

US Broker 에서 Jini 를 위한 Adaptor 의 설계¹

오일진*, 임형준*, 황윤영*, 이강찬**, 이승윤**, 이규철[†]

*충남대학교 컴퓨터공학과

**한국전자통신연구원

e-mail : victory25@cnu.ac.kr

Design of the Adaptor for Jini in US Broker

Il-Jin Oh*, Hyung-Jun Yim*, Yun-Young Hwang*,
Kangchan Lee**, Sengyun Lee**, Kyu-Chul Lee*

*Dept. of Computer Engineering, Chungnam National University

**Electronics and Telecommunications Research Institute

요 약

유비쿼터스 환경이란 사용자가 원하는 모든 종류의 서비스를 시간 및 장소에 구애받지 않고 사용가능하도록 지원하는 것을 뜻한다. 유비쿼터스 환경에서는 다양한 종류의 디바이스 및 서비스가 산재하며, 유동적인 특성을 가진다. 이러한 환경에서 사용자가 원하는 서비스를 발견하기 위해서 기존에는 서비스 디스커버리 미들웨어(서브네트워크)가 제공되었다. 하지만 이러한 서브네트워크는 상호운용을 지원하지 못한다. 상호 운용의 문제를 해결하기 위하여 OSGi, DomoNet, WSUN 등의 연구가 진행되었다. 이러한 연구 중에서 디바이스 및 서비스 통합을 위한 연구인 WSUN 는 US Broker 를 이용하여 사용자가 모든 서브네트워크의 서비스를 사용할 수 있도록 한다. US Broker 는 서브네트워크의 서비스를 웹 서비스화 하여 레지스트리에 저장 및 활용하며, 각각의 서브네트워크와 통신하기 위해 Adaptor 를 컴포넌트로 갖는다. 본 논문에서는 여러 서브네트워크 중 JINI 를 위한 Adaptor 에 대해 구체화하였다.

1. 서론

유비쿼터스 환경에서, 사용자는 어떠한 종류의 서비스든지 시간과 장소에 상관없이 존재하는 서비스를 사용할 수 있는 환경이다. 하지만 디바이스마다 다른 인터페이스와 다른 OS 의 사용으로 홈 네트워크의 구성에 어려움을 겪어왔다. 산재하는 서비스를 사용하기 위하여 JINI(Java Intelligent Network Infrastructure)[3][4], UPnP(Universal Plug and Play)[5] 등과 같은 서비스 디스커버리 미들웨어들이 존재한다. 본 논문에서는 서비스 디스커버리 미들웨어를 서브네트워크로 칭한다. 하지만 이러한 서브네트워크들은 상호 운용을 지원하지 못하고 있다. 이로 인하여 홈네트워크를 구성하는데 어려움을 겪게 되며, 사용자가 서브네트워크의 서비스를 사용하기 위해서는 별도의 구현을 필요로 하고, 다양한 통신 프로토콜을 지원해야 한다. 이는 사용자에게 많은 경제적 부담을 주며, 다양한 서비스를 사용할 수 없다는 제약을 주게 된다.

이러한 문제를 해결하기 위하여 서비스 통합을 위한 연구로 OSGi(Open Service Gateway Initiatives)와 DomoNet(Domotics Network)과 같은 연구들이 진행되었다. 그리고 서비스 및 디바이스 통합을 위하여

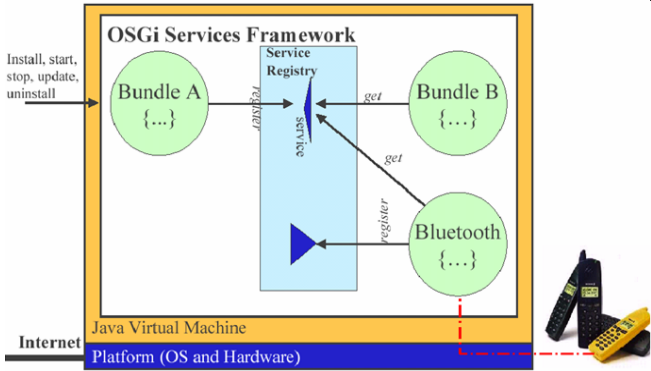
WSUN(Web Services on Universal Networks)이 진행되었다.

