

지니와 웹서비스간의 상호운용지원을 위한 어댑터 설계*

오일진^o, 임형준, 황윤영, 이규철[†]

충남대학교 컴퓨터공학과

{victory25^o, hyungjun25, yywang, kclee}@cnu.ac.kr

Design of Adaptor for interoperability based Jini and Web Services

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

Department of Computer Engineering Chungnam National University Daejeon, The Republic of Korea

요 약

유비쿼터스 환경이 도래하면서 다양한 서비스들이 산재하여 존재한다. 하지만 이러한 서비스들은 각자의 서비스 환경하에서 운용되고 있다. 서로간의 기술형태 및 통신 방식의 차이를 해결하고 진정한 유비쿼터스 환경을 이루기 위하여 웹서비스와의 상호운용도 필요로 한다. 본 논문에서는 지니와 웹서비스간의 상호운용을 위하여 WSUN을 기반으로 웹서비스와 지니 간에 상호운용을 가능하게 하는 지니 어댑터를 설명한다.

1. 서 론

유비쿼터스 환경이 도래하면서 다양한 서비스들이 도래하였다. 유비쿼터스는 언제 어디서든, 어떤 네트워크에 있는 어떤 디바이스의 어떤 서비스도 이용 가능하도록 지원하는 것이다.

유비쿼터스 환경하에서 서비스를 제공하기 위하여 Jini[1], DPWS[1] 등과 같은 서비스 디스커버리 미들웨어들이 제공되게 되었다. 그리하여 현재의 서비스는 <표 1>과 같이 다양한 형태의 서비스들이 존재하게 되었다

표 1 유비쿼터스 환경에서 산재하는 서비스

구분	지니	DPWS	웹서비스
서비스 기술	자바 클래스	WSDL	WSDL
통신방식	RMI	SOAP	SOAP
기술 언어	Java	Any	Any

이와 같은 서비스들은 서비스를 기술하는 방식 통신 방식 등의 차이를 갖고 있다. 이로 인하여 서로간의 상호운용을 지원하지 못하고 있다. 이렇게 산재하는 서비스들간의 상호 운용을 달성하여 진정한 유비쿼터스를 이루기 위해서는 서비스 디스커버리 미들웨어들간의 상호 운용뿐만 아니라 현재 존재하는 웹서비스와의 상호운용을 지원이 필요하게 된다.

본 논문에서는 지니와 웹서비스 간의 상호운용을 위하여 어댑터를 설계하여 각각의 클라이언트가 이질의 서비스를 사용하기 위한 방법을 제시하고자 한다.

지니는 썬마이크로 시스템즈에서 자바 기반으로 네트워크에 접속된 지능형 디지털 장비들이나 소프트웨어들

이 동적으로 상호 작용을 할 수 있게 하는 기술이다[2]

웹서비스는 XML기반으로 서비스를 기술하고 SOAP 메시지를 활용하여 서비스를 사용하는 것으로 단일 엔터티 패키징되고 다른 프로그램에 의해 사용되는 네트워크상에 등록된 기능들의 집합이다. 웹서비스는 오픈 분산 시스템을 만들기 위한 빌딩 블록이며 개인이나 기업들이 빠르고 쉽게 자신 혹은 자사의 디지털 자산을 월드와이드하게 사용할 수 있도록 제공하는 것이다[6]

표 2. 지니와 웹서비스의 비교

Feature	지니	웹서비스
디스커버리	멀티캐스트, 유니캐스트를 이용한 정적, 다이나믹	정적
검색 방식	자바 클래스 네임 기반	키워드, 화이트, 옐로우, 그린 페이지
서비스 기술정보	○	○
Proxy	○	×
Downloadable Code	○	×
기술 언어	Java	Any
원격 이벤트	○	○

<표 2>은 지니와 웹서비스 간의 다음과 같은 사항으로 비교한 결과를 나타낸다[3]

지니는 검색 방식에 있어서 RMI메시지를 활용하여 자바 클래스 네임 기반으로 서비스를 검색하며 웹서비스는 SOAP메시지를 활용하여 키워드기반으로 화이트 옐로우, 그린 페이지를 대상으로 서비스를 검색하게 된다. 또한 지니는 서비스를 사용하기 위하여 Proxy를 추가적으로 사용하며, 서비스를 기술하는 방식도 지니는 Java

* 본 연구는 정보통신연구진흥원의 대학 IT 연구센터 육성, 지원사업 (IITA-2008-C1090-0801-0031)의 연구 결과로 수행되었음.

† 교신저자

에 한정적인 반면 웹서비스는 Java뿐만 아니라 C++로도 기술이 가능하다 그러나 SOA형태를 갖는 지니와 웹서비스는 이와 같이 차이로 인하여 서로간의 상호운용이 불가능하다.

