

## 이질의 네트워크 간의 상호운용을 위한 양방향 지니 어댑터 설계 및 구현

(Design and Implementation of Two-way Jini Adaptor for  
Interoperability between heterogeneous Networks)

오 일 진 <sup>†</sup>      황 윤 영 <sup>†</sup>      이 규 철 <sup>\*\*</sup>  
(Il-Jin Oh)    (Yun-Young Hwang)    (Kyu-Chul Lee)

**요 약** 유비쿼터스 환경이 도래하면서 유비쿼터스 환경을 제공하기 위한 서브네트워크들이 다양화되고 있다. 유비쿼터스 환경에서 서비스 검색을 위해 지니(Jini), UPnP, DPWS 등 여러 서브네트워크들이 제공되고 있다. 하지만 이러한 서브네트워크들은 서비스 기술방식, 통신방식 및 시스템을 구성하기 위한 언어가 상이함으로 인하여 서로 간에 상호운용이 되지 않는다. 유비쿼터스의 정의를 달성하기 위해서는 모든 사용자가 모든 서비스를 발견하고 사용할 수 있어야한다. 하지만 기존의 서브네트워크는 이러한 요구조건을 만족시키지 못하고 있는 실정이다. 본 논문에서는 서브네트워크의 상호운용을 위하여 기존의 방식과는 달리 웹서비스를 중간매개체로 하여 서비스들을 검색하고 사용하는 방식인 WSUN(Web Services on Universal Networks)을 기반으로 지니 시스템과 이질의 서브네트워크 간의 상호운용을 위한 지니 어댑터를 설계하고 구현한다. 지니 어댑터는 Lookup Service, 지니 사용자, 웹서비스 사용자를 확장한 것이다. 지니 어댑터는 지니 서비스를 발견하여 웹서비스로 변환하고 UDDI에 등록한다. 지니 어댑터는 지니 시스템에서 발생하는 이벤트를 처리하여 WSUN을 이용하여 이질 서브네트워크 사용자의 서비스 검색에 활용한다. 지니 어댑터를 통하여 WSUN 클라이언트는 지니 서비스 상태정보를 활용한 서비스 검색 및 사용이 가능하게 한다. 그리고 지니 사용자는 지니 어댑터를 통해 이질의 서브네트워크에서 제공하는 서비스를 검색 및 사용이 가능하게 한다.

**키워드** : 유비쿼터스, 웹서비스, 지니, 서비스 상호운용

**Abstract** The advent of the ubiquitous environments, the sub-networks is various to offer ubiquitous environments. There are diversify sub-networks such as Jini, UPnP, DPWS. To realize ubiquitous, any users can discovery and services in any sub-networks. However, they cannot support these requirement, because they don't do interoperability with other networks. In this paper, we introduce a Jini adaptor of WSUN(Web Services on Universal Networks). The WSUN is the framework which supports interoperability through the intermediation of Web services. The Jini adaptor has there roles : Lookup Service, Jini Client, Web Service Client. It converts Jini services to Web services and registers them. to UDDI registry. In addition the Jini adaptor process the events which happening in Jini system then that is utilized when another sub-network client search a service. Because the adaptor updates a service status continuously, it is possible that provides a service discovery and use to apply the status information. In conclusion, a Jini client is able to find and use a service which exists in heterogeneous sub-network through the Jini adaptor.

**Key words** : Ubiquitous, Web Services, Jini, Service Interoperability

· 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업(IITA-2008-C1090-0801-0031)의 연구결과로 수행되었음

<sup>†</sup> 학생회원 : 충남대학교 컴퓨터공학과  
victory25@cnu.ac.kr  
yyhwang@cnu.ac.kr

<sup>\*\*</sup> 종신회원 : 충남대학교 컴퓨터공학과 교수  
kclee@cnu.ac.kr

논문접수 : 2008년 8월 25일  
심사완료 : 2008년 10월 21일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제14권 제9호(2008.12)

### 1. 서론

컴퓨팅 및 네트워크의 발전은 정보가전의 기능을 다양화하고 지능화를 촉진시키고 있다. 이러한 정보가전의 발전으로 유비쿼터스에 대한 관심이 증가하고 있다. 유비쿼터스는 언제, 어디서든, 어떤 네트워크에 있는 어떤 디바이스의 어떤 서비스도 이용 가능하도록 지원하는 환경이다[1]. 유비쿼터스 환경을 실현하기 위하여 서비스 동적 발견 및 바인딩을 제공하는 다양한 서비스 디스커버리 미들웨어(지니[2], DPWS[3], HAVi[4] 등)들이 개발되었다. 본 논문에서는 서비스 디스커버리 미들웨어를 서브네트워크라 칭한다. 하지만 이러한 서브네트워크들은 표 1과 같이 서비스 기술방식, 통신방식, 시스템 기술 언어에서 차이점을 갖는다.

각각의 서브네트워크는 서비스를 사용하기 위해 필요한 서비스 정보를 자바 클래스, WSDL, 구조체를 이용하여 기술하고 있다. 그리고 각각은 자바의 통신방식인 RMI(Remote Method Invocation), HTTP(Hypertext Transfer Protocol)기반의 SOAP(Simple Object Access Protocol)메시지, 무선통신 기반인 IEEE1394를 이용하여 통신을 하고 있으며, 지니와 HAVi는 자바를 이

용하여 구현하고, DPWS(Devices Profile for Web Services)는 C++ 및 자바를 이용하여 구현이 가능하다.

진정한 유비쿼터스를 실현하기 위해서는 서브네트워크 간의 상호운용이 필요하다. 그러나, 현실적으로 각 서브네트워크는 서로 다른 형태의 서비스 기술방식과 통신 프로토콜을 사용하고 있어서 이러한 요구조건을 해결하지 못하고 있다.

서브네트워크의 상호운용을 위한 방식으로는 서브네트워크 간의 개별 브릿지 방식과 통합 미들웨어 구조 기반 방식이 있다.

서브네트워크 간의 브릿지 연동 방식은 이기종 서브네트워크들을 연동하기 위한 서브네트워크 간에 1:1 브릿지 방식으로 상호 운용을 해결하는 방식이다[10,11]. 이 방식은 두 서브네트워크간의 연동은 용이한 반면, 다양한 서브네트워크간의 연동을 해결하지는 못한다. 새로운 서브네트워크가 등장하게 되면 현재 존재하는 서브네트워크에 대하여 n(n-1)만큼의 브릿지를 생성해야 한다. n은 현재 존재하는 서브네트워크의 수를 나타낸다. 새로운 서브네트워크의 등장은 브릿지의 개수도 증가할 뿐만 아니라 연결에 있어서 복잡성을 갖게 되며, 서브네트워크간의 연결을 위해 일관된 방법을 제시하지 못하고 있으며, 확장성에도 문제가 있다.

