

# 분산 OSGi 프레임워크들로 구성된 홈 네트워크에서 P2P을 이용한 RSP 번들 설계 및 구현

윤기현<sup>o</sup> 김은희 최재영

송실대학교 컴퓨터학부

{khyun, ehkim} @ss.ssu.ac.kr, choi @ssu.ac.kr

## A Design and Implementation of P2P based RSP bundle for Home Network of Distributed OSGi Frameworks

Kihyun Yun<sup>o</sup>, Eunhoe Kim, Jaeyoung Choi

School of Computing, Soongsil University

### 요 약

현재 OSGi 프레임워크의 서비스는 로컬 장비에서만 공유할 수 있기 때문에 하나의 장비에 서비스 부하가 집중되는 문제점을 가진다. 이와 같은 문제점을 해결하기 위해 확장 서비스 레지스트리를 이용하는 방법이 제안되었지만, 확장 서비스 레지스트리를 재 검색해야 하므로 응답시간이 늦어지는 단점을 가진다. 따라서 본 논문에서는 추가적인 확장 서비스 레지스트리를 재 검색할 필요가 없고, 단지 로컬 서비스 레지스트리만을 검색하여 원격 서비스를 실행할 수 있는 P2P 기반의 RSP 번들을 설계하고 아기 보호를 위한 유비쿼터스 홈 시스템에 실제로 적용하여 구현하였다. 그러므로 본 논문에서 제안한 RSP 번들을 이용하면 하나의 OSGi 프레임워크에 집중될 수 있는 부하를 분산시킬 수 있고, 기존에 개발된 서비스들도 원격 서비스로 재정의하여 다른 OSGi 프레임워크와 서비스를 공유할 수 있게 된다.

### 1. 서론

전자, 정보 산업의 놀라운 발전으로 인해 유무선 홈 네트워크와 인터넷을 연동하여 가정내의 PC를 비롯한 정보 가진 기기를 언제 어디서나 자유롭게 제어하거나 사용하기 위한 요구가 높아지고 있다. 따라서 가정내의 모든 기기들을 하나의 정보시스템에 통합하여 제어 할 수 있는 다양한 솔루션 연구와 UPnP, Jini, HAVi 등과 같은 미들웨어 기술들이 연구되어 왔다. 하지만 서로 다른 미들웨어간에 상호 운용이 어렵고 새로운 서비스의 배포가 어렵다는 문제가 발생하였다. OSGi(Open Service Gateway initiative)는 이러한 문제점을 해결하기 위한 기술 표준으로 주목 받는 기술이다.

OSGi 기술의 핵심인 OSGi 프레임워크는 동적인 컴포넌트들의 변화에 따른 관리를 최소화하기 위해서 서로를 동적으로 발견하는 것을 가능하게 한다. 이러한 동적으로 변화하는 서비스들간의 협력을 위해 OSGi 프레임워크는 서비스 레지스트리를 제공한다.

현재 홈 네트워크에서는 OSGi 프레임워크의 서비스 레지스트리를 이용하여 가정 내의 수 많은 장치들이 제공하는 다양한 서비스들을 관리하고 배포하고 있다. 그러나 현재 OSGi 프레임워크의 서비스 레지스트리는 로컬 장비에서만 공유할 수 있다는 문제점을 가진다. 따라서 하나의 OSGi 프레임워크가 작동하는 장비가

중앙 서버의 역할을 하게 된다. 이 때, 홈 네트워크를 구성하는 모든 장치는 서비스를 제공받기 위해 하나의 서버 장비에 집중될 수 밖에 없다.

이와 같은 문제점을 해결하기 위해 OSGi 프레임워크를 분산 시키고, 분산된 OSGi 프레임워크간의 서비스를 공유하는 방법이 필요하다. 분산된 OSGi 프레임워크간의 서비스를 공유하기 위해 RMI 레지스트리와 웹 서비스의 UDDI을 이용한 방법이 제안되었다. 그러나 이러한 기존의 솔루션들은 원격의 서비스를 호출하기 위해 확장 서비스 레지스트리를 재 검색해야 하고 RMI나 웹 서비스의 기술을 OSGi와 연동시켜야 하므로 프로그램 과정이 복잡하다는 단점이 있다.