2 장은 서비스 통합을 위한 연구로 OSGi, DomoNet 을 설명하며, 3 장에서는 디바이스, 서비스 통합을 위한 WSUN 을 설명한다. 4 장에서는 WSUN 을 이루는 컴포넌트 중에서 JINI 와 관련된 Adaptor 에 대하여 설명하도록 한다. 5 장은 결론 및 향후 연구를 기술한다.

2. 관련연구

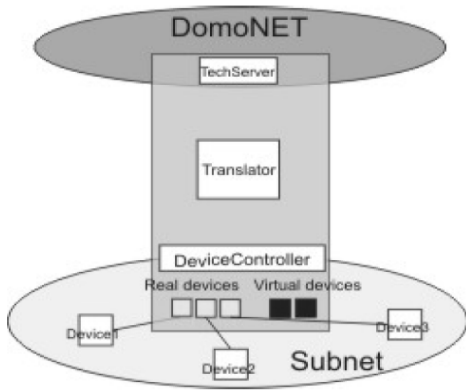
OSGi(Open Service Gateway initiative)는 서비스를 로컬 네트워크나 디바이스에 전달하고 전달된 서비스가 운용되는 개방적 표준을 제공한다. 또한 OSGi 는 서비스의 연결 및 제어, 서비스와 OSGi 프레임워크간의 연결 및 제어, OSGi 프레임워크와 외부 서비스관리 시스템과의 연결 및 제어를 담당한다. OSGi 는 그림 1 과 같이 OSGi 프레임워크와 실제 서비스 제공을 위한 번들로 구성된다. 번들은 서비스를 이용하기 위한 최소한의 단위로 서비스를 사용하기 위한 중간 매개체 역할을 한다. 서비스는 특정 기능을 수행하는 자바 인터페이스 또는 실제 구현 객체를 말한다.

¹ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성·지원 사업(IITA-2005-C1090-0502-0016)의 연구결과로 수행되었음. († 교신저자)



(그림 1) OSGi 번들의 구성

Domonet(Domotics Network)은 SOA 기반의 프레임워크로서 홈 네트워크 간의 상호 운용성을 해결하는데 그 목적이 있다[2].



(그림 2) TechManager 구조

Domonet 은 서브네트워크의 상호운용성을 위해 TechManager(TM)를 가진다. 또한 서브네트워크의 서비스를 웹 서비스화 (DeviceWS)하며, 서브네트워크별로 가상 디바이스를 생성한다. 가상 디바이스는 다른 서브네트워크의 디바이스가 해당 디바이스를 DomoNet 환경에서 하나의 통신 프로토콜로 여러 환경의 서브네트워크의 디바이스를 사용할 수 있게 하는 것이다. DomoNet 은 DomoNet Network 와 서브네트워크를 연결하기 위해서 그림 2 과 같은 TM 을 설계 하였다. TM 은 TechServer, DeviceController, Translator 로 구성되며 이들 각각의 역할은 다음과 같다.

- 1) TechServer: TCP 서버로서 DeviceWS 에 연결될 수 있고, DomoNet 의 다른 미들웨어에게 메시지를 보내기도 한다.
- 2) DeviceController: 다른 프로토콜을 사용하는 미들웨어의 디바이스를 다룬다.
- 3) Translator: DomoNet 에서 사용하는 프로토콜과 다른 미들웨어의 프로토콜 간의 중계역할을 담당한다.