통합 미들웨어 방식은 하나의 추상화된 공통 계층을 통하여 각 서브네트워크들을 연결하는 구조를 가지고 있다[5,6]. 이러한 구조는 새로운 미들웨어가 출현하더라도 해당 브릿지만 구현되면 쉽게 다른 서브네트워크들

표 1 다양한 서비스 디스커버리 미들웨어

구분	지니	DPWS	HAVi
서비스기술	자바 클래스	WSDL[13]	구조체
통신방식	RMI	SOAP	IEEE1394
언어	자바	다양함	자바

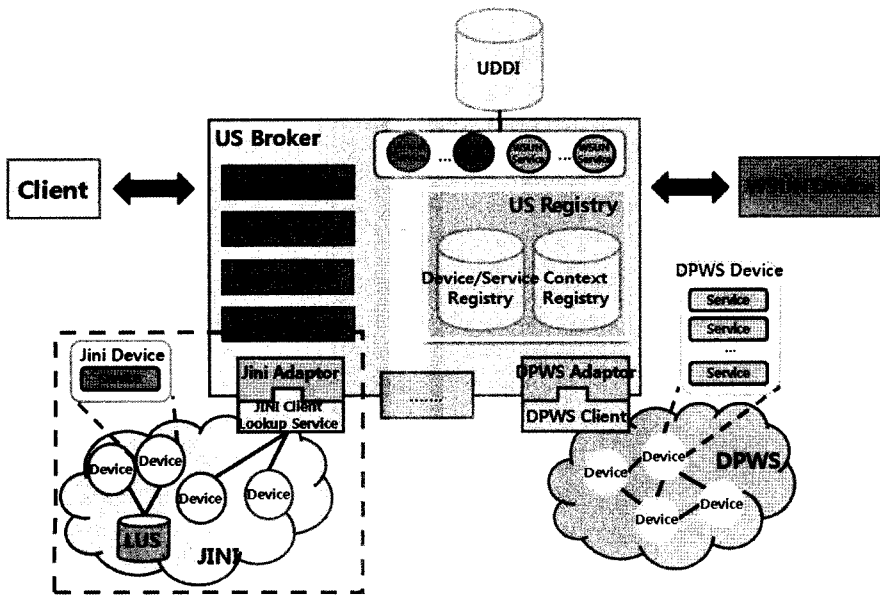


그림 1 WSUN 아키텍처

과 통합될 수 있다.

본 논문에서는 웹서비스를 중간 매개체로 하여 서버네트워크 간에 상호운용을 해결하기 위하여 Web Services on Universal Networks(WSUN)[7]을 활용한다.

WSUN은 그림 1과 같은 구조를 갖는다. WSUN은 통합마들웨어 방식의 아키텍처로 유니버설 서비스 브로커(Universal Service Broker)를 중심으로 서버네트워크의 서비스들의 상호운용을 제공하고 있다. WSUN은 어댑터를 이용하여 서버네트워크에 존재하는 서비스를 발견하게 되고 등록 에이전트(Publish Agent)를 이용하여 서비스를 UDDI[12] 및 유니버설 서비스 레지스트리(US Registry)에 저장하게 된다. 모든 사용자는 질의 에이전트(Query Agent)를 통하여 UDDI에 등록된 서비스를 발견하게 되며, 사용자에게 의해서 선택된 서비스를 라우팅 프록시(Routing Proxy)를 통하여 서비스와 바인딩을 맺게 된다.

본 논문에서는 WSUN을 기반으로 웹서비스 사용자가 지니 서비스를 발견 및 사용, 지니 사용자가 웹서비스 발견 및 사용을 위한 지니 어댑터(Jini Adaptor)를 기반으로 설명한다.

본 논문의 구성은 다음과 같다. 2장에서는 지니 시스템과 지니와 웹서비스간의 서비스 변환 및 검색을 위한 시스템을 소개한다. 3장에서는 지니 어댑터에 대한 설계를 소개하며, 4장에서는 지니 어댑터의 설계에 따른 구현을 소개한다. 5장에서는 결론 및 향후연구를 소개한다.

**2. 관련 연구**

본 장에서는 지니 어댑터에서 룩업서비스에 등록된 서비스 및 이벤트를 발견하여 UDDI 및 WSUN내의 유니버설 레지스트리에 반영하기 위하여 지니 시스템에 대하여 알아본다. 그리고 기존에 지니 시스템과 웹서비스 간의 서비스 변환을 통한 서비스 검색 및 사용을 위한 연구인 General brOkering Architecture Layer(GOAL)에 대하여 설명한다.

**2.1 지니(Jini)[2,8]**

지니 시스템은 자바를 기반으로 다양한 방식으로 네트워크에 접속되어 있는 지능형 기기들이나 소프트웨어들이 동적으로 상호 작용을 할 수 있는 기술이다. 지니 시스템은 그림 2와 같은 구조를 갖는다.

지니 서비스는 프록시(Proxy)에 서비스를 검색하고 사용하기 위한 정보를 담아 룩업서비스(Lookup Service)에 등록한다. 룩업서비스는 서비스와 사용자 간의 브로커 역할을 담당한다. 그리고 지니 시스템은 RMI를 이용하여 서로 간에 통신을 한다.

지니 시스템은 프록시에 서비스 아이디(ServiceID), 서비스 인터페이스 오브젝트(Service Interface Object),

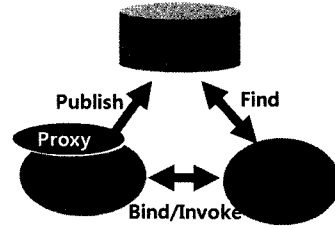


그림 2 지니 구조

어트리뷰트세트(AttributeSets)정보를 제공한다. 어트리뷰트세트에는 서비스에 대한 추가 정보로, 주소, 위치, 서비스 설명, 서비스 상태 정보관련 정보를 담게 된다. 그리고 지니 시스템은 그룹정보를 활용함으로써 서비스에 대한 이용제한을 하기위한 방법으로 활용한다. 모든 사용자가 사용가능한 서비스는 'Publish'의 그룹에 속하게 되며, 그룹정보는 관리자에 의해서 등록, 삭제, 변경될 수 있다. 관리자에 의해서 등록된 그룹은 그룹에 속한 사용자만이 사용이 가능하다.

지니 시스템은 룩업서비스의 서비스레지스터럴(Service-Registrar)에 이벤트 정보를 저장하게 된다. 저장된 이벤트 정보는 다음과 같은 세 가지 범주를 가지고 있다.

- TRANSITION\_NOMATCH\_MATCH : 서비스의 등장 및 재등장과 관련 이벤트를 처리한다.
- TRANSITION\_MATCH\_NOMATCH : 서비스의 퇴장 및 서비스의 속성이 변화가 발생하였을 때 이벤트를 처리한다.
- TRANSITION\_MATCH\_MATCH : 서비스에서 발생하는 오류 및 서비스의 상태를 바꿀 때 발생하는 이벤트를 처리한다.

지니 사용자는 서비스레지스터럴에 저장된 이벤트를 위의 세 가지 범주를 이용하여 지니 서비스와 관련된 이벤트를 처리할 수 있다.

**2.2 General brOkering Architecture Layer(9)**

General brOkering Architecture Layer(GOAL)은 그림 3과 같은 구조를 갖는다. GOAL은 웹서비스를 UDDI에 등록하고 동일 웹서비스에 대하여 지니 서비스로 변환을 하여 룩업서비스에도 등록을 하게 된다. 지니 서비스는 웹서비스로 변환이 되어 UDDI에도 등록되게 된다. 이렇게 각각 등록된 서비스에 대하여 웹서비스 사용자는 UDDI를 검색을 하여 모든 서비스를 사용하게 되며, 지니 사용자는 룩업서비스를 통하여 모든 서비스를 검색하고 사용하는 형태를 갖는다.

그러나 GOAL은 이벤트 정보를 활용하지 않는다. 이로 인하여, 웹서비스 사용자가 지니 서비스를 사용함에 있어서 지니 서비스가 현재 사용가능함을 판단할 수 없으며, 서비스의 현재 위치를 고려하지 않으므로써 사

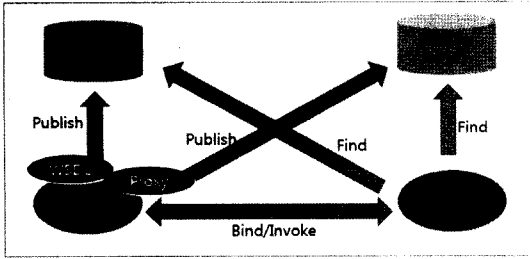


그림 3 GOAL의 구조

용자의 위치에서 사용가능한 서비스 검색이 불가능하다. 지니 사용자가 웹서비스를 사용함에 있어서도 새롭게 등장한 서비스에 대하여 검색이 불가능하다.