따라서 본 논문에서는 효율적으로 원격 OSGi 프레임워크간의 서비스를 공유하기 위한 방법으로 P2P 기술을 이용한 RSP(Remote Service Provider) 번들을 설계하고 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 설명하고 3장에서는 본 논문에서 제안하는 P2P 기반 RSP 번들을 설명한다. 4장에서는 3장에서 설계한 P2P기반의 RSP번들을 실제 분산된 OSGi 프레임워크들로 구성된 홈 네트워크에 적용하여, 아기 보호를 지원하는 아기 보호 u-시스템을 설계한다. 5장에서는 구현 및 실험결과를 논하고 6장에서는 결론과 향후 연구를 논한다.

## 2. 관련 연구

### 2.1 OSGi의 서비스 레지스트리

OSGi 기술의 핵심인 OSGi 프레임워크에서는 JVM(Java Virtual Machine)에서 제공하는 클래스 공유를 사용하여 번들이 서비스를 공유하여 협력한다. OSGi에서 번들이란 하나의 어플리케이션 단위로 0개 이상의 서비스를 가지고 있는 서비스들의 집합이다. 그러나 동적으로 설치(installing) 또는 삭제(uninstalling)되는 번들에 대해서는 일반적인 클래스 공유로 서비스들을 공유할 수 없기 때문에 상호 협력 관계를 유지할 수 없다. 따라서 OSGi 프레임워크는 동적인 번들간의 서비스들을 공유하기 위해 번들간의 협력 모델을 제공하고 서비스 의존성을 관리하기 위해서 서비스 레지스트리(service registry)를 제공하고 있다 [1].

하지만 OSGi 스펙에 정의된 서비스 레지스트리는 로컬 시스템에서만 검색이 가능하다는 한계점을 가진다. 즉, 하나의 JVM에 동작하는 OSGi 프레임워크의 로컬 서비스 레지스트리를 다른 JVM에서 동작하는 OSGi 프레임워크에서 검색 할 수 없다.

그런데 다양한 장비에 OSGi 프레임워크가 별도로 존재 하는 유비쿼터스 환경의 홈 네트워크 같은 시스템에서 사용자에게 다양하고 적절한 서비스를 제공하기 위해서는 서로 다른 OSGi 프레임워크들이 제공하는 서비스들을 공유해야 한다. 이를 위하여 RMI와 웹 서비스를 이용한 확장 서비스 레지스트리 방법이 제안되었다 [3,4].

### 2.2 RMI 레지스트리를 이용한 확장 서비스 레지스트리

RMI 레지스트리를 이용한 확장 서비스 레지스트리는 OSGi 프레임워크의 서비스 레지스트리와 RMI의 서비스 레지스트리를 이용한다. 번들이 서비스를 요청하면, 먼저 번들이 설치되어 있는 OSGi 프레임워크의 로컬 서비스 레지스트리를 검색한다. 요청한 서비스가 로컬 서비스 레지스트리에 존재하지 않을 때, 확장 서비스 레지스트리(XSR) 서비스가 원격에 있는 OSGi 프레임워크의 RMI 서비스 레지스트리를 검색하게 된다. 원격 서비스를 검색하기 위해 RMI 을 사용하고 있다. 만약 요청 서비스가 있는 RMI 레지스트리를 찾으면, 해당 OSGi 프레임워크의 서비스 레지스트리에서 요청한 서비스의 RMI 클라이언트 인스턴스를 리턴 받는다 [3].

RMI 레지스트리를 이용한 서비스가 로컬 서비스 레지스트리에 없을 경우 RMI 레지스트리를 포함하고 있는 모든 OSGi 프레임워크들을 검색해야 하는 문제가 발생 한다. 또 원격 서비스 제공을 위해 시스템을 구성하는 모든 장비들이 RMI 기술을 이용하여 제작한

서버/클라이언트 오브젝트들을 포함하고 있어야 한다는 단점을 가진다.

### 2.3 Web Service의 UDDI을 이용한 확장 서비스 레지스트리

UDDI을 이용한 확장 서비스 레지스트리 방법은 기존의 OSGi 서비스 레지스트리와 웹 서비스 기술의 UDDI(Universal Description, Discovery and Integration) 레지스트리를 이용하여 서로 다른 OSGi 프레임워크상의 서비스와 이기종 시스템의 서비스들을 공유할 수 있게 한다.

