

## UPnP를 이용한 전자 액자 서비스 구현

박진성<sup>o</sup> 김동환 박종서 최진구 이상호

한국산업기술대학교 컴퓨터공학과

e-mail : {pjs9153<sup>o</sup>, hwaneed, pjs0919, jkchey, sangho}@kpu.ac.kr

### Implementation of a Multimedia frame service using UPnP

Jin-Seong Park<sup>o</sup>, Dong-hwan Kim, Jong-seo Park, Jin-goo Choi, Sang-ho Lee

Dept. of computer Engineering, Korea Polytechnic University

#### 요 약

인터넷 보급화와 기술의 발달로 가정 내 가전기기들을 네트워크로 연결하여 서로의 정보를 공유한다. 분산 환경인 연결 동작을 지원하는 시스템을 홈 네트워크의 기술을 바탕으로 사이버 홈이라는 개념이 생겨나게 되었다. 그리고 가전기기들의 인터넷을 통하여 제어할 수 있는 기술들이 생겨났다. 로컬화 되던 가전 기기들이 디지털화되고 이기종간 연결을 위해서 미들웨어를 필요하다. 미들웨어를 사용하지 않을 때에는 서로 다른 가전기기들을 연결하여 사용하려면 중간에 Converter를 설치해야만 하는 불편한 점을 미들웨어가 해소해주게 되었다. 미들웨어로 UPnP와 Jini등 다양한 기술들이 생겨났다. 본 논문은 아직 미들웨어의 표준화가 되지 확정 않은 시점에서 미들웨어 하나인 UPnP를 택하여 가정내에서 전자 액자 서비스를 구현하고 그것에 대한 효용성을 보이고자 한다.

키워드 : UPnP( Universal Plug & Play )

#### 1. 서 론

현재 기업이나 공공단체에서 사용되던 네트워크 기술이 가정에 있는 다양한 정보 가전을 원격제어서비스 목적으로 인터넷과 연결하여 하나의 홈 네트워크기술로 발전하여 많은 사람들의 관심을 받게 되었다. 또한 로컬화 되던 가전기기들이 기종에 관계없이 연결해 주는 미들웨어들이 많은 연구와 개발로 생겨나게 되었다. 가정 내 가전기기들을 제어하기 위해 생겨난 미들웨어로는 마이크로소프트사에서는 네트워크에 연결된 가전기기들을 자동 제어하도록 설계된 UPnP(Universal Plug & Play), 소니를 주축으로한 일본과 유럽 기업들은 홈 네트워크 시스템으로 HAVi(Home Audio/Video Interoperability Architecture), SUN Microsystems사는 프로토콜에 관계없이 네트워크에서 주변 장치들의 식별을 가능하게하는 네트워크 분산기술 Jini(Java Intelligent Network Infrastructure)등이 있다. 국내에서 제안된 미들웨어로는 삼성전자에 의해 제안되어 VESA(Video Electronics Standards Association) 홈 네트워크 표준 및 EIA/CEA 775.1 표준으로 채택된 HWW(Home Wide Web)이 있다. 미들웨어들은 집이나 작은 사무실과 같이 네트워크 관리자가 없는 곳에서도 추가 작업없이 쉽게 표준화된 방법이다. 특히 UPnP는 개인용 컴퓨터(PC)나 인쇄기 등의 각종 주변 기기를 가정의 네트워크에 접속해 공유할 수

있도록 하는 기술로서, 일반 기기들이 자신의 고유한 IP 주소를가지고 인터넷 프로토콜을 이용한 기기들 간의 통신이 가능하도록 TCP/IP, 동적 호스트 설정 통신 규약(DHCP)등의 인터넷 관련 프로토콜을 사용하고, 그 구현은 확장성 생성 언어(XML)를 통해서 이루어진다. 기존의 플러그엔 플레이는 PC에 주변 기기를 접속할 수 있는 것이지만 범용 플러그 앤 플레이는 이를 확장하여 홈 네트워크에서도 적용하여 인터넷 서비스와의 연동이 용이하다는 큰 장점을 가지고 있다.

따라서 앞으로의 표준 미들웨어로는 외부인터넷의 연결과 많은 기기들간의 연결을 해결한 UPnP가 정해질 가능성이 높다고 보여진다. 본 논문에서는 이런 UPnP를 통하여 가정에 UPnP를 이용한 전자액자를 구현해 보이고자 한다.