GOAL은 모든 웹서비스를 지니 서비스로 1:1 변환을 함으로써 지니 서비스를 저장하기 위하여 대용량의 북업서비스를 필요로 한다. 그리고 북업서비스가 재부팅되는 경우 등록되어 있던 모든 정보가 사라지게 됨으로써, 지니 서비스를 재등록시키고, 웹서비스에 대하여 다시 지니 서비스로 변환 및 등록이 이루어져야 한다.

### 3. 지니 어댑터

지니 어댑터는 웹서비스와 지니 시스템 간의 상호운용을 위하여 그림 4와 같이 지니 사용자, 북업서비스, 웹서비스 사용자를 확장하여 어댑터의 기능을 담당하게 된다.

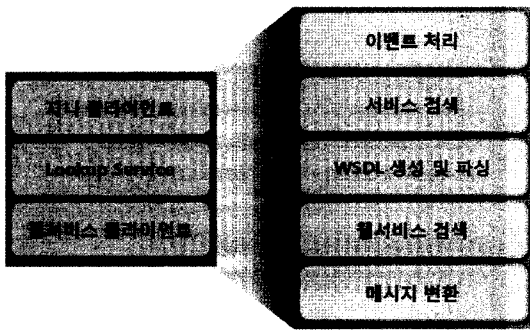


그림 4 지니 어댑터의 역할 및 기능

북업서비스에 등록된 서비스 및 이벤트를 발견 및 처리를 위하여 지니 사용자를 이용한다. 북업서비스를 이용하여 지니 사용자가 웹서비스를 검색하기 위하여 사용된다. 그리고 지니 사용자가 웹서비스를 사용하기 위하여 라우팅 폴록시를 호출하는데 웹서비스 클라이언트 기능을 사용한다. 그리고 추가적으로 메시지 및 프로토콜의 차이를 해결하기 위하여 메시지 변환 기능을 갖게 된다.

### 3.1 지니 어댑터 설계

지니 어댑터는 그림 5와 같은 구조를 갖는다.

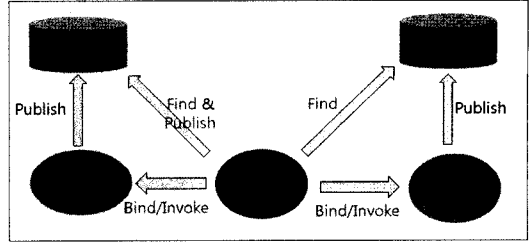


그림 5 지니 어댑터의 아키텍처

지니 어댑터는 북업서비스를 검색하여 서비스에 대한 정보를 UDDI에 등록한다. 이질의 사용자는 지니 어댑터를 통하여 UDDI에 등록된 웹서비스를 검색하게 되며, 지니 서비스를 사용하게 된다. 그리고 지니 사용자는 지니 어댑터를 통하여 웹서비스를 검색하게 되며, 지니 어댑터를 통하여 웹서비스를 사용하게 된다.

### 3.2 웹서비스 사용자의 지니 서비스 발견 및 사용

그림 5의 설계에 따라 웹서비스 사용자가 지니 서비스를 발견하고 사용하기 위해서는 지니 서비스 및 이벤트 정보를 발견하고 지니 서비스에 대하여 웹서비스로 변환 과정이 필요로 한다. 그리고 이와 같은 정보에 대하여 UDDI와 유니버설 레지스트리에 저장하게 된다.

#### • 웹서비스 생성

이질의 네트워크 사용자가 WSUN을 이용하여 지니 서비스를 발견하고 사용하기 위해서는 웹서비스로의 변환 과정을 거치게 된다. 웹서비스를 생성하기 위해서는 다음과 같이 4단계를 필요로 한다.

- ① 지니 어댑터에서 지니 서비스 소스파일을 요청한다.
- ② 지니 어댑터에서 지니 서비스 소스파일을 다운로드한다.
- ③ Axis[14]를 이용하여 웹서비스를 생성한다.
- ④ WSDL 주소 값을 레지스트리에 저장한다.

위와 같은 과정을 거쳐 생성된 웹서비스는 UDDI에 비즈니스 엔터티에는 WSUN, 비즈니스 서비스에서는 지니 서비스 이름을 등록한다.

#### • 지니 시스템의 이벤트 처리

본 논문에서는 지니 시스템과 이질의 시스템이 각각 운용되고 있는 환경에서 유니버설 서비스 브로커가 등장하였을 때의 이벤트(Design Time)와 지니 시스템에서 발생하는 총 7개의 이벤트(Run Time)에 대하여 처리를 한다.

#### - 북업과정(Hook-up)

Hook-up은 설계 과정(Design Time)으로 지니 시스템과 이질의 시스템이 각각 운용되고 있는 환경에서 유

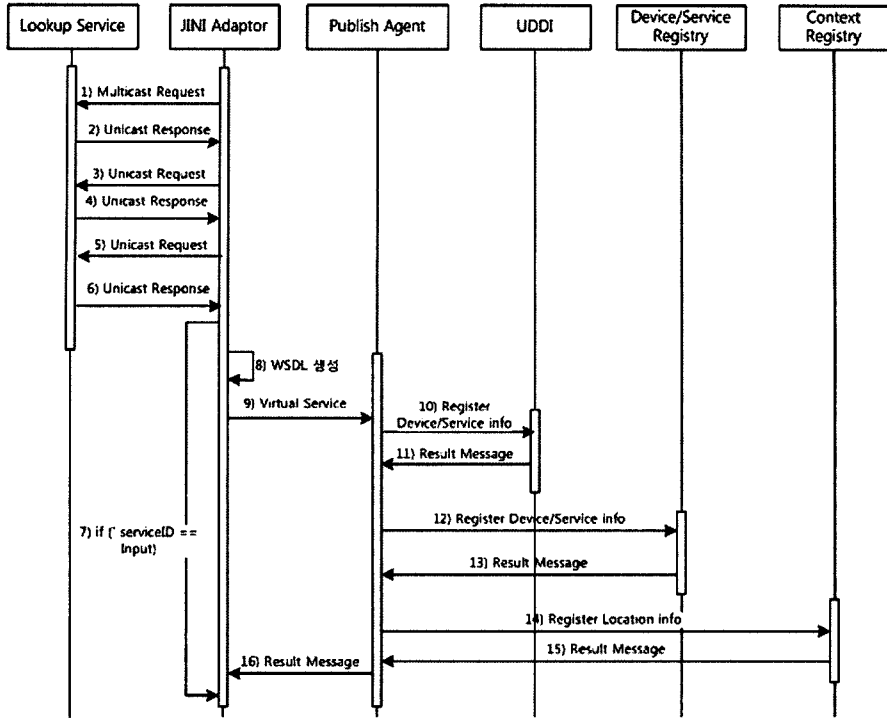


그림 6 Hook-Up과정

니버설 서비스 브로커가 등장하여 지니 어댑터가 동작을 시작하면서 발생하는 이벤트이다. 이때는 룩업서비스에 등록되어 있는 모든 서비스 및 이벤트 정보를 활용하여 서비스를 등록하는 과정으로 그림 6과 같은 과정을 거치게 된다.