번들이 서비스를 요청하면, 먼저 요청한 서비스를 로컬 서비스 레지스트리에서 검색한다. 요청한 서비스가 로컬 서비스 레지스트리에 없을 때, 확장 서비스 레지스트리 서비스가 웹을 통해 UDDI 레지스트리를 검색하여 요청한 서비스의 위치를 찾는다. 서비스를 요청한 OSGi 프레임워크는 서비스의 위치를 리턴 받은 후 해당 서비스를 바인딩하여 서비스를 수행하게 된다 [4].

이 과정에서 OSGi 프레임워크는 SOAP(Simple Object Access Protocol) 기반의 웹 서비스를 사용하게 된다. 그런데 SOAP 기반의 웹 서비스는 XML 문서를 파싱하기 위한 오버헤드를 발생하기 때문에 전체적인 서비스 응답시간이 느리다. 또 UDDI을 검색하여 서비스를 검색하는데 모든 서비스를 통합 관리하는 UDDI가 존재 해야 한다는 단점을 가진다.

RMI와 UDDI을 이용한 확장 레지스트리 방법은 기존 OSGi 프레임워크의 서비스 레지스트리뿐만 아니라 추가적으로 원격 서비스 검색을 위한 RMI 레지스트리나 웹 서비스의 UDDI가 필요하다. 따라서 번들이 서비스를 요청할 때, 기존 OSGi 프레임워크의 서비스 레지스트리를 검색 후, 서비스가 없으면 확장 서비스 레지스트리를 다시 한번 검색해야 하기 때문에 이로 인해서 서비스 요청에 대한 응답 시간이 느려진다는 공통적인 단점을 가진다.

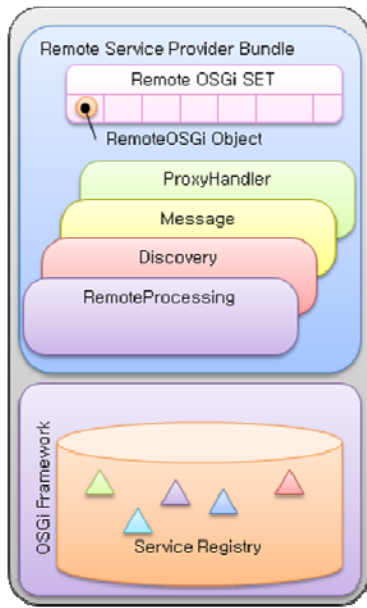
따라서 본 논문에서는 추가적인 확장 서비스 레지스트리를 사용할 때 발생하는 오버헤드를 없애기 위해 P2P 기반의 RSP(Remote Service Provider) 번들을 제안한다. P2P 기반의 RSP 번들은 원격 서비스를 검색하기 위해 별도의 기술을 사용하는 확장 서비스 레지스트리가 필요 하지 않고, 기존의 OSGi 프레임워크의 로컬 서비스 레지스트리만을 사용한다. 또한 서비스를 하나의 Peer로 구성하고, Peer-to-Peer의 구조를 가지기 때문에 클라이언트-서버 구조를 가지는 RMI나 UDDI을 이용한 확장 서비스 레지스트리 방법의 단점을 극복 할 수 있다.

## 3. P2P기반 RSP(Remote Service Provider) 번들 설계

### 3.1 RSP (Remote Service Provider) 번들 구조

RSP 번들은 다양한 서비스들을 효율적으로 사용자에게 제공할 수 있도록 서로 다른 JVM에서 동작중인 OSGi 프레임워크에 존재하는 서비스들을 공유할 수 있게 해주는 번들이다. 서로 다른 OSGi 프레임워크에 존재하는 서비스들을 공유하기 위해 멀티캐스트를 이용하고 공유된 원격 서비스를 Proxy와 소켓통신으로 호출해 서비스를 수행한다.

RSP 번들의 핵심 요소는 Discovery, ProxyHandler, RemoteOSGi, RemoteProcessing 이다.



