPnP 메시지를 해석하고, 해석한 메시지의 내용에 따라 해당되는 모듈을 호출해주는 역할을 담당한다. UPnP 디바이스 관리자는 발견(Discovery) 단계에서 새롭게 발견된 UPnP 디바이스들을 테이블로 관리하고, UPnP 디바이스 제어와 관련된 간단한 정보들을 저장하고 있다. 웹 인터페이스 업데이트는 UPnP 디바이스가 UPnP 네트워크에 새롭게 추가되거나 제거되었을때 홈 게이트웨이의 프리젠테이션 페이지를 갱신하는 역할을 담당한다. 그리고 서비스 요청기는 UPnP 디바이스를 제어하기 위한 명령을 생성하고 UPnP 디바이스에게 전달하는 모듈이다. 마지막으로 이벤트 핸들러는 서비스에 대한 응답이나 UPnP 디바이스의 상태변경과 같은 이벤트 발생시에 이벤트에 대한 적절한 처리를 해준다.

3.3 UPnP 디바이스 응용 프로그램 구조



[그림3] UPnP 디바이스 응용 프로그램 구조

UPnP 디바이스의 응용 프로그램은 [그림3]에서 보이는 바와 같이 UPnP 메시지 해석기, 상태 변수 테이블, 이벤트 서버, 내장형 디바이스 관리자, 서비스 실행기로 이루어진다. UPnP 디바이스 응용 프로그램을 구성하는 모듈들은 홈 게이트웨이의 응용 프로그램을 구성하는 모듈들과 마찬가지로 UPnP SDK API를 사용하여 구현되었다. UPnP 메시지 해석기는 홈 게이트웨이와 UPnP

메시지를 주고 받기 위해 UPnP 메시지를 생성하거나 홈 게이트웨이로부터 받은 UPnP 메시지를 해석하여 적절한 모뎀에 메시지를 전달하는 역할을 담당한다. 상태 변수 테이블은 UPnP 디바이스와 관련된 상태 변수들을 관리하는 테이블이며, 이벤트 서버는 이러한 상태 변수 테이블에 변화가 생기거나 홈 게이트웨이에게 보고해야만 하는 이벤트 발생시에 이벤트 서버에 등록된 홈 게이트웨이들에게 생성된 이벤트를 전송한다. 내장형 디바이스 관리자는 UPnP 디바이스에 연결된 센서 네트워크를 관리하기 위한 모듈로써 센서 네트워크에 연결된 가전기기를 제어하거나 센서 노드들에 의해 측정된 값들을 처리한다. 이러한 작업들을 수행하기 위해 내장형 디바이스 관리자는 센서 네트워크의 메시지를 해석할 수 있는 기능을 포함하고 있다. 마지막으로 서비스 실행기는 UPnP 메시지 해석기로부터 서비스 요청을 전달받으면 UPnP 디바이스가 제공할 수 있는 서비스인지를 판단하여 자신이 제공할 수 있는 서비스라면 서비스를 담당하는 내장형 디바이스에게 메시지를 전달한다.

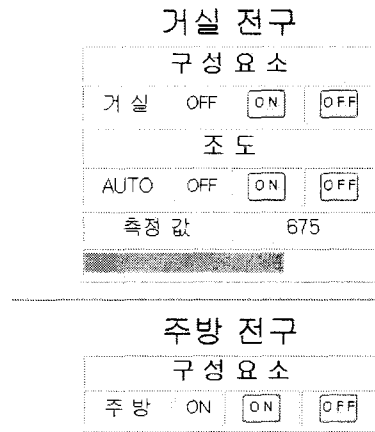
### 3.4 홈 오토메이션 시스템 수행과정

UPnP 네트워크상의 홈 게이트웨이가 SSDP 프로토콜을 사용하여 UPnP 디바이스를 발견하게 되면, UPnP 디바이스와 관련된 명세서를 UPnP 디바이스에게 요청한다. 이러한 요청에 대한 응답으로써 UPnP 디바이스는 XML 형식으로 구성된 명세서를 홈 게이트웨이에게 넘겨주게 된다. 홈 게이트웨이는 새로운 UPnP 디바이스와 관련된 명세서에 포함된 정보를 해석하여 얻은 정보를 UPnP 디바이스 매니저에 등록함으로써, 이후에 UPnP 디바이스를 제어하거나 상태를 점검할 수 있는 정보들(서비스 목록, 제어 URL등)을 확보하게 된다. 또한 홈 게이트웨이의 웹 인터페이스 업데이트는 새로운 UPnP 디바이스의 프리젠테이션 URL을 홈 게이트웨이의 프리젠테이션 페이지에 추가한다.