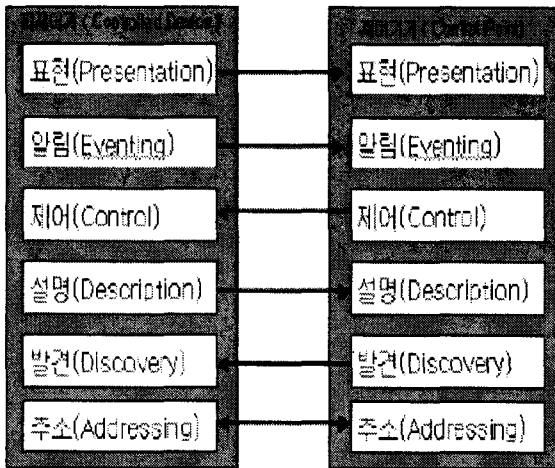
본 논문의 구성은 제 2장에서 본 논문의 핵심 기술인 UPnP에 대한 연구과 OSGi의 관한 내용에 대해서 논하고, 제 3장에서는 본 논문에서 구현된 설계 및 구현한 시스템에 관하여 논한다. 제 4장에서는 메시지 방식의 통신에서의 야기되는 문제점 그에 대한 해결방안능 모색에 대해서 논하며 제 5장에서는 구현 및 성능평가를 하고, 마지막 제 6장에서는 향후 과제 및 결론을 맺는다.

2. 관련연구

본 장에서는 UPnP에 대한 관련 연구에 대해서 논의하고자 한다.

2.1 UPnP

UPnP는 기존 PC에서 디바이스를 제어한 Plug & Play 개념을 조금 더 확장하여 사용자에게 어떤 작업도 요구하지 않고, 이기종간의 다양한 디바이스들을 네트워크에 접속시킨다. 그림 1과 같이 UPnP에서 피제어기는 Addressing과정에서 네트워크로부터 IP를 할당을 받은 후, Discovery과정에서 자신이 연결되었음을 다른 기기들에게 알린다. Discovery과정에서 찾은 기기들은 Control Point에게 자신의 특징이나 기능을 적은 Description을 XML이용하여 전달한다. Control Point는 기본적으로 디바이스 발견과 제어를 실행하기 위해 네트워크 전체를 계속적인 검색과 디바이스 이름 및 주소를 등록, 삭제를 기본으로 제어기능을 행한다. 또한 Discovery기능을 수행하고 들어오는 메시지들을 분석하는 Parser부분과 매칭하는 부분이 필요하다. Message분석은 HTTP Parser와 XML Parser에서 분석을 행한다. 각각의 Parser에서 넘어온 값을 매칭하는 과정을 통해 디바이스 등록과 삭제가 이루어진다. Control Point에 전달받은 XML은 그 내용을 통지하는 Control과 디바이스 상태가 변했을 때 그 내용을 컨트롤 포인터로 통지하는 Eventing, 디바이스의 상태를 체크하는 Presentation으로 구성되어져있다.

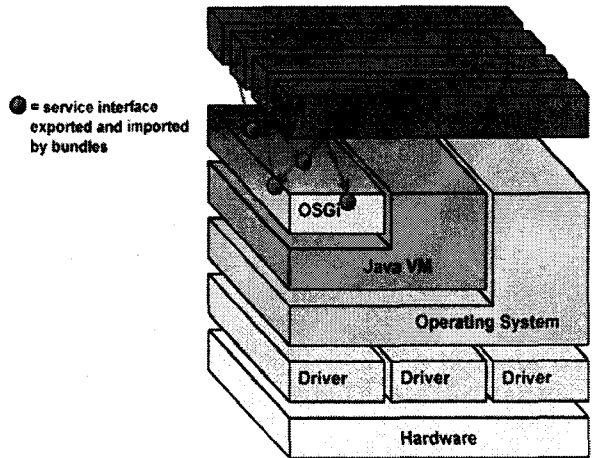


( 그림 1 ) UPnP 프로토콜의 메시지 단계 구조

2.2 OSGi

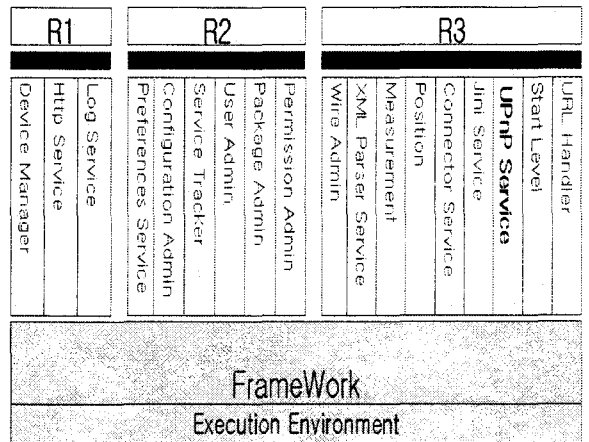
OSGi(open service gateway initiative)는 다양한 환경에서 운용가능한 개방형 서비스 플랫폼 제공을 목표로 1999년에 설립된 국제 표준화 단체이다. 현재 IBM과 SUN Microsystems를 비롯한 전세계 40여 유명회사원사를 확보하고 있으며, 여러 가임기업들은 산업계에서 직접 적용 가능하기 위해 활발한 활동을 전개하고 있다. OSGi는 미들웨어라 부르기 보다는 미들웨어도 응용프로