[그림 1] OSGi 프레임워크의 서비스 레지스트리와 RSP 번들 구조

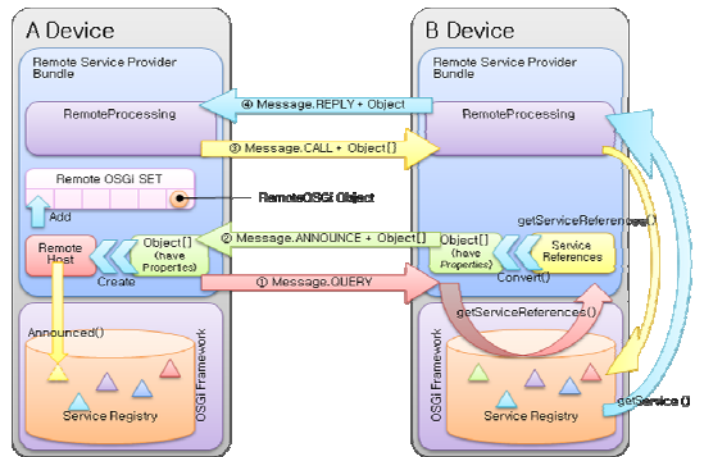
Discovery는 멀티캐스트를 이용하여 서로 다른 OSGi 프레임워크간의 서비스를 공유할 수 있게 한다. 이때, OSGi 프레임워크간 사용하는 메시지 타입은 질의(QUERY), 공표(ANNOUNCE) 타입을 가진다. ProxyHandler는 공유된 원격 서비스의 Proxy 인스턴스를 생성하고 생성된 Proxy 인스턴스로 공유를 원하는 OSGi 프레임워크의 로컬 서비스 레지스트리에 등록이 된다. ProxyHandler는 Proxy 인스턴스의 호출 핸들러를 가지고 있다. 즉, Proxy 인스턴스를 호출하면 실제로는 invoke 메소드에 기술된 코드를 실행하여 실제 서비스가 있는 다른 OSGi 프레임워크에게 서비스를 실행 요청한다 [5]. OSGi 프레임워크가 다른 OSGi 프레임워크에게 서비스 실행 요청을 할 때 사용하는 메시지는 호출(CALL) 타입이다. RemoteOSGi는 다른 OSGi 프레임워크와 1:1관계로 생성되며 OSGi 프레임워크의 IP 주소와 Port를 포함하고 원격 서비스로 제공하는 서비스 목록을 가지고 있다. 이 서비스 목록은 제공하던 원격 서비스가 삭제될 때, 등록된 서비스 목록을 갱신하는데 도움을 준다. RemoteProcessing은 소켓 통신으로

서비스를 제공하는 OSGi 프레임워크가 다른 OSGi 프레임워크로부터 호출(CALL) 메시지를 받은 후, 실제 서비스를 실행하고 실행결과를 응답(REPLY) 메시지에 포함하여 서비스를 호출한 OSGi 프레임워크에 보낸다.

원격 OSGi 프레임워크간 서비스를 공유하기 위해서 RSP 번들은 멀티캐스트와 Proxy를 이용하여 서로 다른 OSGi의 서비스를 공유한다. 그런데 OSGi 프레임워크의 서비스 레지스트리에 수 많은 서비스들이 등록되어 있을 수 있기 때문에, 모든 서비스를 공유하는 것은 비효율적이다. 따라서 본 논문에서는 OSGi R4.1에서 제공하는 Filter를 이용하여, 공유할 필요가 있는 서비스들만 별도의 정의로 공유한다. Filter는 OSGi 프레임워크의 로컬 서비스 레지스트리에 등록할 때 Properties 값을 갖는데, 이 Properties를 이용하여 서비스 레지스트리에서 원하는 서비스를 찾는 것이 가능하다. RSP 번들에 의해서 공유되는 서비스는 "(ServiceType=remote)"라는 Properties 값을 포함한다.

### 3.2 RSP 번들을 통한 OSGi 프레임워크간 서비스 공유

RSP 번들이 원격 서비스를 공유하여 수행하는 과정은 [그림 2]와 같다.



[그림 2] RSP 번들을 통한 OSGi 프레임워크간의 서비스 공유 및 수행 과정

RSP 번들이 구동되는 서로 다른 OSGi 프레임워크가 동작중인 A와 B 장치가 있을 때, 장치 A의 OSGi 프레임워크(이하 A)에서 다른 장치 B의 OSGi 프레임워크(이하 B)에 있는 원격 서비스를 공유하는 과정은 다음과 같다. 먼저, A의 RSP 번들은 다른 OSGi 프레임워크의 서비스를 알기 위해 A의 Discovery는 QUERY 메시지를 멀티캐스트로 다른 OSGi 프레임워크에서 동작중인 RSP 번들에게 보낸다(①). 본 논문에서는 설명을 돕기 위해 RSP 번들이 구동되고 있는 OSGi 프레임워크가 멀티캐스트를 통해 QUERY 메시지를 받은 것을 B라고 하겠다. QUERY 메시지를