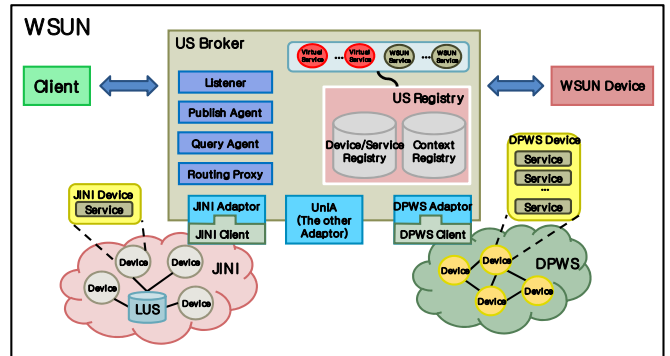
이러한 연구들은 서브네트워크의 서비스간의 통합을 목표로 한다. OSGi 를 통해 외부에서 홈 네트워크 상의 서비스를 컨트롤하고 사용할 수는 있지만, 아직까지는 홈 네트워크 상의 서비스간의 상호운용은 지

원하지 못한다.

DomoNet 은 OSGi 의 번들과 유사한 TM (TechManager)를 두어 서브네트워크간의 상호운용을 도모한다. 그러나, 유비쿼터스 환경의 특성인 유동적인 서비스의 상태 및 위치 변화를 반영하고 있지 못한 실정이다. 또한 DomoML 이라는 언어를 사용함으로써, 언어에 종속적이다.

3. Web Services on Universal Networks

WSUN 은 시스템에서 서비스를 검색하고 사용할 수 할 수 있는 범위를 나타낸다. 그림 3 과 같이 클라이언트, 서비스 제공자, US Broker 로 구성이 된다.[6]



(그림 3) WSUN 아키텍처

US Broker 는 Service Oriented Architectures (SOA)를 기반으로 한다. 클라이언트는 서비스를 사용할 때 US Broker 를 이용하여 유비쿼터스 웹서비스를 검색하고 사용하게 된다. 그리고 US Broker 는 다음과 같은 모듈을 중심으로 구성된다.

- 1) Query Agent: 서비스 검색을 위한 사용자 질의를 처리한다.
- 2) Publish Agent: US Registry 에 해당하는 메타데이터의 저장을 담당한다.
- 3) Routing Proxy: 사용자가 원하는 서비스의 해당 서브네트워크와 연결 담당한다.
- 4) Listener: 사용자와 WSUN 디바이스가 US Broker 와 통신하기 위한 US Broker 의 기본 정보 제공한다.
- 5) US Registry: Device/Service Registry 는 서비스의 동적인 서비스 검색을 위하여 메타데이터의 공통, 정적, 필수적인 요소를 저장하며, Context Registry 는 디바이스의 위치 정보를 저장한다.
- 6) Adaptor: 서브네트워크와 US Broker 사이의 중계 역할을 담당하며 서브네트워크의 서비스를 가상의 웹 서비스로 변환한다.

4. JINI Adaptor

Adaptor 는 각각의 서브네트워크를 위하여 존재하며, OSGi 의 번들이나 DomoNet 의 TM 과 비슷한 개념이다. 하지만 US Broker 에 존재하는 Adaptor 의 경우 각각의 서브네트워크에서 존재하는 클라이언트 및 서로 간에 관계를 구성하는 요소들 중에서 한 개의 요소를 확장한 것이 Adaptor 로 구성된다.

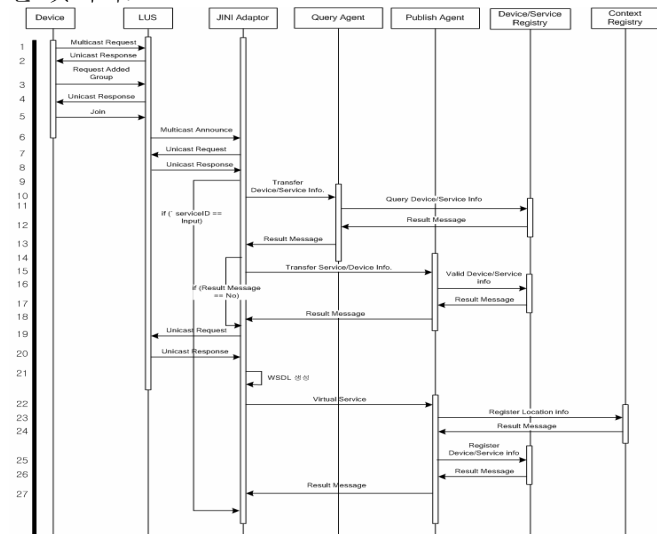
Adaptor 는 서브네트워크와 US Broker 에 필요한 정보를 전송하는 기능을 담당하며, 서브네트워크의 서

비스를 가상의 웹서비스로 변환을 하며, 서브네트워크로부터 레지스트리에 저장하기 위한 정보를 검색하는 기능을 담당한다.