지니 어댑터가 등장하여 지니 사용자 모듈을 이용하여 룩업서비스를 찾고 등록된 서비스에 대한 정보를 얻어온다(1-6). 한 개의 지니 서비스가 여러 개의 룩업서비스에 등록이 가능하기 때문에, (7)에서 서비스에 대하여 ServiceID를 판별하여 전에 등록이 되었던 서비스인가를 판별하고 등록된 서비스가 아니라면 다음을 진행한다. 지니 서비스는 웹서비스 생성과정(8)을 거쳐 각각의 레지스트리에 저장하기 위하여 추출된 서비스 정보를 저장하기 위하여 등록 에이전트에게 인자 값을 넘기게 된다(9). 등록 에이전트는 지니 어댑터로부터 넘어온 값을 각각의 레지스트리에 저장하는 과정을 거친다(10-16).

다음에 기술하는 이벤트 처리는 동작과정(Run Time)으로 지니 어댑터가 동작하고 있는 상태에서 지니 시스템에서 발생하는 이벤트이다.

- 지니 서비스 등장

지니 서비스 등장 이벤트는 지니 어댑터가 동작하고 있는 상태에서 지니 시스템의 룩업서비스에 새로운 서비스가 등장 하였을 때를 말한다. 이와 관련된 이벤트는

룩업서비스의 레지스터(Registrar)에 등록된 이벤트를 활용하게 된다. 지니 어댑터는 관련 이벤트를 이용하여 등록된 지니 서비스에 대하여 등록 에이전트(Publish Agent)를 통하여 지니 서비스 정보를 디바이스/서비스 레지스트리, 컨텍스트 레지스트리 및 UDDI에 등록시키는 과정을 나타낸다.

지니 서비스 등장과 관련하여 이벤트를 처리하는 과정은 그림 7과 같은 과정을 처리한다. (1-5)과정은 지니 시스템에서 서비스가 룩업서비스에 등록되는 과정과 동일하다. 서비스가 등록되어 이벤트가 발생하게 되면 지니 어댑터는 룩업서비스로부터 관련 이벤트 정보를 받아오게 된다(6-7). 관련 정보를 받게 되면 서비스에 대한 ServiceID를 이용하여 질의 에이전트를 통하여 동일 서비스가 레지스트리에 등록이 되어 있는 상태인가를 판별하게 된다(8-11). 지니 서비스에 대한 정보가 존재할 경우, 그 서비스는 웹서비스로 등록이 되었던 지니 서비스이다. 그러한 서비스의 경우 (13-16)의 과정을 거치게 된다. (13-16)과정은 현재의 서비스가 전에 등록이 되었으나, 서비스를 사용할 수 없는 상태이므로, 서비스의 상태를 가능상태로 바꾸는 과정을 처리한다.

하지만 존재하지 않는 서비스의 경우, 지니 어댑터는 관련 서비스에 대한 정보를 룩업서비스에 요청하게 되고, 서비스의 정보를 받아오게 된다(17-18). (19-27)은

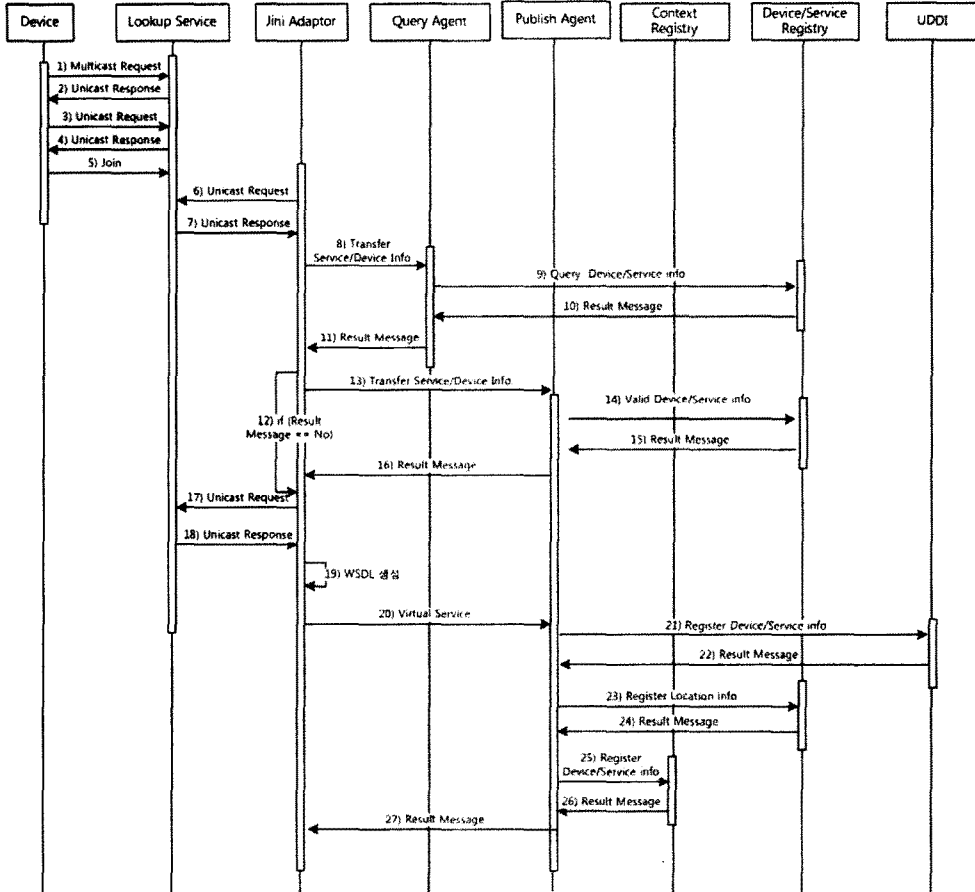


그림 7 지니 서비스 등장관련 이벤트 처리 과정

Hook-Up 단계에서 서비스를 저장하기 위한 과정과 동일하다.

- 서비스 상태정보 변경

그림 8은 지니 서비스의 상태 정보가 변경되는 이벤트를 처리하는 과정을 나타낸다. 서비스 상태 정보 변경 이벤트는 서비스가 현재 사용 중이거나 서비스에서 오류가 발생하여 사용하지 못하는 경우를 말한다.

지니 서비스는 상태정보가 변경이 발생하게 되면, 록

업서비스에 관련 이벤트를 등록하게 된다(1). 지니 어댑터는 록업서비스로부터 관련 이벤트를 받아오게 된다(2-3). 지니 어댑터는 관련 이벤트를 이용하여 레지스트리에 등록된 서비스의 상태정보를 변경하기 위한 과정을 거치게 된다(4-7). 지니 서비스는 상태정보를 다양하게 처리하고 있지만, 본 논문에서는 서비스의 상태정보를 사용가능과 사용불가능으로 구분하고 있다. 현재 서비스가 사용 중이거나 오류가 발생하게 되면 다른 사용자는 서비스를 사용할 수 없기 때문이다.

- 서비스 기술 정보 변경

그림 9는 지니 서비스의 기술 정보가 변경되는 이벤트를 처리하는 과정을 나타낸다. 서비스 기술정보 변경은 지니 서비스의 주소, 위치와 같은 정보가 변경이 되어, 록업서비스에 등록되는 경우이다. 지니 서비스는 관련 이벤트를 록업서비스에 등록하게 되고(1), 지니 어댑터는 관련 이벤트 정보를 록업서비스로부터 받아오게 된다(2-3). 관련 정보에 대하여 레지스트리에 등록된 서비스정보를 변경하게 된다(4-9).