받은 B의 Discovery는 B의 로컬 서비스 레지스트리를 검색한다. OSGi에서 제공하는 Filter를 이용하여 공유하고자 하는 서비스를 찾은 후, 서비스의 Properties를 오브젝트로 변환시켜 ANNOUNCE 메시지에 포함시킨다. ANNOUNCE 메시지는 멀티캐스트를 통해 A로 보낸다(2). ANNOUNCE 메시지를 받은 A의 Discovery는 메시지를 분석하여 B가 제공하는 원격 서비스를 A의 서비스 레지스트리에 등록한다. A의 서비스 레지스트리에 등록되는 서비스 인스턴스는 ProxyHandler를 이용해 생성된 Proxy 인스턴스이다. 동시에 B에 대한 정보를 포함하는 RemoteOSGi를 생성하여 A의 RSP 번들 내부에 가지고 있다. 이로써 다른 OSGi 프레임워크의 서비스를 공유할 수 있다.

이와 같이 공유된 원격 서비스의 실행과정은 OSGi 프레임워크의 로컬 서비스 레지스트리를 검색하여 발견한 로컬 서비스를 수행하는 방법은 같다. A에서 서비스를 요청하면 서비스를 찾기 위해 A의 로컬 서비스 레지스트리를 확인한다. 요청한 서비스가 있다면 서비스 인스턴스를 획득하여 해당 서비스를 수행한다. 하지만 등록된 서비스가 Proxy 인스턴스이면, 내부적으로는 다음과 같은 과정을 통해서 서비스를 수행한다.

원격 서비스를 제공하는 장치가 B일 때, B 장치는 RSP 번들의 RemoteProcessing에 의해 ServerSocket이 생성되고 대기상태가 된다. 만약 B 장치의 원격 서비스를 제공 받으려고 하는 장치가 A이라면, A는 로컬 서비스 레지스트리를 검색하여 서비스로 등록된 Proxy 인스턴스를 획득한다. 획득한 Proxy 인스턴스의 invoke 메소드에 의해서 실제 서비스를 가지고 있는 OSGi 프레임워크에게 해당 서비스 실행요청 CALL 메시지를 소켓통신을 이용하여 보낸다(3). OSGi 프레임워크의 위치는 A의 RSP 번들 내부의 RemoteOSGi를 이용하여 알아 낼 수 있다. A로부터 CALL 메시지를 받은 B의 RemoteProcessing은 메시지를 분석한 후 해당 서비스를 찾아 서비스를 실행한다. 실행 결과는 소켓 통신으로 REPLY 메시지를 통해서 리턴받는다(4).

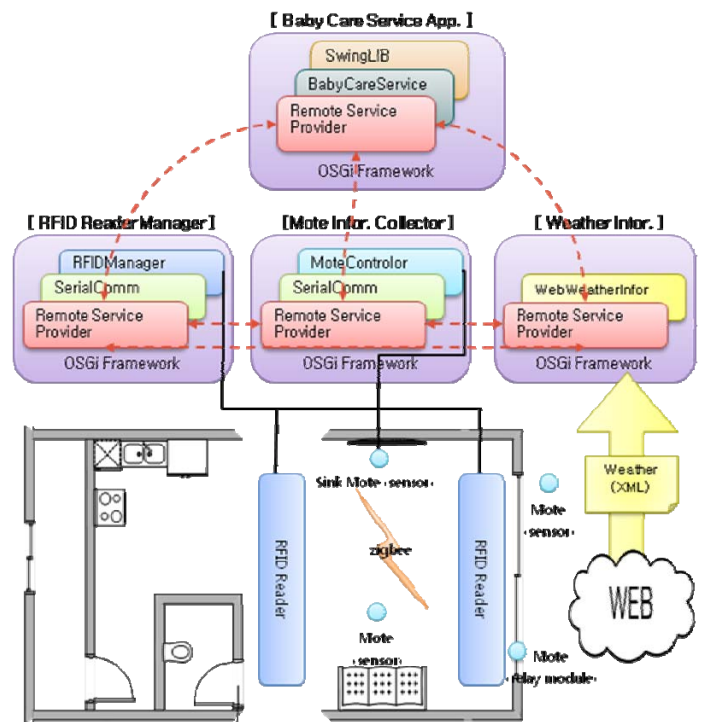
이와 같이 멀티캐스트를 제공하는 P2P기반의 Discovery, Proxy 인스턴스의 invoke 메소드를 정의하기 위한 ProxyHandler, 다른 OSGi 프레임워크의 정보를 가지고 있는 RemoteOSGi, 소켓통신을 제공하는 RemoteProcessing을 이용하여 서로 다른 OSGi 프레임워크간 서비스를 공유할 수 있게 된다.

