홈 네트워킹 다중 사용자간 디바이스 잠금

신근태°, 김윤삼, 조은선
충남대학교 전기전자통신 컴퓨터 공학부
shinkt_m@nate.com, {bijak,eschough}@cnu.ac.kr

Locking Controlled Device between Multi-users on Home Networking using UPnP Architecture

Keuntae Shin°, Yunsam Kim, Eunsun Cho
Dept. of Computer Engineering, Chungnam National University

요 약

홈 네트워킹은 검색, 광고, 제어, 이탈 등 여러 가지 기능적 요소로 이루어진다. 이러한 기능 중 장치 제어의 경우 사용자에 의해서 제어 되는 장치는 특정 사용자에 의해서만 조작 되어야 하는 필요가 있다. 본 논문에서는 다중 사용자간 특정 장치를 다른 사용자가 사용하지 못하도록 보호하는 방법을 제안하기 위해 UPnP architecture 를 이용 한 홈 네트워크 구현 후 특정 사용자에 의하여 장치가 제어 되도록 하는 디바이스를 잠금 하는 방식에 대해 소개하고자 한다.

1. 서 론

최근까지 홈 네트워크에 연결된 가전제품을 제어 하는 대부분에 구현 사례는 PLC(Power line communication)에 많은 의존을 하고 있다. 그러나 이는 매우 하드웨어 의존 적인 기술이라 제공 할 수 있는 제어 정보 자체 양이 극히 드믈 뿐더러 하드웨어 제조사와 PLC 보드 간에 긴밀한 협업이 필요 하고 제공되는 기능에 대한 수정이 어렵다는 것이 단점 이라 할 수 있다.

이에 반해 소프트웨어 적인 조작이 용이하도록 풍부한 서비스를 제공하는 홈 네트워크 미들웨어와 실제 미들웨어를 이용한 네트워킹을 할 수 있는 장비가 눈에 띄게 시장에 진입을 하고 있는 상황이다. 그 예로 UPnP[5]를 들 수 있다. UPnP는 Universal Plug and Play 의 약자로 마이크로소프트사가 제안하고 홈네트워킹에 사용 되는 미들웨어이다. 이 UPnP architecture 1.0 를 이용한 디지털액자, 휴대용 멀티미디어 플레이어, 회선 공유기, 프린터 등의 제품들이 개발되었다.

본 연구 에서는 홈네트워킹 환경에서 UPnP 프로토콜과 네트워크 방식을 이용한 장치에 대한 조작이 이루어 졌을 때 특정 사용자에 의해 장치에 대한 독자적인 조작 권한을 갖기 위한 잠금 방식을 제안한다. 제안 방식은 다른 Access Control 등에 비하여 UPnP architecture 1.0가 사용되는 시스템에 대한 부하를 줄일 수 있다.

2장에서는 기본적인 UPnP의 동작 과정을 설명 하며, 3장에서는 잠금 장치의 적용을 위한 시나리오와 함께 잠금 방식 구현에 대한 기법을 제시하고, 4장에서는 구현 결과를, 5장에서는 관련 연구를 6장에서는 결론을 제시 하고자 한다.

2. UPnP의 기본 구조

UPnP는 그림 1과 같은 단계를 통해 UPnP 네트워킹을 하도록 제시 하고 있다. 장치가 IP 주소를 할당 받고, 홈 네트워크 환경 내의 제어 가능 한 장치를 찾고, 해당 장치에 대한 제어 정보를 습득 후, 제어 정보를 기반으로 피제어 장치를 제어하면, 피제어 장치가 제어된 그 결과에 의해 수정된 자신에 정보를 통지하며, 현재 피제어 장치에 대한 상태 정보를 보여 준다. 각각은 Addressing, Discovery, Description, Control, Eventing, Presentation stage로 구현된다.