JINI Adaptor 는 JINI 클라이언트에 Adaptor 의 기능을 추가 시킨 것이다. 또한 한 개의 서비스가 여러 개의 LUS 에 등록이 가능하기 때문에 클라이언트에 Reference Count 를 추가한다. Jini 시스템에서는 다음과 같은 이벤트 및 메시지가 발생 가능하며, 이벤트에 따른 JINI Adaptor 의 동작을 설명한다.

1) Lookup Service 에 Group 정보의 삽입

그림 4 는 LUS 에 존재하지 않는 서비스 그룹에 대하여 Grant 권한을 가진 디바이스가 서비스에 대한 그룹을 추가하고 등록됨으로써 일어나는 이벤트를 나타낸 것이다.



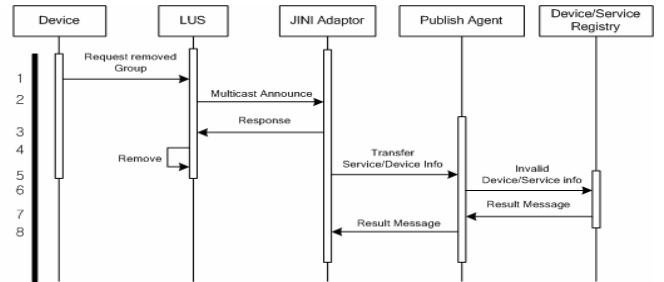
(그림 4) Group 정보의 삽입

- 1~8: 기존의 JINI 와 동일하게 그룹을 추가하고 그에 대한 정보를 클라이언트가 얻어오게 된다.
- 9: JINI Adaptor 에 서비스에 대한 정보를 검색하고, 서비스가 존재하는 경우, 현재 등록되어서 사용할 수 있는 서비스 임으로 Reference Count 를 1 을 증가시키고 처리를 마치게 된다. 하지만 존재하지 않는 경우, 다음의 처리를 진행하게 된다.
- 10~13: JINI Adaptor 에 존재하지 않는 경우, 그 서비스가 등장했다가 퇴장한 서비스인지, 처음 등장하는 서비스인가를 판단하기 위하여 Query Agent 를 통하여 서비스 검색을 하고 그 결과를 JINI Adaptor 에 알려주는 과정이다.
- 14: Query Agent 로부터 전달받은 값에 따라 레지스트리에 존재하는 경우, 현재 이 서비스는 한번 US Broker 에 등록이 되었던 서비스임으로 서비스의 상태가 Invalid 임을 뜻하게 된다. 이 경우 15~18 번 과정을 수행한다. 하지만 존재하지 않는 경우는, 그 서비스가 US Broker 에 처음 등장을 의미하는 것으로, 19 번부터 과정을 수행하게 된다.
- 15~18: Publish Agent 를 통하여 레지스트리에 등록된 서비스에 대하여 서비스 상태 정보를 Valid 로 변경하는 과정을 나타낸다.

- 19~21: 서비스에 대하여 WSDL 을 생성하기 위하여 추가적으로 필요한 정보를 요청하여 WSDL 을 생성하는 과정을 나타낸다.
- 22~27: 서비스에 대한 정보를 Publish Agent 를 통하여 Device/Service 레지스트리와 Context 레지스트리에 등록시키는 과정이다.

2) Lookup Service 에 Group 정보의 삭제

그림 5 는 LUS 에 등록되어 있는 그룹에 대하여 삭제하는 경우 일어나는 이벤트를 처리하는 과정을 나타낸다.

