

UPnP-to-Jini 서비스의 설계 및 구현

김철민, 은성배, *한상숙
한남대학교 정보통신공학과, *대전기능대학교 멀티미디어학과
전화 : 042-627-8886 / 핸드폰 : 016-779-3622

Design and Implementation of UPnP-to-Jini Service

Chulmin Kim, Seongbae Eun, Sangsuk Han
Dept. of Information & Communication Engineering, Hannam University
Email : cmkim@daniel.hannam.ac.kr

Abstract

We investigate the middleware technology which will take a major role of extending the usage of home networking. UPnP and Jini belong to the most important technologies among them. In this paper, we suggest the interoperability between UPnP and Jini. We describe the design and implementation of UPnP-to-Jini service which makes it possible for a Jini client to utilize a UPnP device.

I. 서론

네트워크 기술이 발달 하고 컴퓨터 및 가전 기기가 점차 지능화 되어 정보 기기화 되면서 가정 내에서 정보 기기들을 연결하고 이를 외부 네트워크와 연결한 홈 네트워크가 발달 하게 되었다.

홈 네트워크를 구성하는 요소 기술은 크게 세 부분으로 나눌 수 있는데 첫째, 외부 네트워크와 홈 네트워크를 연결하는 xDSL, Cable Modem, 전력선 통신, FTTH 등의 기술이 있다. 둘째, 정보 기기들을 연결 하는 HomePNA, USB, Ethernet, IEEE1394, Bluetooth, WirelessLAN, HomeRF 등의 홈 네트워크 접속 기술이 있다. 셋째, 연결된 정보 기기들의 상호 연결, 상호 운용을 제공하는 UPnP, Jini, HAVi, OSGi 등의 기술이 있다.

홈 네트워크를 위한 접속 기술들은 점차 발전되어 점차 더욱 낮은 가격으로 구성 할 수 있게 되어 홈 네트워크 확장을 촉진하는 원동력이 되고 있다.

홈 네트워크 미들웨어 기술은 홈 네트워크를 구성하는 기기의 상호 연결, 상호 운용 기능을 제공하여 홈 네트워크의 유용성을 극대화 시키는 기술로서 현재는 다양한 미들웨어 기술들이 제안되어 사용 되고 있다. 홈 네트워크의 발전을 위해서 다양한 홈 네트워크 미들웨어 기술의 상호 호환성을 필요로 한다. 현재 호환성을 위한 기술로서 OSGi 등의 연동 기술이 제시되고 있으나 미들웨어 기술간의 직접적인 연동 기능보다는 각각의 기술을 통합하는 기술적 한계를 지니고 있다.

본 논문에서는 UPnP 서비스와 Jini 서비스의 연동 방법을 제시하고 UPnP를 지원하는 기기를 Jini 서비스를 통해 사용 가능하도록 하는 UPnP-to-Jini 서비스를

를 설계하고 구현 하였다.

II. 배경

1. UPnP(Universal Plug and Play)

1-1. 개요

UPnP(Universal Plug and Play)는 1999년에 Microsoft, Intel 등의 회사가 모여 제안한 기술로서, 정보 가전 기기가 네트워크에 참가하고 서비스를 제공하는 과정을 정의한 것이다. UPnP는 쉬운 설치, 쉬운 사용, 유연성, 확장성을 목표로 설계 되었다.

UPnP 네트워크는 Device, Service, Control Points 의 세 부분으로 구성 된다.

Device는 Device 또는 Service를 포함하는 정보 가전 기기를 의미한다. Device는 UPnP 네트워크에 참가하는 독립적인 단위로 볼 수 있으며 포함 하고 있는 Device 또는 Service를 통해서 UPnP 서비스를 제공 한다.

Service는 Device가 제공하는 실제적인 서비스를 의미하며 기기의 제어 및 관리를 할 수 있는 서비스를 제공한다.