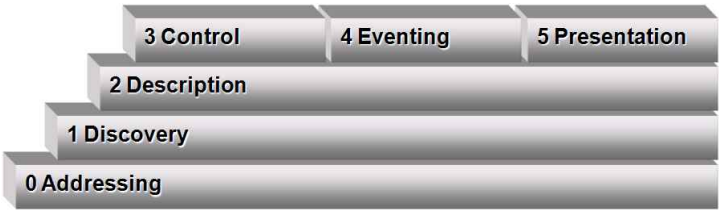

그림 1. UPnP Networking 단계 (stage) [5]

단계 0. Addressing : Addessing은 UPnP 네트워킹이 TCP/IP 스택을 가지고 통신을 하기 때문에 장치에는 반드시 IP 주소를 가지고 있어야 한다는 점에 존재해야 한다. "UPnP Device Architecture 1.0" 에서는 장치에 DHCP 클라이언트를 구현하여 IP 주소를 할당 받도록 명시를 하고 있다.

단계 1. Discovery Stage : 제어 장치가 홈 네트워크에 참여 했을 시에 피제어 장치를 찾거나, 피제어 장치가 홈 네트워크에 참여 했을 시에는 제어 장치에 홈 네트워크에 참여 했음을 알리는 단계 이다.

단계 2. Description Stage : Discovery 단계에서는 단순 장치에 대한 간략한 정보만을 주고 받게 된다. 그 단계로는 장치에 어떠한 서비스를 제공하는 능력이 있는지

알 수가 없다.

Description 단계에서는 해당 장치에 대한 어떤 서비스가 있으며, 서비스를 호출하기 위한 방법에 대한 정보를 주고 받는지에 대하여 기술한다.

단계 3. Control Stage : Description 단계로부터 취득한 정보를 기반으로 제어 장치가 피제어 장치를 조작 하는 단계 이다.

단계 4. Eventing Stage : Eventing 단계로부터 조작되어진 피제어 장치는 요청된 서비스 별로 사용 하고 있는 상태 값을 수정하게 된다.

단계 5. Presentation Stage : 이 단계는 장치 제작자의 구현에 따라 존재 할 수도 있고 없을 수도 있는 단계이다. 존재하는 URL를 통하여 장치 자신에 상태 값 및 유저 인터페이스를 사용자 혹은 또 다른 장치에 제공할 수 있는 단계 이다.

3. 장치의 잠금 방식 제안

기본적으로 UPnP에서는 장치 잠금에 대한 언급은 없다. 특정 사용자가 피제어 장치에 대한 독자적인 제어권을 갖기 위해 잠금 장치가 필요하다. 본 장에서는 피제어 장치에 대한 독자적인 제어권을 갖는 경우에 대한 시나리오와 제어권 획득 방식을 제안한다.

3.1 시나리오

피제어 장치인 디지털 액자는 한 개가 홈 네트워크에 참여 하고 있으며, 제어 장치는 사용자 A,B 의 제어 장치 역시 네트워크에 참여를 하고 있는 상황이다. 사용자 A는 자신이 즐겨 보는 사진 만을 설정하기 위해 B가 디지털 액자를 제어 하는 것을 방지 하고 싶다. 그러나 보통의 경우 B는 A의 목적과는 상관없이 디지털 액자를 제어할 수 있다. 따라서 디지털 액자에 대한 B의 제어권한을 박탈하고 A만이 디지털 액자를 조작 할 수 있는 상황을 만들 수 있는 방식이 필요 하다.

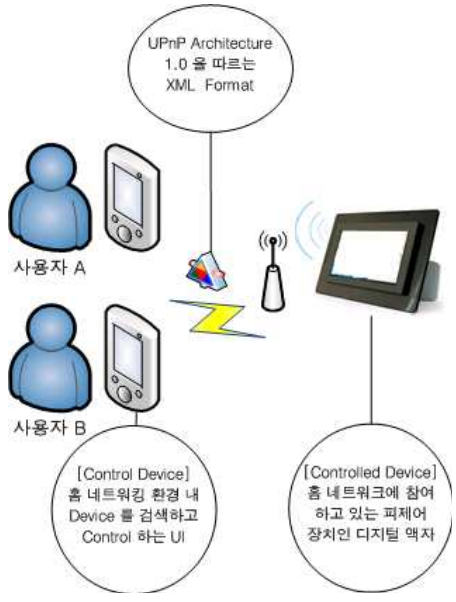

그림 2. 시나리오

3.2 제안하는 제어 방식

홈 네트워크 구성 방식은 두 가지로 나누어 질 수 있다. 프록시 즉 제어 장치에 의해 조작 되어지는 피제어 장치에 대한 제어 권한을 서버가 가지고 있는 경우와 제어권을 제어 디바이스마다 직접 가지고 있는 경우 가 있다.