#### 4. 아기 보호 u-시스템 (Baby Care u-System)

본 절에서는 3절에서 설계한 P2P 기반의 RSP 번들을 실제 분산된 OSGi 프레임워크들로 구성된 홈 네트워크 상의 아기 보호를 위한 유비쿼터스 홈 시스템(이하

아기 보호 u-시스템)에 적용하여 RSP 번들의 실제 응용에 대해서 논한다.

아기 보호를 위한 아기 보호 u-시스템은 온도와 습도에 민감하고 위험에 노출되기 쉬운 아기들을 위해 쾌적한 환경 서비스와 안전 서비스를 제공하는 시스템이다. 예를 들면, 외부의 날씨 상태를 체크하여 쾌적한 날씨라면 내부의 공기를 순환시키기 위해서 창문을 열어서 환기 시켜주는 서비스와 열려있는 창문은 아기에게 위험 요소이므로 아기가 창문에 접근하면 창문을 닫아주고 아기의 시선을 다른 방향으로 유도해주는 서비스를 제공한다. u-아기 보호 시스템의 전체 구조는 [그림 3]과 같다.



[그림 3] 아기 보호 u-시스템을 구현하기 위한 각 컴포넌트와 컴포넌트의 구성 번들

아기 보호 u-시스템에서 <Baby Care Service App> 컴포넌트는 아기에게 쾌적한 환경을 제공하고 위험요소로부터 보호할 수 있는 서비스를 제공한다. 서비스를 제공하기 위해서는 외부의 날씨 정보, 센서를 통한 실내·외의 온도·습도와 RFID Reader를 통한 아기의 위치 정보를 모두 받아 들여야 한다. 그러므로 각 정보를 제공하는 3개의 컴포넌트가 필요하다. 그 중에 하나는 <Weather Info> 컴포넌트로 웹에서 날씨 정보를 XML문서로 받아서 현재 날씨 정보를 제공한다. 또 하나는 <Mote Infor Collector> 컴포넌트로 실내·외에 위치한 센서들로부터 온도·습도·조도의 정보를 취합하여 컨텍스트 정보를 제공한다. 마지막으로 <RFID Reader Manager> 컴포넌트로

거실과 창문에 설치되어 있는 RFID 리더기를 이용하여 아기의 위치를 실시간으로 제공한다.

각 컴포넌트들은 기능을 수행하기 위해 다음과 같은 다양한 번들들을 포함한다. 먼저 <Baby Care Service App> 컴포넌트는 수집되는 컨텍스트 정보를 보여주기 위한 GUI를 제공하는 SwingLIB 번들들을 포함한다. 또한 수집된 정보를 취합하여 상황에 맞는 서비스를 제공하는 BabyCareService 번들이 있다. <Weather Info> 컴포넌트는 웹을 통해 날씨 정보를 XML문서로 받아 파싱한다. 파싱된 정보를 분석하여 현재 날씨 정보를 제공하는 WebWeatherInfo 번들들을 포함하고 있다. WebWeatherInfo번들의 getCurrentWeather 서비스는 현재 날씨 정보를 획득할 때 사용되는 서비스이다. getCurrentWeather 서비스는 원격으로 서비스를 제공하기 위해 서비스의 Properties 값으로 "(ServiceType=remote)"을 가진다. <Mote Info Collector> 컴포넌트는 시리얼통신 라이브러리와 시리얼 통신으로 주고 받는 메시지를 정의한 SerialComm 번들들을 이용한다. SerialComm번들은 싱크모트와 시리얼 통신을 할 수 있게 하는 번들이다. 센서모트로부터 수집되는 온도·습도·조도와 같은 컨텍스트 정보는 MoteControlor 번들들을 이용하여 저장한다. MoteControlor 번들의 getSensingData 서비스는 저장된 정보를 시간단위로 추출할 때 사용하는 서비스이다. getSensingData 서비스는 원격 서비스를 위하여 서비스를 등록할 때 Properties 값으로 "(ServiceType=remote)"을 가진다. 마지막으로 <RFID Reader Manager> 컴포넌트는 여러 개의 RFID 리더기와 시리얼로 통신하여 아기와 엄마의 위치 정보를 제공한다. 시리얼 통신을 위해 SerialComm 번들들을 이용한다. RFIDManager번들에서 RFID 리더기로부터 획득되는 정보를 저장하고 RFID 리더기가 설치된 위치에 누가 존재하는 지를 알고자 할 때 호출되는 getLocationPerson 서비스를 등록할 때 Properties 값으로 "(ServiceType=remote)"을 설정 한다. 분산되어 있는 OSGi 프레임워크기반으로 작동하는 컴포넌트들이 제공하는 서비스를 공유하기 위해 각 컴포넌트들은 P2P기반의 RSP 번들들을 모두 포함하고 있다. RSP 번들들을 사용하여 원격 OSGi 프레임워크의 서비스를 찾기 위해 멀티캐스트 통신을 사용한다. OSGi의 Filter 기능으로 서비스의 Properties 값이 "(ServiceType=remote)" 인 서비스만을 원격 서비스로 공유하고, Proxy와 소켓통신을 이용하여 원격 OSGi 프레임워크에 있는 서비스를 실행하고 실행결과값을 받는다.