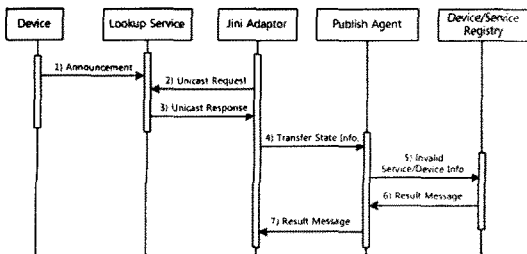


그림 8 상태정보 변경 처리

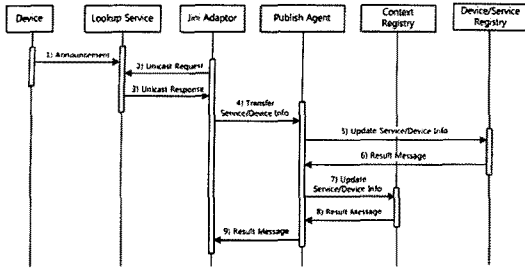


그림 9 서비스 기술정보 변경 처리

- 지니 서비스 그룹정보 등록

지니 시스템은 관리자에 의하여 그룹정보를 활용하여 서비스를 사용을 제한하고 있다. 그림 10은 관리자에 의하여 추가된 그룹에 대하여 서비스가 등록되어 지니 어댑터에서 관련 이벤트를 처리되는 과정을 나타낸다. 디바이스는 룩업서비스에 그룹정보를 추가 시키고 서비스

를 그룹에 추가 시키게 된다. 현재 다른 룩업서비스에 등록되어 있는 서비스는 그룹에 속한다면, 그룹이 추가된 룩업서비스에도 추가가 되게 된다(1-5). (7-28)까지는 서비스의 등장과 동일하게 동작한다.

- 지니 서비스 그룹정보 삭제

지니 서비스 그룹정보 삭제는 관리자에 의해서 그룹 정보가 삭제되면서 그룹에 속해있던 서비스에 대하여 사용이 불가능한 상태가 발생하기 때문에 이를 처리하는 과정을 나타낸다. 그림 11은 룩업서비스에 등록된 그룹 정보들 중에서 그룹정보가 삭제되는 경우의 처리를 나타낸다.

룩업서비스에 그룹정보 삭제 시 그룹에 속한 지니 서비스는 사용이 불가능하므로 서비스의 상태 정보를 변경하는 과정을 거치게 된다(5-8). 하지만 다른 룩업서비스에 등록된 동일 지니 서비스에 대해서는 상태 변경이 일어나지 않는다.

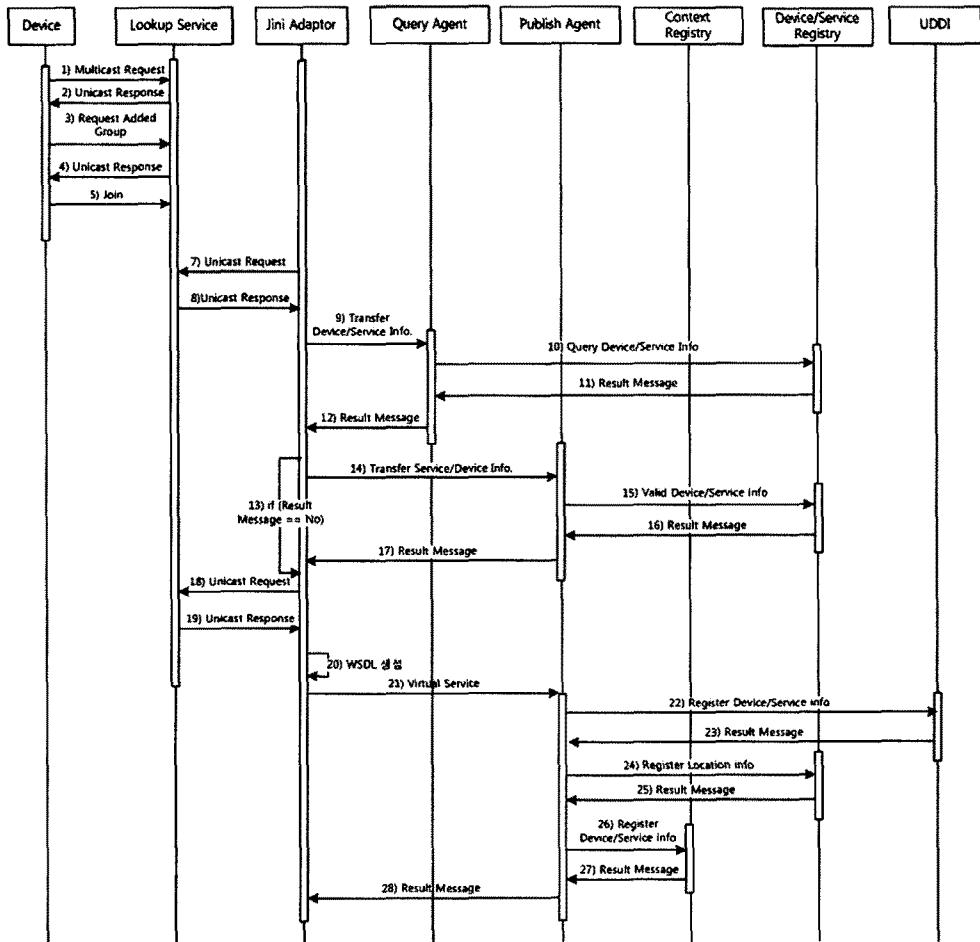


그림 10 그룹정보 등록 처리 과정

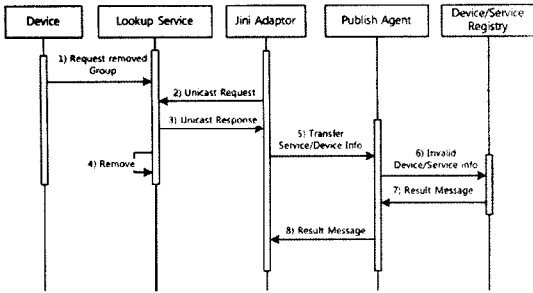


그림 11 그룹정보 삭제 처리 과정

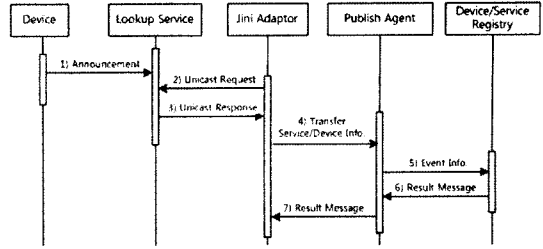


그림 13 정상적인 지니 서비스 퇴장 처리 과정

- 록업서비스의 퇴장

록업서비스 퇴장은 록업서비스의 기능을 하지 않을 경우에 록업서비스에 등록된 서비스는 더 이상 서비스를 사용할 수 없는 상태가 된다. 따라서 그림 12와 같이 이벤트를 처리할 필요가 있다.

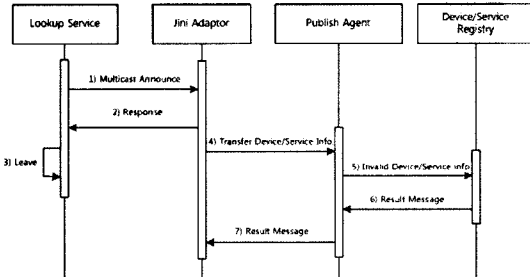


그림 12 록업서비스 퇴장 처리 과정

록업서비스는 더 이상 기능을 하지 않는 경우 모든 디바이스에게 알리게 되며, 모든 디바이스로부터 응답이 오게 되면 록업서비스의 기능을 중지하게 된다. 지니 어댑터는 이와 같은 이벤트 정보를 활용하여 록업서비스에 등록된 서비스에 대하여 상태 정보를 변경하는 과정을 거치게 된다.

록업서비스는 더 이상 기능을 하지 않기를 원할 경우 모든 디바이스에게 정보를 알리고 응답을 받게 되면 록업서비스기능을 중지한다(1-3). 지니 어댑터는 알림 메시지를 받아 록업서비스와 관련된 모든 서비스에 대하여 서비스의 상태를 변경한다(4-7).