3.2.1 프록시 서버가 존재 하는 경우

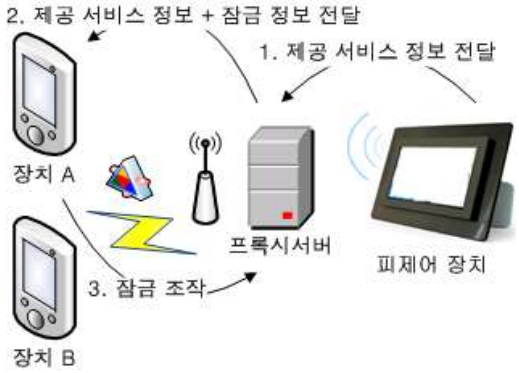

그림 3. 프록시 서버가 존재하는 경우

대형 아파트 혹은 건물이 새로 지어진 경우에는 프록시 서버를 사용하여 네트워크를 구성하는 경우가 많다.

이 경우 장치 제작자는 디바이스 잠금에 대한 신경을 쓰지 않아도 된다. 프록시 서버가 제어 명령을 전달하고 그 결과를 대신 통지하기 때문에 프록시 서버가 피제어 장치로 부터 제어 정보를 받는 경우 해당 정보에 잠금 정보를 포함하여 프록시 서버가 보유 하고 있으면 된다. 장치A는 결국 피제어 장치에는 구현 되어 있지 않은 잠금 서비스에 대한 정보를 소유하게 되며, 소유한 정보를 바탕으로 피제어 장치에 대한 잠금을 시도 할 수 있게 된다. 피제어 장치 또한 잠금에 대한 정보는 없지만 프록시 서버가 잠금 장치를 구현함으로써, 홈 네트워크에 참여 하는 장치 제작자에게 장치 수정이라는 부담 없이 잠금을 제공 할 수 있다.

3.2.2 프록시 서버 부재의 경우

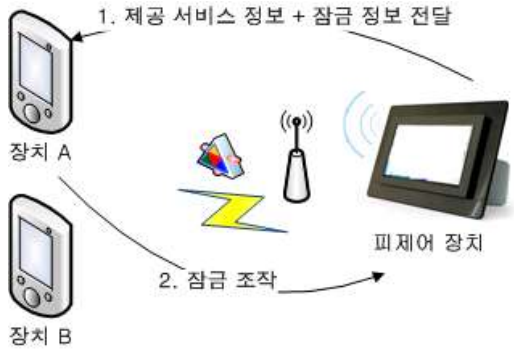

그림 4. 프록시 서버가 존재하지 않는 경우

프록시 서버가 존재하지 않는 상업적 솔루션은 XBox 360, Play Station3와 같은 가정용 게임기에서 많이 드러난다. 이 경우 제어 장치, 피 제어 장치 제공자는 잠금 정보에 대한 구현이 반드시 이루어 져야 한다.

그림 3의 제공 서비스 정보 전달 단계와 제공 서비스 정보+잠금 정보 전달 단계는 그림 4의 제공 서비스 정보+잠금 정보 전달 단계와 같다. 또한 그림 3의 잠금 조작 단계는 그림 5의 잠금 조작 단계와 같은 기능을 수행한다. 본 절에서는 각각의 단계에 대한 방법을 제시 하고자 한다.

```
<?xml version="1.0"?>
<root
xmlns="urn:schemas-upnp-org:device-1-0">
--- 중략 ---
<device>
<deviceType>urn:schemas-up...
<friendlyName>short user-friendly
title</friendlyName>
<manufacturer>manufacturer </Manufacturer>
<manufacturerURL>URL to manufacturer
site</manufacturerURL>
<Lock>
 <ip>잠금한 장치의 ip 주소</ip>
 <time>잠금 설정 시간</time>
</Lock>
--- 생략 ---
```