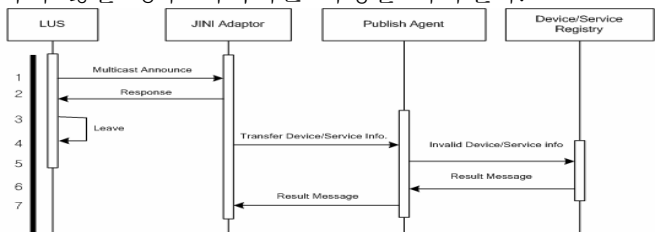


(그림 5) Group 정보의 삭제

- 1~4: Grant 권한을 갖는 디바이스에서 LUS 에 그룹 삭제 요청을 하게 되면 LUS 는 이벤트를 발생시키게 된다. 그리고 LUS 는 모든 디바이스로부터 응답이 도착하게 되면, 관련된 그룹정보를 삭제하게 된다.
- 5~8: 이 과정을 수행하는 경우는 한 개의 서비스가 한 개의 LUS 에 등록이 된 상황에서 이러한 이벤트가 발생하는 경우이다. 이 경우 JINI Adaptor 에 있는 Reference Count 가 0 이 되게 되며, 이는 더 이상 이 서비스를 사용할 수 없음을 의미한다.

3) Lookup Service 의 퇴장

그림 6 는 디바이스가 더 이상 LUS 기능을 수행하지 않을 경우 처리되는 과정을 나타낸다.



(그림 6) LUS 의 퇴장

- 1~3: 더 이상 LUS 기능을 하지 않을 경우, 디바이스에게 이벤트를 발생시키게 되고 모든 디바이스에게 응답이 오게 되면 LUS 기능을 정지하게 된다.
- 4~7: 하나의 서비스는 여러 개의 LUS 에 등록이 가능하기 때문에 서비스에 대하여 Reference Count 가 0 인 경우, 그 서비스에 대하여 레지스트리에 등록된 서비스 상태정보를 Invalid 로 변경하게 된다.

4) JINI Service 의 등장

JINI 서비스가 등장하는 경우, 서비스가 레지스트리에 등록되는 과정을 말한다. 이 경우, 기존의 JINI

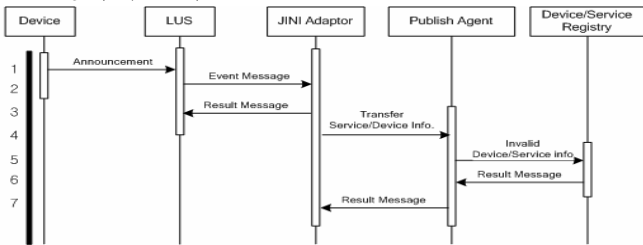
와 동일하게 동작하며, 추가적인 기능은 LUS 에 Group 정보 삽입과정과 동일하다.

5) JINI Service 의 퇴장

그림 7 는 디바이스가 서비스를 더 이상 제공하지 않을 경우, 이벤트를 처리하는 과정을 나타낸 것이다.

- 1~3: 디바이스는 LUS 에 더 이상 서비스를 제공하지 않음을 알리게 되고, LUS 는 클라이언트에 알리게 된다. 그리고 LUS 를 떠나는 것은, 서비스가 등록되어 있는 모든 LUS 로부터 떠나는 것을 의미하게 된다.

- 4~7: JINI 서비스 퇴장과 관련된 이벤트가 발생하게 되면 여러 LUS 중에 첫 번째 도착한 메시지에 대하여 이벤트를 처리하게 되고, 차후에 도착한 메시지에 대해서는 Reference Count 를 감소시키는 작업을 진행하게 된다.

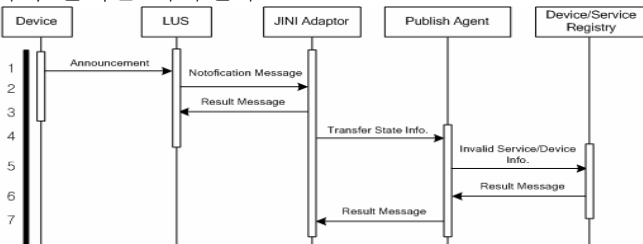


(그림 7) 서비스의 퇴장

6) 서비스 상태의 변경

WSUN 은 서비스의 현재 사용가능여부에 따라서 사용자에게 서비스를 제공하게 된다. 서비스 상태의 변경에서 Invalid 는 서비스 제공자가 참여를 하여야 하는 문제가 발생했을 경우를 말한다

그림 8 은 서비스 상태 정보 변경에 따른 이벤트 처리 순서를 나타낸다.