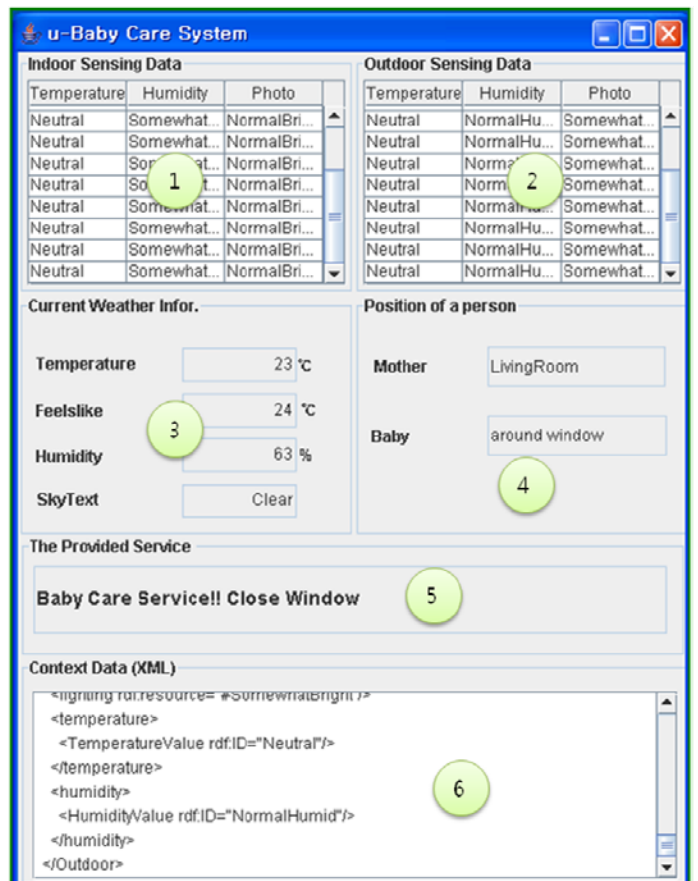
5. 구현 및 실행 결과

분산된 OSGi 프레임워크 환경을 구축하기 위해 VMWare를 이용한다. 각 VMWare의 운영체제는

WindowXP를 사용하였으며, JRE 1.5에서 OSGi 프레임워크를 실행했다. OSGi 프레임워크는 Knopflerfish에서 제공하는 프레임워크로 2.0.1 버전을 사용하였다. 하드웨어 배치는 [그림 3]과 같이 RFID리더기와 센서 모트들을 배치한다. RFID리더기는 거실과 창문에 설치되어 각 위치에 누가 존재하는지를 알기 위한 목적으로 이용한다.

센서 모트는 TinyOS와 NesC을 이용하여 만든 프로그램으로 동작하며, 총 4개를 사용한다. 하나는 싱크모트(Sink Mote)로 다른 센서가 센싱한 정보를 수집하는 역할과 센서를 제어하기 위해 <Mote Info Collector> 컴포넌트와 시리얼통신으로 연결된다. 다른 두 개의 센서 모트는 실내·외에 위치하여 온도·습도·조도를 측정하여 싱크모트로 센싱된 데이터를 제공한다. 마지막 하나는 릴레이 모듈과 연결되어 싱크모트의 제어 신호에 따라서 창문을 작동시킨다.