- 지니 서비스 퇴장

지니 서비스 퇴장은 록업서비스에 등록된 서비스가 더 이상 제공되지 않는 경우를 말한다. 지니 서비스의 퇴장은 두 개로 나누어 볼 수 있는데, 정상적으로 록업서비스에 알리고 서비스 기능을 중지하는 경우와 디바이스의 문제로 인하여 서비스를 제공할 수 없는 상태로 나누어 볼 수 있다. 전자의 경우 록업서비스에 관련 이벤트 정보가 등록되지만, 후자의 경우 리즈 시간을 활용

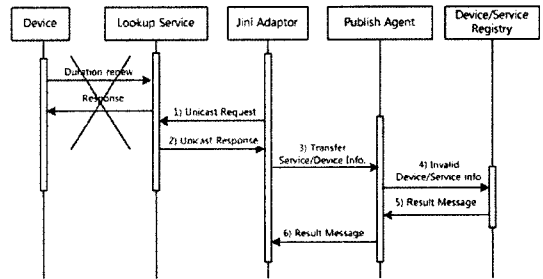


그림 14 비정상적인 지니 서비스 퇴장 처리 과정

하여 이벤트를 처리하게 된다.

그림 13은 지니 서비스가 정상적인 퇴장과 관련 이벤트를 처리하는 과정을 나타낸다. 정상적인 퇴장은 록업서비스에 퇴장과 관련 이벤트를 저장하게 된다. 지니 어댑터는 이벤트 정보를 활용하여 서비스에 대한 상태 정보를 변경하는 처리를 하게 된다.

하지만, 비정상적으로 디바이스의 전원이나, 네트워크의 이상과 같은 상황으로 인하여 더 이상 서비스가 사용이 불가능한 경우는, 그림 14와 같은 처리가 이루어지게 된다. 지니 시스템에서는 리즈(Lease)를 이용하여 서비스 제공시간을 정하게 되는데, 리즈 시간이 되었을 경우 서비스 제공과 관련하여 리즈 시간을 늘리는 이벤트가 발생하지 않는다. 지니 어댑터는 이벤트가 발생하지 않으면, 현재 사용이 불가능한 서비스로 판단하고 서비스의 상태를 변경하는 처리를 하게 된다. 이때는 리즈 시간이 길수록 서비스 제공에 있어서 문제가 발생하게 된다.

• 웹서비스 사용자의 지니 서비스 발견 및 사용

지니 어댑터는 모든 이벤트를 레지스트리에 반영하게 된다. 웹서비스 사용자는 그림 15와 같은 형태로 서비스를 발견하고 사용하게 된다. 웹서비스 사용자는 서비스를 발견하기 위하여 질의 에이전트(Query Agent)를 통하여 서비스를 발견하게 된다(1-4). 웹서비스의 경우, 바로 서비스를 사용을 한다. 하지만 서브네트워크에서 제공하는 서비스의 경우 디바이스/서비스 레지스트리에서 서비스의 상태정보를 검색하게 된다(7-10). 웹서비스



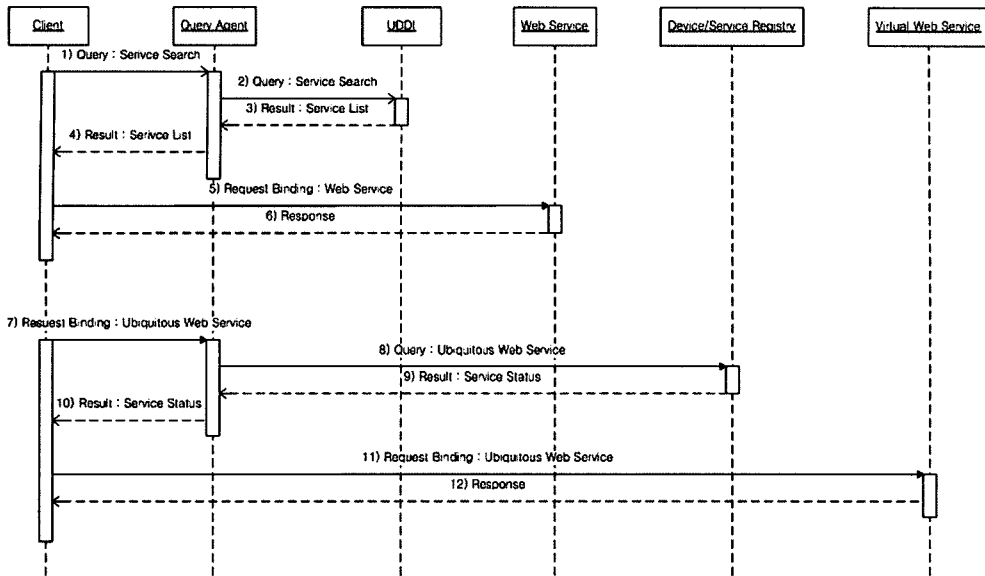


그림 15 웹서비스 사용자의 서비스 발견 및 사용

사용자는 상태정보를 반영하여 서비스를 사용하기 위하여 가상의 웹서비스를 호출하여 서비스를 사용하게 된다(11-12).

**3.3 지니 사용자의 웹서비스 발견 및 사용**

지니 시스템은 멀티 캐스트 메시지를 이용하여 룩업 서비스를 찾게 된다. 지니 사용자는 발견한 룩업서비스로부터 서비스레지스트럴 객체를 이용하여 Lookup()메소드를 호출하여 서비스를 발견하게 된다. 서비스를 발견하기 위하여 서비스템플릿(ServiceTemplate)를 활용하게 되는데, 서비스템플릿은 ServiceID, ServiceType, AttributeSetTemplate를 포함하게 되며, ServiceType을 활용하여 서비스를 검색하게 된다.

룩업서비스는 지니 사용자에게 ServiceID, ServiceName, Endpoint, Service Interface를 돌려준다.

지니와 웹서비스 간에 메시지 관계를 알아보면 표 2, 표 3과 같이 매핑이 가능하다.

지니 요청 메시지를 이용하여 웹서비스 검색하기 위하여 비교하게 되면, 표 2와 같이 비교를 하게 된다. 지니 시스템은 룩업서비스에 있는 서비스를 검색하기 위하여 ServiceID, ServiceType, AttributeSetTemplate를 이용하여 서비스를 검색한다. 이러한 정보 중에서 ServiceType을 이용하여 서비스를 검색한다. 본 논문에

표 2 지니 요청 메시지와 웹서비스 매핑관계

지니	웹서비스
ServiceID	UUID
ServiceType	Business Service Name
AttributeSetTemplate	X

표 3 지니 결과 메시지와 웹서비스 매핑관계

지니	웹서비스
ServiceID	UUID
ServiceName	Business Service Name
Endpoint	Jini Adaptor Address
Service Interface	제너릭 WS를 호출하기 위한 Interface

서는 웹서비스를 검색하기 ServiceType정보를 활용하여 웹서비스 검색한다. 이를 통하여 웹서비스를 지니 서비스로 변환하지 않고 웹서비스를 발견할 수 있게 된다.

지니 사용자는 서비스검색 결과에 대하여 룩업서비스로부터 ServiceID, ServiceName, Endpoint, Service Interface를 받게 된다. 이와 매핑되는 웹서비스 정보는 표 3과 같다.

지니 어댑터는 지니 사용자에게 웹서비스의 UUID, 비즈니스 서비스 이름, 어댑터 주소 및 라우팅 프록시 내의 제너릭 웹서비스(Generic Web Service)를 호출하기 위한 인터페이스 받게 된다. 지니 사용자는 인터페이스를 통하여 제너릭 웹서비스 호출하게 되고 웹서비스와 바인딩이 된다.