(그림 8) 상태정보 변경

- 1~3: 서비스에 대한 상태정보가 변경되게 되면 LUS 를 통하여 관련된 클라이언트에게 메시지를 보내게 된다.

- 4~7: 서비스를 사용하기 위하여 서비스 제공자의 도움이 필요한 문제와 관련된 경우 서비스에 대한 상태 정보를 Invalid 로 변경한다.

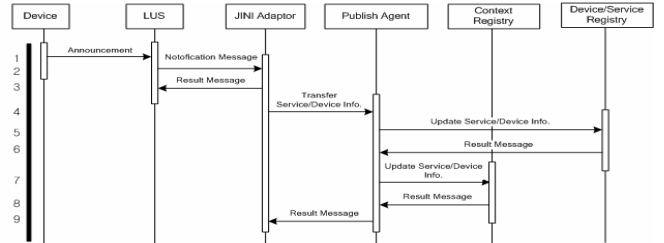
7) Attribute Set 의 변경

그림 9 은 서비스에 대한 기술 정보가 변경되는 경우에 대한 처리과정을 나타낸 것이다. JINI 시스템은 서비스에 대하여 클래스로 세분화하여 기술하고 있다. 그 결과 서비스의 기술정보가 변경이 되게 되면 WSUN 의 레지스트리에 변경에 따른 서비스 정보

를 변경하기 위한 과정이다.

- 1~3: 서비스에 대한 기술정보가 변경되면 LUS 에 알리게 되고 LUS 는 클라이언트에 이벤트를 전달하게 된다.

- 4~9: 클라이언트로 전달된 이벤트에 대하여 서비스에 대한 기술정보의 변경의 경우, 변경된 정보에 대하여 디바이스/서비스 레지스트리와 컨텍스트 레지스트리에 각각 저장된 정보를 변경한다.



(그림 9) Attribute Set 의 변경

5. 결론 및 향후 연구

지금까지 WSUN 에서 JINI 시스템과 관련하여 서비스를 US 레지스트리에 등록하고, 서비스와 관련하여 이벤트 처리 방식에 대하여 설명하였다. 이로 인하여 JVM 이 설치 되지 않은 클라이언트도 JINI 시스템에서 제공하는 서비스를 사용할 수 있게 되었다. 현재 시스템은 웹서비스를 사용하는 사용자 디바이스가 유비쿼터스 웹서비스를 사용하도록 구성이 된다. 하지만, 완전한 서비스 및 디바이스 통합을 위해서는 US Broker 를 통한 서브네트워크간의 서비스 검색 및 사용이 가능해야한다. 이를 위하여 향후에 하나의 서브네트워크의 클라이언트가 US Broker 를 통하여 다른 서브네트워크의 서비스를 사용할 수 있도록 연구를 진행할 것이다.

참고문헌

- [1] OSGi Site, <http://www.osgi.org>
- [2] Vittorio Miori, Luca Tarrini, Maurizio Manca, Gabriele Tolomei, "DomoNet: a framework and a prototype for interoperability of domotic middlewares based on XML and Web Services", IEEE, 2006
- [3] JINI Specifications Archive – v2.1, http://java.sun.com/products/jini/2_1_index.html
- [4] UPnP Forum, <http://www.upnp.org>
- [5] 오일진, 임형준, 황윤영, 이강찬, 이승윤, 이규철, "유비쿼터스 웹서비스의 동적 디스커버리를 위한 브로커 설계", 한국전자거래학회 2007 년도 춘계 워크숍, pp.165-174, 한국전자거래학회
- [6] Yun-Young Hwang, Il-Jin Oh, Hyung-Jun Im, Kang-Chan Lee, Seong-Yun Lee, Kyu-Chul Lee, "UWS Broker for Ubiquitous Web Services Dynamic Discovery", IPC 2007,