Control Points는 Device가 제공하는 Service를 사용할 수 있는 능력을 갖는 것을 의미 한다. Control Points는 Device Description를 통해 UPnP 기기로부터 서비스 리스트를 획득하여 서비스를 사용할 수 있으며 다른 서비스로부터의 이벤트를 받아서 상태 변화를 알 수 있는 기능을 갖고 있다.

UPnP은 TCP/IP, Http, XML과 같은 인터넷에서 사용되는 표준을 이용하여 설계 되었다. TCP/IP는 UPnP 네트워크에 참가한 기기들의 기본 통신 프로토콜이고 TCP/IP상에서 WEB과 XML을 통해 기술된 Description, Control 등의 정보를 주고 받아서 동작 하게 된다.

1-2. 기본 동작과정

가. Control Points 와 UPnP 가능 기기

Control Point는 네트워크에 참여한 UPnP 가능 기기가 제공하는 서비스를 이용할 수 있는 지점이다. 사용자는 일반적으로 PC, PDA에 포함된 Control Points를 통해서 네트워크에 존재하는 UPnP 가능 기기의 서비스를 사용 한다.

UPnP 가능 기기는 UPnP를 통해 서비스를 제공할 수

있는 정보 가진 기기를 의미하며, UPnP 규격을 지원하여 서비스를 제공하는 주체이다.

ㄴ. Discovery

UPnP 가능 기기는 SSDP(Simple Service Discovery Protocol)를 이용하여 자신이 네트워크에 참가했음을 알린다.

SSDP는 HTTP를 기반으로 메시지를 주고 받으며 네트워크에 참가한 기기의 존재를 알리는 방법이다. SSDP에서 이용되는 HTTP는 UDP를 이용하여 전송하는 HTTPU(HTTP Unicast over UDP)와 멀티캐스트 UDP를 이용하여 전송하는 HTTPMU(HTTP Multicast over UDP)로 확장 되어 사용 된다.

ㄷ. Description

UPnP 가능 디바이스가 네트워크에 참가했음을 알린 후 제어 포인트는 접속된 기기로부터 자세한 서비스 내용을 얻어오는 작업을 한다. HTTPU, HTTPMU를 통해 정보를 받는다.

UPnP 가능 기기가 제공하는 Description 정보를 XML을 통해 기술 된다.

ㄹ. Control

Control Points는 Description을 통해 UPnP 가능 기기가 UPnP 네트워크에 참가 했음을 알게 되고 참가한 UPnP 가능 기기에 명령을 내릴 수 있다. 이것을 Control 이라 한다.

Control은 SOAP(Simple Object Access Protocol)을 이용한다. SOAP는 HTTP를 통해 전달 되며 XML로 기술된 제어 정보를 전달 한다.

그림 1은 UPnP의 동작과정을 보이고 있다.

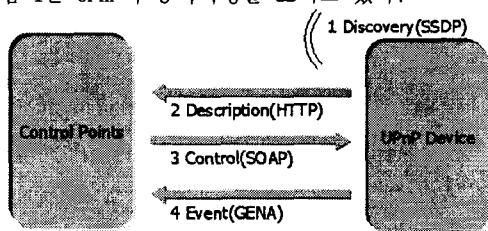


그림 1 UPnP의 동작 과정

2. Jini

2-1 개요

Jini는 썬 마이크로 시스템 사에서 개발한 미들웨어로서 Java를 기반으로 하여, 다양한 방식으로 망에 접속된 디지털 장비나 소프트웨어를 동적으로 상호 작용하도록 하겠다는 기술이다. JVM(Java Virtual Machine) 상에서 동작하기 때문에 운영체제나 하드웨어 플랫폼에 무관하게 동작하며 Plug and Play를 지원 한다.