• 제너릭 웹서비스(Generic Web Services)

제너릭 웹서비스는 라우팅 플락시 내에 존재한다. 제너릭 웹서비스는 모든 웹서비스를 사용할 수 있는 서비스이다. 그림 16과 같이 지니 사용자는 지니 어댑터로부터 제너릭 웹서비스를 호출할 수 있는 인터페이스를 통하여 웹서비스를 사용하게 된다.

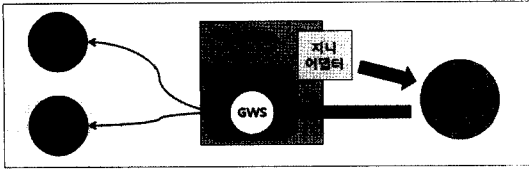


그림 16 제너릭 웹서비스

• 지니 사용자의 웹서비스 발견 및 사용

본 논문에서는 룩업서비스를 확장하여 wsLookup()메소드를 활용하여 UDDI를 검색하기 위한 질의 에이전트(Query Agent)를 통하여 서비스를 검색하게 된다.

지니 사용자가 웹서비스 발견하고 사용함에 있어서는 그림 17과 같은 과정을 거치게 된다.

지니 사용자는 룩업서비스에 서비스 검색을 요청하게 되면 ServiceType을 이용하여 UDDI 및 디바이스/서비스 레지스트리를 검색하게 된다(3-8). 지니 사용자는 웹서비스를 사용하기 위하여 지니 어댑터에 웹서비스를 호출하게 되고 라우팅 프로시를 통하여 서비스를 사용한다(11-14).

4. 지니 어댑터 구현

지니 어댑터는 그림 18과 같은 모듈로 구성된다.

- DiscoveryLookupService : 룩업서비스를 찾기 위한 모듈써 멀티캐스트로 룩업서비스를 찾게 된다.
- DiscoveryService : DiscoveryLookupService를 이용하여 찾은 룩업서비스로부터 모든 지니 서비스를 검색한다.
- MessageTrans : SOAP메시지와 RMI메시지 간의 변환을 담당한다.

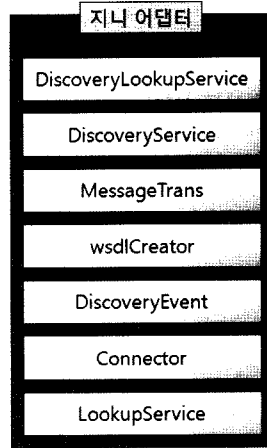


그림 18 지니 어댑터 모듈

- wsdlCreator : 지니 서비스를 웹서비스로 변환하는 모듈이다.
- DiscoveryEvent : 룩업서비스에 등록된 이벤트 정보를 추출하는 기능을 담당하는 모듈이다.
- Connector : 등록 에이전트(Publish Agent)와 연결하는 기능을 담당한다.
- LookupService : 기존 지니 시스템의 룩업서비스를 확장한 것으로 웹서비스를 검색하기 위한 지니 사용자의 요청을 처리하여 질의 에이전트를 호출하는 기능을 담당한다.

지니 서비스를 UDDI에 등록하기 위해서는 그림 19와 같은 형태로 모듈을 사용하여 등록을 하게 된다.

지니 어댑터는 DiscoveryLookupService를 통하여 룩업서비스를 발견하고 등록된 서비스를 DiscoverySer-

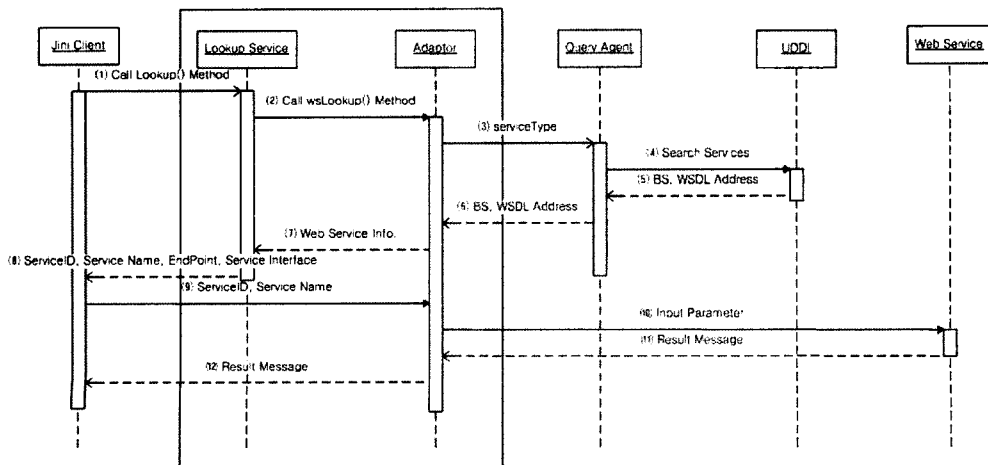


그림 17 지니 사용자의 웹서비스 발견 및 사용

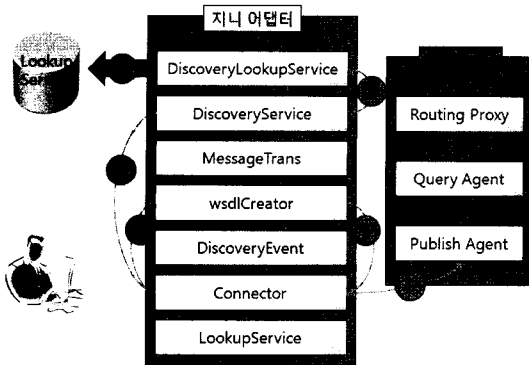


그림 19 지니 서비스의 UDDI등록

vice를 이용하여 서비스를 발견하고 Connector로 전달된다. 전달된 정보를 활용하여 wsdlCreator를 이용하여 웹서비스를 생성하고 등록 에이전트(Publish Agent)를 통하여 등록을 하게 된다.

지니 사용자는 서비스를 발견하기 위하여 그림 20과 같이 동작한다.

기존의 룩업서비스와 어댑터 내의 룩업서비스를 발견하고 질의를 보내어 서비스를 발견한다. 지니 어댑터는

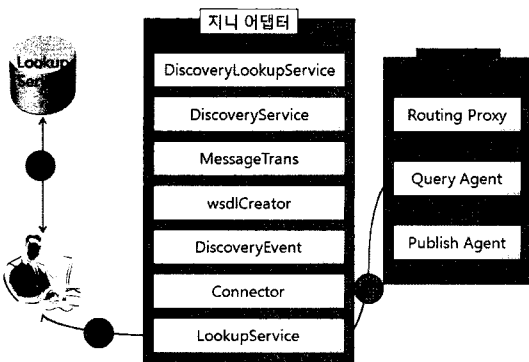


그림 20 지니 사용자의 질의 처리

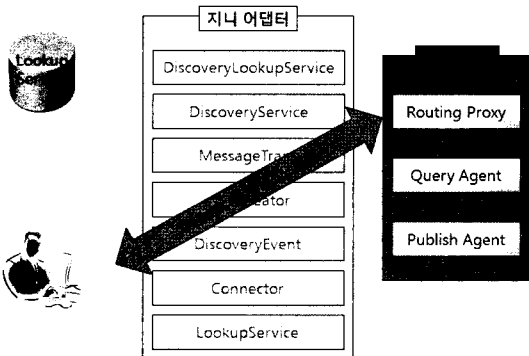


그림 21 지니 사용자와 웹서비스 바인딩

질의를 파싱하여 질의 에이전트를 통하여 웹서비스를 발견하게 된다.