그램간의 API를 정의하고 어떤 특정 응용프로그램과 미들웨어들을 분리할 수 있는 역할을 담당한다. OSGi의 내부구조는 그림 2와 같다. 그림에서 각 번들 서비스들은 OSGi의 프레임워크의 서비스가 등록기에 등록되어 다른 서비스가 등록된 서비스의 서비스 인터페이스( Service interface )를 통해 사용할 수 있다. 등록된 각 번들들은 생명주기( life cycle )를 관리하는 번들 활성화기( Bundle Activator )와 다른 서비스로부터의 참조를 위한 서비스 인터페이스(service interface) 그리고 인터페이스들을 구현한 서비스 구현( service implement )으로 구성된다. 끝으로 번들 문맥( Bundle context )는 각 번들과 OSGi 서비스 프레임워크 간의 인터페이스 역할을 담당한다. OSGi는 JAVA를 프로그램 중심으로 구현을 하는데 이는 JAVA언어의 이식성에 중점을 두고있다.



( 그림 2 ) OSGi 내부 구조

그림 3번은 OSGi내의 프레임 워크의 구조를 보여준다. 현재 R1, R2, R3는 개발 버전을 뜻한다. 현재 R4까지 개발되어있고 본 논문에서 제시한 UPnP부분은 R3버전 때에서부터 프레임워크에 포함되어서 상용화 하고있다.

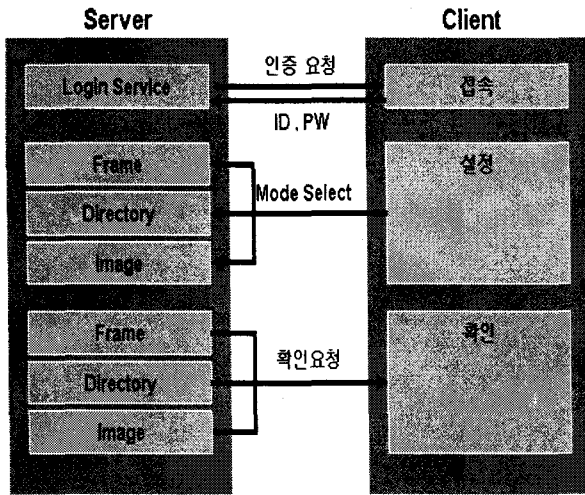


( 그림 3 ) OSGi 내부 FrameWork

### 3. Home Server 및 Multimedia Frame 구현

#### 3.1 Home Server 구현

Home Server는 HTTP 서비스를 제공하여 인터넷을 통하여 외부로 연결되어있으며 다수의 Frame들을 선택 및 제어하기 위한 프레임 관리와 서버의 Directory와 이미지 파일들을 삽입, 삭제, 변경하기 위한 Directory관리, 그림 파일들을 관리하는 Image Manager로 구성되어져있다. Servlet은 집안 내 Multimedia Frame이 UPnP통신을 하여 자동연결 및 이미지 재생제어기능을 제공한다. Home Server의 주요 모듈로는 Frame Manager 모듈과 Image Manager 모듈로 구성되어져 있다. Frame Manager 모듈은 Frame과 연결을 위하여 Multimedia Frame들의 상태 메시지를 전달받는다. 전달 받은 상태 메시지들은 인터넷을 통한 클라이언트들에게 보여준다. 그림 4는 Home Server가 외부에서 사용자가 접근을 하였을 경우 Home Server에서는 인증을 요청하여 클라이언트의 ID와 Password를 입력받아서 저장된 내용과 비교하여 접속을 허가 및 거부를 한다. 승인된 사용자는 제어를 하려는 Frame을 선택하여 외부로부터 이미지 파일들을 업로드 다운로드 할 수 있다.