그림 5. 피제어 장치 제어 정보

그림 5는 제어 장치의 조작 정보를 전달하는 부분의 데이터 일부분이다 장치 제어 정보 부분에 Lock 태그를 삽입함으로써 피제어 장치가 장치 잠금 기능이 있다는 것을 알리며, 제어 장치의 IP주소와 해당 잠금 시간을 설정하여 장치 잠금 기능을 실행 할 수 있음을 알려준다.

```
<?xml version="1.0"?>
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soa...
s:encodingStyle="http://schemas.xmlsoap....
<s:Body>
<u:Lock
xmlns:u="urn:schemas-upnp-org:service:service
Type:v">
<ip>xxx.xxx.xxx.xxx</ip>
<time>xx</time>
other in args and their values go here, if any
</u:actionName>
</s:Body>
</s:Envelope>
--- 생략 ---
```

그림 6. 제어 장치 잠금 정보

그림 6은 그림 5로부터 받은 제어 정보를 이용하여 제어 장치가 피제어 장치를 잠금 할 때 보내는 데이터의 일부분을 나타내고 있다. 해당 디바이스에 대한 명령을 보낼 때 해당 서비스에 대해서 Lock 태그를 통해 설정을 하며 자신의 IP 주소를 ip 태그 안에 삽입을 하고 잠금 시간을 time 태그에 삽입을 하여 명령을 실행 한다.

4. 구현

UPnP에 잠금 기능을 구현시 사용한 라이브러리는 표1과 같으며, 제어 장치는 PDA , 피제어 장치는 window XP 환경에서 에뮬레이터를 만들어 테스트하였다.

[Communication Module]
Intel SDK WinSock1 & 2
[Data processing Module]
Intel Xml Parser
[Controlled & Control Processing Module]
ANSI C
[Device Emulator & User Interface Application]
MFC Programming & EVC Programming For Pocket Pc 2003

표 1 . 구현 시 사용 라이브러리

통신 모듈은 WinSock을 사용 하였으며, UPnP Description 정보 처리를 위하여 XML Parser를 사용하였다. 그리고 제어 장치와 피제어 장치에 대한 동작 과정을 기술한 모듈은 MFC 와 EVC Programming을 이용하여 구현하였으며, 이러한 모듈 중 제어 장치는 PDA에서 구동이 되며, 피제어 장치는 Windows XP기반의 에뮬레이터를 이용하였다.

그림 7은 구현의 결과물로 제어 장치에서 피제어 장치에 대한 잠금 기능이 추가 된 것을 보여 주고 있다.

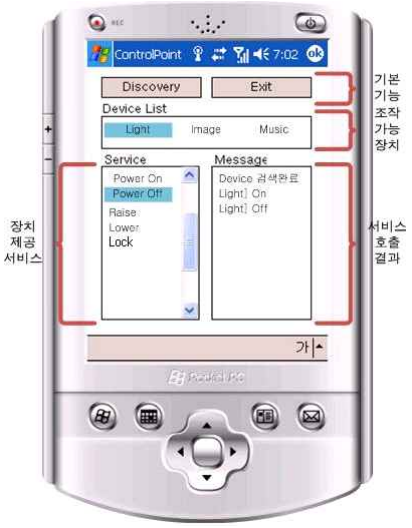

그림 8. 제어 인터페이스

5. 관련 연구

5.1 XACML[6]

XACML은 OASIS가 표준으로 정하고 있는 접근 제어를 위한 언어이다. XACML의 접근 제어를 위한 정책은 PolicySet, Policy, Rule의 계층구조로 정의되며, 가장 하위에 존재하는 Rule은 자원들에 대한 수행 가능한 Action과 그에 대하여 발생하는 효과 등을 기술한다.