지니 사용자에게 웹서비스 검색 결과에서 서비스를 사용하기 위해서 그림 21과 같은 형태를 이용하게 된다.

사용자가 받은 인터페이스를 통하여 라우팅 플록시 내의 제너릭 웹서비스를 호출하여 웹서비스와 바인딩을 하게 된다.

### 5. 결론 및 향후연구

지니 어댑터는 지니 시스템에서 발생하는 이벤트를 처리하여 검색에 활용을 하였다. 이를 통하여, 웹서비스 사용자에서 지니 서비스를 발견하고 사용함에 있어서 서비스 상태정보 및 위치 정보를 활용하여 서비스 정보를 제공할 수 있다.

지니 사용자에서 웹서비스를 발견하고 사용함에 있어서 룩업서비스를 확장하여 UDDI를 검색하여 웹서비스를 발견하게 되며, 웹서비스를 사용함에 있어서 제너릭 웹서비스를 이용하여 웹서비스를 사용하게 된다.

GOAL과 지니 어댑터를 비교하였을 때 다음과 같은 사항으로 비교가 가능하다.

- 웹서비스 사용자의 지니 서비스 검색 및 사용

GOAL과 지니 어댑터를 웹서비스 사용자가 지니 서비스를 발견하고 사용함에 있어서 서로간의 서비스 검색, 이벤트 정보, 웹서비스 생성시점에 따른 비교를 하면 표 4와 같다.

표 4 지니 서비스 사용에 따른 비교

구분	GOAL	지니 어댑터
서비스 검색	가능	가능
서비스 상태 이벤트정보활용	미사용	사용
웹서비스 생성	설계 과정	설계/동작 과정

두 시스템은 서비스를 동일하게 검색이 가능하지만 지니 어댑터는 이벤트 정보를 활용하여 서비스를 제공할 수 있다. 이는 현재 사용 가능한 정보의 검색이 가능하게 된다. 또한 실행이 되고 있는 런 타임에도 웹서비스가 생성됨으로써 새롭게 등장한 지니 서비스도 검색이 가능하게 된다.

- 지니 사용자의 웹서비스 검색 및 사용

GOAL과 지니 어댑터를 지니 사용자가 웹서비스를 발견하고 사용함에 있어서 서로간의 서비스 검색, 검색 대상, 서비스 변환 및 새롭게 등장한 웹서비스 검색에 따른 비교를 하면 표 5와 같다.

두 시스템은 동일하게 웹서비스를 검색이 가능하나 서로간의 검색대상이 GOAL은 룩업서비스인 반면, 지니 어댑터는 UDDI를 활용하게 된다. 지니 어댑터는 새롭

표 5 웹서비스 사용에 따른 비교

구분	GOAL	지니 어댑터
웹서비스 검색	가능	가능
웹서비스 검색 대상	복업서비스	UDDI
웹서비스의 지니 서비스로 변환	모든 서비스 변환	제너릭 웹서비스 활용
새롭게 등장한 웹서비스검색	불가능	가능

게 등장한 웹서비스에 대하여 검색이 가능하게 한다. 또한 검색대상의 차이에 따라 GOAL은 모든 웹서비스를 지니 서비스로 1:1 생성을 해야 한다. 그러나 지니 어댑터는 제너릭 웹서비스를 활용함으로써 서비스 변환이 없이 서비스 사용이 가능하게 한다.

향후에는 현재 지니 어댑터는 설계를 입증하기 위한 구현을 진행하였으나 유비쿼터스 환경에 최적화하여 구현을 변경이 필요로 한다. 또한 다른 서브네트워크 사용자에서 지니 서비스를 사용하기 위한 연구를 필요로 한다. 또한 다양한 시나리오를 적용하여 테스트를 필요로 한다.

참고 문헌

[1] M. Satyanarayanan. Pervasive Computing: Vision and Challenges. IEEE Personal Communications, pp.10-17, August 2001.

[2] Sun Microsystems. Jini Architecture Specification. <http://www.sun.com/jini/>.

[3] Hendrik Bohn, Andreas Bobek, Frank Golasowski, "SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains," April 2006.

[4] The Havi Organization, Havi Version 1.1 Specification. <http://www.havi.org>.

[5] Michael Ditzte, Guido Kamper, Isabell Jahnich, Reinhard Bernhardt-Grisson, "Service-based access to distributed embedded devices through the open service gateway," INDIN 2004, pp. 493-498, June 2004.

[6] Vittorio Miori, Luca Tarrini, Maurizio Manca, "DomomNet: A Framework and a Prototype for Interoperability of Domestic Middlewares Based on XML and Web Services," ICCE 2006 on Consumer Electronics, pp. 117-118, January 2006.

[7] 임형준, 오일진, 황윤영, 이경하, 이강찬, 이승운, 이규철, "유니버설 네트워크를 위한 웹서비스 프레임워크 설계 및 구현", 정보과학회논문지:컴퓨팅의 실제와 레터, 2008.4.

[8] Jan Newmarch, "Foundations of Jini2 Programming," Apress, 2006.

[9] Franco Cicirelli, Angelo Furfaro, Libero Nigro, "Integration and Interoperability between Jini services and Web Services," IEEE international Con-

ference on Service Computing(SCC 2007), 2007. 10.

[10] J. Allard, V. Chinta, S. Gundala, G. Richard III, "Jini Meets UPnP: An Architecture for jini/UPnP Interoperability," Symposium on Applications and the Internet, pp. 268-275, January 2003.

[11] B. Guillaume, R. Kumar, B. Helmut, and S. Thomas, "Methods for Bridging a HAVi Subnetwork and a UPnP Subnetwork and Device for Implementing said Methods," Thomson Multimedia, 2002.

[12] UDDI, <http://www.uddi4j.org/>

[13] WSDL, <http://wsdl4j.sourceforge.net/>

[14] AXIS, <http://ws.apache.org/axis/>



오 일 진  
2006년 공주대학교 컴퓨터공학과 졸업 (학사). 2006년~현재 충남대학교 컴퓨터공학과 과장. 관심분야는 XML, 웹서비스, 유비쿼터스 웹서비스(UWS), 웹 2.0



황 윤 영  
2002년 충남대학교 컴퓨터공학과(학사) 2004년 충남대학교 대학원 컴퓨터공학과(석사). 2004년~현재 충남대학교 대학원 컴퓨터공학과 박사과정. 관심분야는 서비스 지향 아키텍처, 유비쿼터스 웹서비스(UWS)



이 규 철  
1984년 서울대학교 컴퓨터공학(학사). 1986년 서울대학교 컴퓨터공학(석사). 1990년 서울대학교 컴퓨터공학(박사). 1989년~현재 충남대학교 컴퓨터공학과 교수. 1989년 3월~1994년 6월 미국 IBM Almaden Research Center 초빙연구원. 1995년 8월~1996년 8월 미국 Syracuse University, CASE Center 초빙 교수. 1997년 1월~1998년 1월 교육부 학술진흥재단 부설 첨단학술센터 파견 교수. 1997년 7월~현재 한국정보과학회 논문편집위원. 2000년 2월~2004년 2월 산업자원부 한국 ebXML. 전문위원회 위원장. 2003년 3월~현재 한국전자거래학회 이사. 2003년 3월~현재 조달청 목록자문위원. 2003년 8월~현재 웹 코리아포럼 부위원장. 2005년 1월~현재 한국기록관리학회 이사. 2006년 2월~현재 충남대학교 소프트웨어연구소 소장. 2006년 8월~현재 대통령자문 정부혁신 지방분권위원회 위원. 관심분야는 데이터베이스, XML, 정보 통합, 멀티미디어 시스템, e-비즈니스 시스템, 유비쿼터스 웹서비스(UWS)