지니는 Lookup Service, Service, Client 및 Network의 네 부분으로 나뉜다. Lookup Service는 Service의 등록을 받거나 Client의 사용을 가능하도록 Service를 관리하는 관리 서버의 역할을 한다. Service는 실제적인 기기에 포함된 것으로 서비스를 제공한다. Client는 Service를 사용하는 부분이다. 마지막으로 Network는 이전 세 개의 부분을 유기적으로

연결을 지원하는데 Jini의 설정은 Network 전송 매체나 프로토콜은 규정하지 않고 있지만 현재까지 Jini 관련 Library는 TCP/IP를 이용하도록 구현되어 있다.

2-2. 기본 동작과정

ㄱ. Discovery

Jini Service가 Jini 네트워크에 참가하면 Jini 서비스는 Lookup Service를 찾는 작업을 한다. 이를 Discovery라 한다.

ㄴ. Join

Lookup Service를 찾으면 Jini Service는 자신이 제공하는 서비스를 Lookup Service에 등록 한다.

ㄷ. Lookup

Jini Client가 Lookup Service로부터 Jini Service를 검색 하는 단계이다.

ㄹ. Lease

Jini Client가 Jini Service로부터 서비스를 제공 받는 단계 이다.

ㅁ. Control

Jini Client가 Lease를 통해 받은 Service는 Proxy, RMI등의 방식으로 사용 할 수 있다. Proxy 방식은 Client에 Service class 파일을 받아서 수행 하는 방식이고 RMI 방식은 Client는 호출만 담당하고 실제 Service의 수행은 Jini Service에서 일어나는 방식이다.

그림 2는 Jini의 동작과정을 보이고 있다.

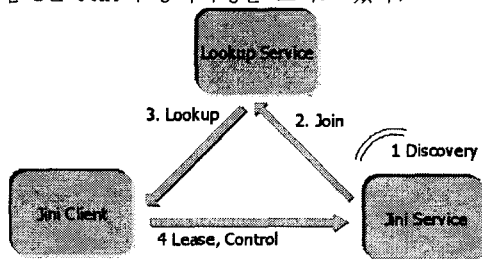


그림 2 Jini의 동작 과정

3. UPnP 및 Jini 의 서비스 시나리오

그림 1과 2에서 볼 있듯이 UPnP와 Jini의 대략적인 동작 과정은 매우 유사하다고 볼 수 있다. 또한 UPnP와 Jini의 구현 문서를 보면 구현 목표도 매우 유사할 수 있다.

UPnP의 경우 UPnP 서비스의 기능 명세를 XML을 통해 하고 있어 기능 명세 파일을 분석하여 UPnP 기기를 사용하는 범용적인 Control Point를 만들 수 있다. 이를 통해 UPnP 서비스를 Jini 서비스에서 사용할 수 있도록 하는 UPnP2Jini 서비스를 설계 하였다.

III. 설계

1. 기본 구조

UPnP-to-Jini 서비스는 Java 프로그램 및 C/C++ Library로 구현되며 크게 세 부분으로 나뉜다. 그림 3

은 UPnP2Jini 서비스의 기본 구조를 나타내고 있다.

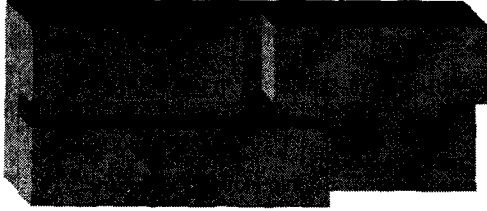


그림 3 UPnP2Jini 서비스의 기본 구조

2. Control Points 부

UPnP의 Control Points의 기능을 하는 부분으로 UPnP 네트워크에 UPnP 가능 기기가 참가할 경우 UPnP 기기로부터 Description 정보를 얻어서 참가한 UPnP 기기가 제공하는 Service 별로 Action List와 State List를 얻은 후 Service 생성자 부를 호출 한다. Linux SDK for UPnP Devices 1.2.1를 이용하여 Linux 환경에서 동작하는 Shared Library로 작성 한다.