RFID 태그는 사용자에게 하나씩 제공되고 신원확인을 위해 사용한다. 시스템 구현을 위해 사용한 태그는 13.56MHz RFID 태그를 이용한다.



[그림 4] u-아기 보호 시스템의 에뮬레이터 화면

u-아기 보호 시스템을 실행시킨 결과는 [그림 4]와 같이 에뮬레이터 화면으로 확인할 수 있다. <Weather Info>, <Mote Info Collector>, <RFID Reader Manager> 컴포넌트들은 날씨 정보(③), 현재 실내·외의 온도, 습도,

조도값(①,②), 리더기가 설치된 위치에 존재하는 사용자 정보(④)를 <Baby Care Service App> 컴포넌트에게 제공한다. <Baby Care Service App>는 각 정보를 취합하여 그 상황에 맞는 알맞은 서비스를 제공한다(⑤). 서비스로 제공되는 것은 XML 형태로 제공되고 있다(⑥). 서로 다른 OSGi 프레임워크의 원격 서비스를 호출하기 위해 RSP 번들은 효율적으로 사용되었다.

## 6. 결론 및 향후 연구 방향

본 논문에서는 분산된 OSGi 프레임워크들이 원격 서비스를 공유할 수 있도록 P2P기반 RSP 번들을 제안하였다. 제안된 P2P 기반의 RSP 번들을 아기 보호를 위한 유비쿼터스 홈 시스템에 실제 적용하여 분산되어 있는 OSGi 프레임워크의 서비스 공유가 효율적으로 이루어져 유비쿼터스 홈 네트워크와 같은 이질적이고 다양한 유비쿼터스 환경에 유용하게 사용될 수 있음을 보였다.

P2P 기반의 RSP 번들은 기존의 확장 서비스 레지스트리 검색으로 인한 서비스 응답 시간이 느려질 수 있는 문제점을 해결하기 위해서 로컬의 서비스 레지스트리만을 검색하는 방법이다. 내부적으로 멀티캐스트와 Proxy을 이용하여 다른 OSGi 프레임워크가 제공하는 서비스를 사용 할 수 있다. 이에 따라, 서비스를 이용하기 위해 번들 개발자나 사용자는 원격 서비스가 어디에 있는지를 알 필요가 없다. 마지막으로 원격 서비스의 공유가 가능해져 서비스를 여러 곳으로 분산시킬 수 있다.

OSGi 기반의 장비는 계속 증가하는 추세이므로 향후 더 많은 서비스들을 RSP번들을 통해 공유할 필요가 생길 것이다. 하지만 권한 설정에 대한 기능이 없어서 원격 서비스로 정의한 서비스는 멀티캐스트 그룹내의 모든 사용자가 마음대로 사용할 가능성이 있다. 따라서 분산된 OSGi 프레임워크에서 제공하는 서비스의 보안기술을 연구하여 허가된 사용자만 이용 할 수 있도록 보완할 예정이다.

## 참고 문헌

- [1] Open Service Gateway initiative (OSGi), "About the OSGi Service Platform - Technical Whitepaper", OSGi Alliance, <http://www.osgi.org>, June 2007.
- [2] C. Lee, D. Nordstedt, and S. Helal, "Enabling Smart Spaces with OSGi," IEEE Pervasive Computing, Vol. 2, Issue 3, pp. 89-94, July\_Sept. 2003.
- [3] Kyuchang Kang, Jeunwoo Lee, Hoon Choi, "Extended Service Registry for Distributed

Computing Support in OSGi Architecture" Advanced Communication Technology, 2006. ICACT 2006. The 8<sup>th</sup> International Conference, pp. 1631-1634, Feb. 2006

- [4] 김동욱, 최성술, 지충원, 김정선, "OSGi 아키텍처에서 이기종간의 분산 컴퓨팅 지원을 위한 확장된 서비스 레지스트리", 한국정보과학회, 학술발표논문집, Vol. 33, No. 2(D), pp. 426-430, 2006.
- [5] Tom Harpin, "Using java.lang.reflect.Proxy to Interpose on Java Class Methods", <http://java.sun.com/developer/technicalArticles/JavalP/Interposing/>, July 2001.