그리하여 본 논문에서는 지니와 웹서비스 간에 상호운용을 지원하기 위하여 지니 어댑터를 제안한다

2장에서는 지니와 웹서비스 간에 상호운용을 위한 연구로 GOAL(General brOkering Architecture Layer)을 소개하며, 3장에는 본 논문에서 기반으로 하고 있는 WSUN에 대하여 설명한다 4장에서는 상호운용을 지원하기 위하여 지니 어댑터의 기능을 기술하며 서비스를 변환, 검색 및 바인딩을 위한 방법을 기술한다 5장에서는 결론 및 향후 연구를 기술한다

2. 관련 연구

GOAL(General brOkering Architecture Layer)[4]은 와 <그림 1>같은 아키텍처를 갖으며, 웹서비스를 지니 서비스로 변환하고 지니 서비스를 웹서비스로 변화하여 각각의 레지스트리에 저장함으로써 두 개의 레지스트리가 동일 서비스에 대한 정보를 갖게 된다 지니 클라이언트는 Lookup Service를 검색하여 서비스를 검색하며 웹서비스 클라이언트는 UDDI를 대상으로 검색을 하게 된다.

서비스를 변환하기 위하여 두 개의 메타 서비스를 제공한다. GOAL2WS 메타 서비스는 지니 서비스를 웹서비스로 변환하기 위한 기능을 담당하며 WS2GOAL 메타 서비스는 웹서비스를 지니 서비스로 변환하기 위한 작업을 담당하게 된다. GOAL은 지니 서비스를 웹서비스 변환하기 위하여 Axis를 사용하였으며, 웹서비스를 지니 서비스로 변환하기 위하여 별도의 메타서비스를 이용하였다. 이는 Axis를 사용하여 웹서비스를 지니 서비스로 변환할 경우, 서비스의 중심이 되는 함수 부분을 자동적으로 생성하지 못하기 때문이다

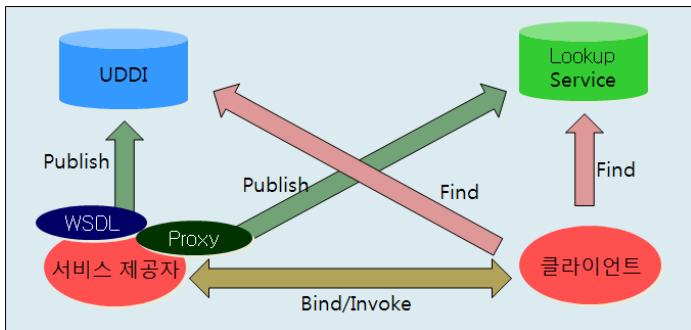


그림 1. GOAL

검색된 서비스 중에서 이질의 네트워크에서 제공하는 서비스의 경우 GOAL을 통하여 서비스와 바인딩을 하게 되는데, 이를 위하여 브릿지를 활용한다



그림 2. GOAL의 아키텍처

<그림 2>와 같이 GOAL에서 정의하고 있는 브릿지는 WS2GOALBridge와 GOAL2WSBridge가 존재한다.

GOAL2WSBridge는 웹서비스를 이용하여 지니 서비스에 접근이 가능하도록 하며 SOAP메시지를 RMI메시지로 변환하며, 그 반대의 기능도 담당하게 된다

WS2GOALBridge는 지니 클라이언트에서 웹서비스에 접근이 가능하도록 하기 위하여 사용하며 클라이언트와 웹서비스 간에 P2P형태로 서비스를 사용하게 된다

하지만, GOAL은 지니 시스템에서 발생하는 이벤트를 처리하지 않는다 이로 인하여 웹서비스 클라이언트가 지니 서비스의 사용가능 여부를 판단하지 못한다 또한 서비스의 위치를 검색에 있어서 활용하지 않음으로써 웹서비스 클라이언트가 현재 위치에서 사용가능한 지니 서비스를 판단하기가 어렵게 된다 또한 웹서비스에서 발생하는 이벤트를 처리하지 않음으로써 시스템이 가동되고 있는 상황에서 새롭게 등장한 웹서비스에 대하여 검색이 불가능하게 된다 또한 대량의 웹서비스 정보를 Lookup Service에 저장하기 위하여 대용량의 하드웨어를 필요로 하게 됩니다. 또한 Lookup Service는 휘발성의 성격을 갖기 때문에 Lookup Service가 등장하는 경우, 서비스를 등록시키기 위하여 시스템에 과부하가 걸리게 된다.