3. Service 생성자 부

Control Points 부로부터 받은 Action List와 State List를 Jini 서비스 템플릿을 이용하여 실제 사용 가능한 Jini 서비스를 생성하고 실행하는 부분이다.

서비스 템플릿은 Sun에서 제공하는 Jini 1.2.1을 이용하여 Java로 작성 하고 이것의 수행은 Jini의 수행 방법을 따른다. C/C++로 작성된 Control Point 부와 Java로 작성된 Service 생성자부의 연결은 Java에서 제공하는 JNI를 통해 구현 한다.

4. 생성된 Jini 서비스 부

Service 생성자 부를 통해 생성된 후 수행 되면 생성된 Jini Service는 Jini Lookup Service에 등록된 후 Jini 서비스를 제공할 준비를 마친다. 이후에는 Jini 네트워크의 동작 과정을 따른다.

그림 4은 UPnP-to-Jini 서비스의 동작 개념을 나타낸다.

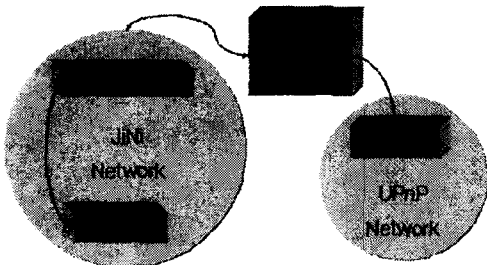


그림 4 UPnP-to-Jini 서비스의 동작 개념도

IV. 구현

1. 구현 환경

Redhat Linux 8.0 상에서 gcc 3.2, jdk 1.4.2_02 를 사용하고 Linux SDK for UPnP Devices 1.2.1, Jini Sdk 1.2.1를 통해 UPnP-to-Jini 서비스를 구현 하였다.

2. 구현

다음의 두 XML 파일은 UPnP 기기인 Light UPnP 서비스의 Description파일의 일부이다. Control Point 부 문은 UPnP 기기로부터 얻을 수 있는 XML파일을 통해 UPnP 기기의 Action List, State List를 얻을 수 있다.

Linux SDK for UPnP Device 1.2.1에서 제공하는 UPnP 관련 API와 XML 파일 처리관련 API를 사용하여 작성 하였다.

다음은 UPnP Light 기기의 상세 명세를 나타 내는 Description 파일의 일부를 보인다.

```

Description.xml File
...
<URLBase>http://192.168.0.121:56463/</URLBase>
<device>
  <deviceType>urn:schemas-upnp-
org:device:BinaryLight:1</deviceType>
  ...
  <UDN>uuid:92cce466-8a35-4d74-b13e-8e2150833ae2</UDN>
  ...
  <serviceList>
  <service>
    <serviceType>urn:schemas-upnp-
org:service:SwitchPower:1</serviceType>
    <serviceId>urn:upnp-
org:serviceId:SwitchPower.0001</serviceId>
    <SCPDURL>_SwitchPower.0001_scpd.xml</SCPDURL>
    <controlURL>_SwitchPower.0001_control</controlURL>
    <eventSubURL>_SwitchPower.0001_event</eventSubURL>
  </service>
  </serviceList>
</device>
...
    
```

다음은 UPnP Light 기기가 제공하는 SwitchPower 동작에 관련된 Action List, State List를 나타내고 있다.

```

_SwitchPower.0001_scpd.xml File
...
<actionList>
  <action>
    <name>GetStatus</name>
    <argumentList>
      <argument>
        <name>ResultStatus</name>
        <direction>out</direction>
      </argument>
    </argumentList>
  </action>
  ...
</actionList>
<serviceStateTable>
  ...
  <stateVariable sendEvents="no">
    <name>Target</name>
  </stateVariable>
  ...
    
```

```
<dataType>boolean</dataType>
</stateVariable>
</serviceStateTable>
```

3. 동작 시나리오

3-1. 그림 4과 같은 UPnP 기기인 UPnP Light 기기를 동작 시킨다. UPnP Light 기기는 Intel에서 제공하는 UPnP 디바이스 예제로서 Windows 상에서 동작하는 프로그램이다.