사용자는 홈 게이트웨이의 프리젠테이션 페이지에 링크된 UPnP 디바이스의 프리젠테이션 페이지를 통해 특정 디바이스를 제어하거나 상태를 점검할 수 있다. 본 논문에서 설계하고 구현한 시스템에서는 UPnP 디바이스로서 내장형 장치의 역할을 담당하는 센서 네트워크를 구축하였

으며, UPnP 디바이스의 프리젠테이션 페이지를 통해 센서 네트워크에 연결된 가전기기를 제어하거나 조도, 습도 센서의 값을 점검할 수 있다.

[그림4]은 센서 네트워크에 연결된 UPnP 디바이스의 프리젠테이션 페이지를 나타낸다. 현재 UPnP 디바이스에는 센서 네트워크를 사용하여 3개의 가전기기가 내장형 디바이스으로써 연결되어 있으며, 이 가전기기를 제어하거나 현재 상태를 표시하기 위한 인터페이스를 제공한다. 거실에 있는 전구는 조도 센서와 연동되어 센서에서 감지된 측정값을 확인하여 자동 제어 할 수 있다. 자동 제어 기능은 사용자가 설정한 조도의 기준값에 따라 자동으로 거실 전구를 제어하는 자동화 시스템이다.



[그림4] UPnP 디바이스 프리젠테이션 페이지

### 4. 결론

본 논문에서는 UPnP의 개념과 구조, 프로토콜의 수행과정을 살펴보고, UPnP 포럼에서 정의한 표준에 맞는 기능들을 제공하는 인텔사의 "UPnP SDK for linux"의 API들을 사용하여 센서 네트워크와 연동되는 UPnP 홈 네트워크 시스템을 구축하였다.

이러한 UPnP 미들웨어를 사용한 홈 오토메이션 시스템은 전문가의 도움없이도 새로운 지능형 가전기기를 쉽게 설치할 수 있을 뿐만 아니라 웹 인터페이스를 통하여 사용자는 어디서나 집안의 가전기기를 제어하고

점검할 수 있는 장점을 가지고 있다. 따라서 UPnP 미들웨어는 앞으로 홈 네트워크 분야에서 커다란 비중을 차지할 것으로 예상된다.

UPnP는 현재 마이크로소프트사의 주도하에 많은 기업들이 표준으로 지지하고 있는 홈 네트워크 미들웨어 개념이지만 Jini, HAVI 등 다른 홈 네트워크 표준들과 호환되지 않고 있다. 홈 네트워크 표준간의 호환을 위해서는 OSGi[8]와 같은 미들웨어가 필요하다. 향후 연구로는 OSGi를 사용하여 UPnP, Jini, HAVI 홈 네트워크 표준들을 통합할 예정이다.

#### 참고문헌

- [1] UPnP Forum, "UPnP News Letter, Second Quarter", 2002  
<http://www.upnp.org/news/default.asp#>
- [2] A.M. Brent, N. Toby, T. Charlie, and D.W. Mark, "Home networking with Universal Plug and Play," IEEE Communications Magazine, Vol.39, No.2, Dec. 2001, pp.104-109
- [3] Jussi Teirikangas, "HAVi: Home Audio Video Interoperability", 2001
- [4] Sun microsystems, "Jini™ Architecture Specification", Oct. 2000
- [5] Intel, "UPnP SDK for Linux",  
<http://upnp.sourceforge.net/>
- [6] 조충래, 박광로, "UPnP 기술 표준화 현황", [정보통신표준화] IT포털 ITFIND, Dec. 2002
- [7] Crossbow, "Mica2 Datasheet",  
<http://www.xbow.com/>
- [8] OSGi Alliance, "About the OSGi Service Platform Technical Whitepaper", July. 2004