정의된 Rule이나 Policy들은 미리 정해진 결합 알고리즘을 이용하여 서로 결합될 수 있으며, 어떠한 결합 알고리즘을 사용할 것인지 각 Rule에서 기술된다. 이러한 Rule과 Policy들의 결합은 접근 제어를 나타내기 쉽다는 장점이 있으나 결합 알고리즘을 통하여 새로운 Rule 또는 Policy를 계산해야 하므로 임베디드 장비와 같이 계산 능력이 적은 경우 이를 이용하는데 무리가 있다.

5.2 목적기반 접근제어[7]

기존 RBAC의 경우 “누가 무엇을 할 수 있다 또는 없다” 만을 기술하기 때문에 상황이나 목적에 따른 접근제어가 되지 않는다. 이에 비하여 목적기반 접근제어의 경우 접근 목적을 판단하여 해당 자원에 대한 접근을 제어하게 된다.

이를 효율적으로 적용하기 위하여 Rule 뿐만 아니라 목적과 객체도 Tree 형태로 관리를 하여 목적에 따른 접근을 제어하게 된다. 이러한 접근 제어의 경우 접근제어의 주체와 객체뿐만 아니라 목적에 대한 접근이 함께 가능하므로 좀 더 안전하게 접근제어를 할 수 있다. 그러나 이러한 접근 제어를 위하여 Rule, Purpose, Resource를 모두 Tree형태로 구조화 시켜 저장해야 하며, 접근 권한의 전파 또한 계산을 하여 접근 제어를 해야 하므로 컴퓨팅 파워가 한정된 장치에 이를 적용할 수가 없다.

6. 결론

홈 네트워크 분야에 UPnP 미들웨어를 탑재 하기란 그리 어려운 일은 아니다. 하지만 상업적인 제품에 있어서 미들웨어의 사용은 오버 헤드가 아닐 수 없다. 기존 제품에 TCP/IP 스택을 사용하기엔 프로세서의 컴퓨팅 파워가 높은 편은 아니다. 그래서 인지 최근엔 멀티미디어 플레이어 , 프린터 등 어느 정도 컴퓨팅 파워가 있는 제품에 UPnP 미들웨어가 탑재가 많이 되어 나오는 편이다. 그러나 이러한 UPnP는 자원의 독점과 같이 발생할 수 있는 여러 가지 문제점에 대한 해결책을 제시하지 않고 있다.

본 연구를 통해 제품들에 대하여 Device Description을 확장함으로써 독자적인 제어권을 갖기 위한 방법을 제시하였으며, 이로 인해 인증 단계가 없는 UPnP의 문제를 최소한의 오버헤드로 해결할 수 있다.

[참고 문헌]

[1] 손지연(Jiyeon Son·孫知延),박준석(Junseok Park·朴俊錫) 저, "Universal Plug and Play 기술 개요 및 동향", 한국정보기술학회, 한국정보기술학회지 韓國情報技術學會誌 제1권 제1호, 2003. 12, pp. 89 ~ 94 (6pages)

[2] 임승옥, 정광모, "UPnP(Universal Plug and Play) 기술 분석", 전자정보센터(EIC) 원고, 2003. 3.

[3] 정성원(Sungwon Jeong), 장영숙(Youngsuk Jang) 저, "UPnP 구조와 테스트 툴에 대한 고찰", 한국정보과학회, 한국정보과학회 학술발표논문집 한국정보과학회 2004년도 봄 학술발표논문집 제31권 제1호(B), 2004. 4, pp. 421 ~ 423 (3pages)

[4] 박동훈, 정우성, 김희천, 우치수 "UPnP 구조에서 규칙 기반 적응형 디바이스 컨트롤" 한국정보과학회 2007 한국컴퓨터종합학술대회 논문집 제34권 제1호(B), 2007

[5] UPnP™ Forum, "UPnP Device Architecture 1.0", http://www.upnp-ic.org/resources/UPnP_device_architecture_docs/UPnP-DeviceArchitecture-v1_0-20060720.pdf, 2006

[6] OASIS eXtensible Access Control Markup Language, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[7] Ji-Won Byun, Elisa Bertino, Ninghui Li, "Purpose Based Access Control of Complex Data for privacy Protection", SACMAT'05, 2005