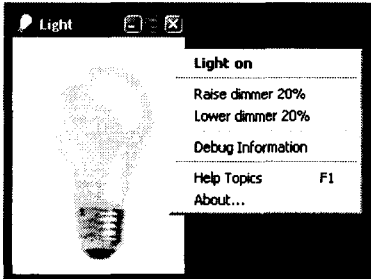


그림 5 UPnP Light 서비스

3-2. Jini 서비스를 수행 하기 위한 각종 서버들을 수행 시킨다.

3-3. UPnP-to-Jini 서비스를 수행 시킨다. 다음과 같이 UPnP 기기로부터 Action List와 State List정보를 얻게 된다.

```
java -cp jini-core.jar:jini-ext.jar:sun-util.jar:./
-Djava.rmi.server.codebase=http://192.168.0.122:8085/
-Djava.library.path=./
-Djava.security.policypolicy.all UPnP2Jini
Initializing UPnP with IP Address = (null) Port = 0
UPnP Initialized (192.168.0.122:-16384)
Registering Control Point
Control Point Registered
.Found Device : [urn:schemas-upnp-org:device:
BinaryLight:1]
Location : http://192.168.0.121:56463/
Found Service : urn:schemas-upnp-org:service:
SwitchPower:1
ServiceId : urn:upnp-org:serviceId:
SwitchPower.0001
SCPD URL : _SwitchPower.0001_scpd.xml
XML Read Success
Action Name : SetTarget
Argument Name : newTargetValue
Relate Variable : Target
State Variable Name : Target
Data Type : boolean
```

3-4. Jini Client를 수행 한다. Jini Client는 임의의 위치에서 수행 할 수 있다.

3-5. Jini Client를 통해 Jini 네트워크에 참가한 UPnP-to-Jini 서비스를 사용 한다. Client를 통해 UPnP Light 서비스에 UPnP Control이 전달 되어 전등이 켜지게 된다. 그림 6은 동작 후 UPnP Light의 상태를 나타내고 있다.

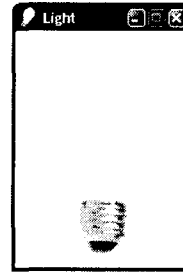


그림 6 UPnP Light 서비스의 동작 모습

V. 결론 및 향후 연구 방향

홈 네트워크의 사용 확대를 위해 홈 네트워크에서 사용되는 마들웨어인 UPnP와 Jini의 설계 목표와 동작 과정이 유사하고 UPnP 기기의 세부 사항을 XML을 사용하여 기술하는 UPnP의 특성을 이용하여 UPnP-to-Jini 서비스를 설계하고 구현 하였다. 이를 통해 홈 네트워크에서 사용되는 Jini에 UPnP와의 연동 기능까지 폭넓게 제공할 수 있는 장점을 얻을 수 있게 되었다.

구현의 편의를 위하여 Action List를 제한하여 구현 하였으나 향후 이러한 제한 없이 UPnP 기기가 제공하는 모든 Action List를 제공하는 UPnP-to-Jini 서비스를 구현하는 방안이 필요하다.

또한 복잡한 Action의 지원 방안에 대해서 연구할 필요성이 있다.

참고 문헌

- [1] UPnP Forum의 기술 문서, <http://www.upnp.org/resources>
- [2] Intel SDK for UPnP Devices (Linux), <http://upnp.sourceforge.net>
- [3] Jini Network Technology, <http://java.sun.com/jini>
- [4] 지니 테크놀러지 프로그래머 가이드, Jan Newmarch 저, 인포북 2001년
- [5] 지니 인어넷셀, 스코트 오스크, 헨리원저, 한빛미디어 1999년