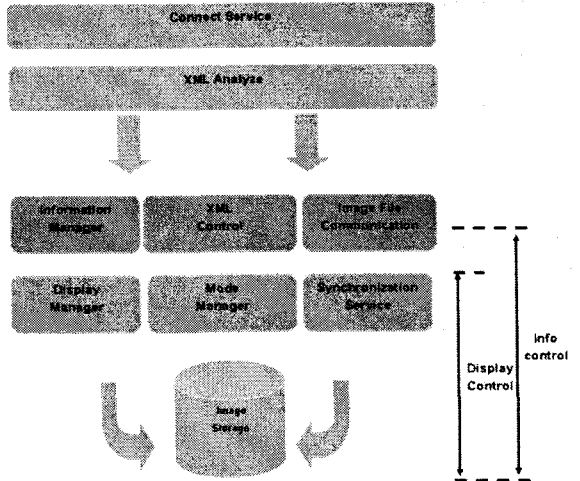


( 그림 4 ) Home Server와 외부 Client와의 관계 구조

#### 3.2 Multimedia Frame

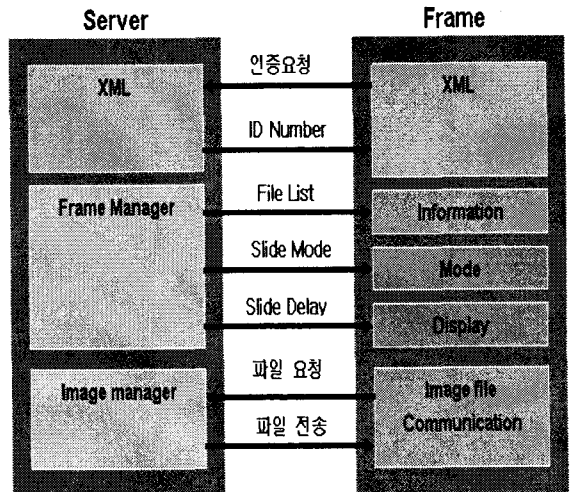
Multimedia Frame은 최초 기기가 홈 네트워크에 연결 했을시 XML을 통하여 Server로 인증을 요청하면 ID와 Number를 부여받는다. Multimedia Frame의 구조는 그림 5 과 같다. Frame은 이미지파일의 정보를 관리하는 Information manager 모듈과 이미지를 화면에 출력시 그 화면의 상태를 관리하는 Mode Manager 모듈, 마지막으로 서버와의 Image파일을 동기화하고 자신의 상태를 전달하는 Display Manager 모듈로 구성되어져있다. Server의 Frame 모듈은 Frame의 Information 모듈과, Display모듈로부터 통신을 수행하는데 Information Manager 모듈은 TCP망으로 Frame의 현재 이미지 리스트의 상태를 제어할 수 있는 기능들을 알린다. Mode Manager 모듈은 Frame의 전달된 사용자의 명령을 받아들여 Display 모드와 리스트를 제어하는 기능을 가지

며, 파일리스트 값들을 변경 또는 이미지 파일을 전달받는다. 그림 6는 Home Server와 Multimedia Frame간의 내부 관계구조이다.



( 그림 5 ) Multimedia Frame 구조

그림과 같이 Server의 Frame Manager에서 파일리스트와 슬라이드 모드, 시간 딜레이시간을 클라이언트에게서 입력받으면 그 설정값을 Frame에게로 전송되어서 그 설정에 맞게 Frame이 변화한다. 또한 Home Server에서 이미지 파일이 새롭게 추가가 되면 그 파일을 Frame과 동기화되어서 전송하면 그 파일을 프레임은 화면에 출력하여 보여준다.



( 그림 6 ) Home Server와 Multimedia frame과의 관계구조

### 4. 메시지 방식의 문제점

Control Point에서 디바이스들로 제어 및 메시지 처리를 보내거나 디바이스들의 새롭게 자신을 알리는 처리를

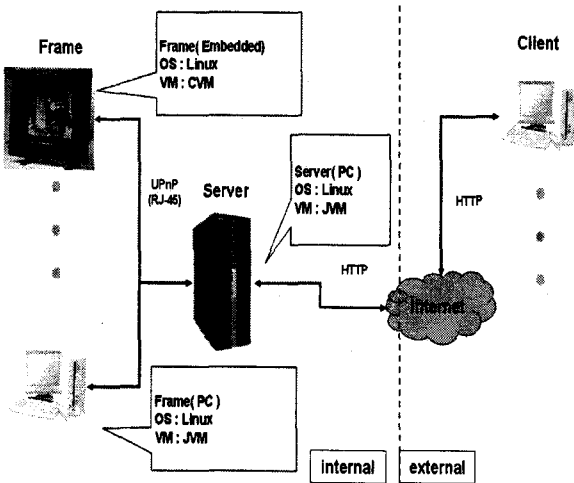
할 때에 1~3개의 디바이스가 아닌 더욱 많은 디바이스들이 메시지를 전송하였을 때 네트워크상에서 과부하 패킷이 발생 할 수 있다는 단점이 있다. 그럴 경우에 전달상의 문제발생과 오류 발생, 정상적인 메시지가 아닌 손실된 메시지가 발생하여 혼란이 야기될 수 있다는 문제점이있다. 이 문제에 현재 많은 연구가 진행되고있다. 근본적인 문제는 해결되지 않았지만, 임시방편으로 생각하면 네트워크부하와 관계없이 메시지를 로그화 함으로서 패킷이 손실되는 문제를 해결할 수 있다.

5. 구현 성능 평가

이제까지 이론적으로 서술해왔던 Home Sever부분과 Multimedia Frame부분을 JAVA언어로 사용하여 소프트웨어들을 LAN을 통하여 서버를 포함한 3대의 기기들로 구성하여 하나의 네트워크를 구성하였다.

5.1 실제 환경 구성

전체 시스템 구조는 그림 5와 같이 가정 내는 Home Server와 Multimedia Frame 2대로 구성된다. 프레임은 각각 UPnP의 기능을 실현시켜보고자 OS는 Linux(Kernel Version 2.6.4)로 구성하고, 디바이스만 달리하였다. 1개의 Frame은 Embedded를 채택하였고, 다른 하나는 PC를 사용하였다. Embedded System에는 PXA255에 SVM을 설치하였다. 현존하는 VM(Virtual Machine)중 JVM(Java Virtual Machine)과 가장 유사한 기능을 가진 것으로서 다양한 기능 및 재사용성을 고려하였다. PC는 Home Server와 같이 JVM으로 구성하였다. Multimedia Frame은 그래픽이미지를 출력하기위한 슬라이드 기능과 UPnP를 사용하여 Home Server와 연결구성하였다. Home Server에 탑재한 소프트웨어는 Servlet과 HTTP를 사용하여서 가정내에 Multimedia Frame기기의 상태 확인 및 제어를 한다.



( 그림 7 ) 전체 System 구조도

5.2 실험 결과 및 분석

실제 네트워크상에서 이루어지는 작업으로 몇 초간의 지연시간이 발생하였다. XML에 작성된 디바이스의 이름,

번호, 식별사항이 정확히 일치해야만 정상적인 작동을 하지만, 일치하지 않으면 다른 디바이스로 판단한다. 또한 디바이스들이 한정되어있을 때에는 문제가 발생하지 않았으나 디바이스들이 많으면 많아질수록 전송에 따른 지연이 발생하는 문제가 있었다. IP가 각각 개별로 만들어지다 보니 어느 것이 먼저 등록이 되는가에 따라 주소가 변하게 되었다. 주소가 변할 때마다 다시 전체를 읽어주는 것으로 문제를 해결하였다.

6. 결론 및 향후과제

기존의 연구를 보더라도 UPnP는 Jini와는 전혀 다른 XML 메시지를 사용함으로써 가전기기 제어능력은 큰 저장공간을 필요치 않으며, Jini보다 네트워크 부하를 줄일 수 있다. 또한 어느 곳이든 지원이 되는 콘텐츠를 연결하여 사용할 수 있는 기술이라고 보여진다. 본 연구에서는 기존의 TCP/IP를 모두 수용하지만 UPnP 기능이 내장되지 않은 가전기기들을 UPnP에 연결하기 위해 개발돼 Bridge 부분을 개발 연구하고, 외부 인터넷 망을 통한 동적 제어기술인 OSGi를 이용하면, 미들웨어의 표준이 될 것으로 전망된다. 결론적으로 이미 많은 분야에서 실시하였던 부분을 구현으로 앞으로의 홈 네트워크의 시장을 엿볼 수 있다.

7. 참고문헌

- [1] UPnP forum. "[www.upnp.org](http://www.upnp.org)"
- [2] knoflerfish.org "<http://www.knopflerfish.org>"
- [3] OSGi Service Platform Release 3 Specification "<http://www.osgi.org>"
- [4] UPnP SDK for Linux "<http://sourceforge.net/projects/upnp>"
- [5] Microsoft Windows XP "<http://www.microsoft.com>"
- [6] Shannon Chan Microsoft, Alec Dara-Abrams Sony Electronics, Mike Dawson OpenGlobe, John Kai Fu Pioneer, Fernando Matsubara Mitsubishi Electric, Jean Moonen Philips Electronics, Yasser Rasheed Intel, Dale Sather Microsoft and Eugene Shteyn Philips Electronics, "ConnectionManager 1.01" Intel June 25 2002.