그리하여 본 논문에서는 이러한 문제를 해결하기 위하여 WSUN을 기반으로 지니 어댑터를 설명하고자 한다

3. WSUN(Web Services on Universal Networks)[1]

WSUN은 시스템에서 서비스를 검색하고 사용할 수 할 수 있는 범위를 나타낸다 <그림 3>과 같이 클라이언트, 서비스 제공자, US Broker로 구성이 된다

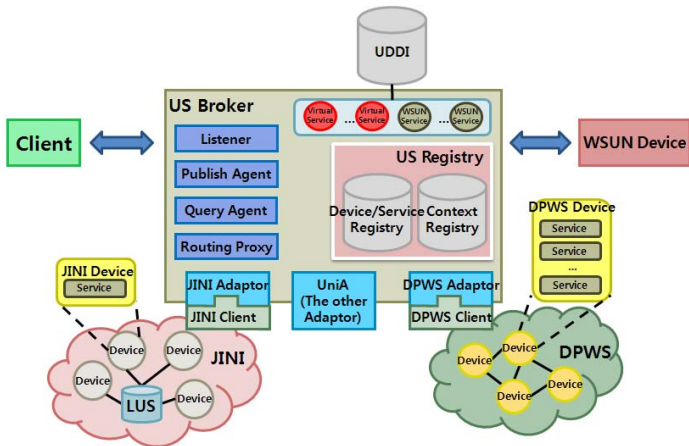


그림 3. WSUN의 아키텍처

US Broker는 SOA를 기반으로 하고 있다 클라이언트는 서비스를 사용할 때 US Broker를 이용하여 유비쿼터스 웹서비스를 검색하고 사용하게 된다 그리고 US Broker는 다음과 같은 모듈을 중심으로 구성된다

- 1) Query Agent: 서비스 검색을 위한 사용자 질의를 처리한다.
- 2) Publish Agent: US Registry에 해당하는 메타데이터의 저장을 담당한다.
- 3) Routing Proxy: 사용자가 원하는 서비스의 해당 서브네트워크와 연결 담당한다
- 4) Listener: 사용자와 WSUN 디바이스가 US Broker와 통신하기 위한 US Broker의 기본 정보 제공한다.
- 5) US Registry: Device/Service Registry는 서비스의 동적인 서비스 검색을 위하여 메타데이터의 공통 정적, 필수적인 요소를 저장하며 Context Registry는 디바이스의 위치 정보를 저장한다
- 6) UDDI : 웹서비스에 대한 정보를 저장하기 위한 레지스트리이다
- 7) Adaptor : 서브네트워크와 US Broker 사이의 중계 역할을 담당하며 서브네트워크의 서비스를 가상의 웹서비스로 변환한다

4. 지니 어댑터

지니 어댑터는 지니 클라이언트 Lookup Service, 웹 서비스 클라이언트를 확장한 것으로 다음과 같은 기능을 담당한다.

- 이벤트 처리
- 가상의 웹서비스 생성
- 메시지 변환
- 지니 서비스 검색

· 웹서비스 검색
지니 시스템에서 발생하는 다음과 같은 이벤트에 대하여 처리를 담당한다

- Hook-up
- 지니 서비스 등장
- 서비스 상태정보 및 기술정보변경
- 서비스 그룹 등록
- 서비스 그룹 정보 삭제
- Lookup Service의 퇴장
- 지니 서비스의 퇴장

Hook-up은 디자인 타임으로 지니 시스템이 동작하고 있는 상황에서 지니 어댑터가 등장하였을 때 처리하는 경우를 나타낸다. 지니 어댑터는 Lookup Service에 등록되어 있는 서비스를 검색하여, 필요한 정보를 레지스트리에 저장하게 된다. 나머지 이벤트는 런 타임시의 이벤트이며, 위의 이벤트를 처리하여 레지스트리에 반영하게 된다. 지니 서비스가 Lookup Service에 등록되게 되면 웹서비스를 생성하고 레지스트리에 정보를 저장하게 된다. 그리고 지니 서비스의 위치와 현재 서비스의 상태를 고려하여 서비스의 정보를 레지스트리에 업데이트를 하게 된다. 지니 서비스를 더 이상 제공하지 않으면 서비스에 대한 상태를 변경하게 되며 Lookup Service의 기능을 더 이상하지 않으면 등록되어 있던 서비스들에 상태를 변경하게 된다 또한 지니는 그룹정보를 제공하게 되는데, 그룹정보와 관련하여 등록 또는 삭제되는 서비스에 대하여 서비스의 정보를 등록하고 상태를 변경하는 과정을 거치게 된다. 이러한 이벤트를 처리함으로써 클라이언트에게 현재 사용이 가능한 서비스를 제공할 수 있게 된다.

그리고 등장한 지니 서비스에 대하여 가상의 웹서비스를 생성하는 단계를 거치게 되는데 다음과 같이 처리한다.

1. 지니 어댑터에서 지니 서비스와 관련한 소스파일을 요청한다.
2. 지니 어댑터에서 지니 서비스와 관련한 소스파일을 다운로드 받는다.
3. Axis를 이용하여 웹서비스를 생성한다
4. 웹서비스를 US레지스트리와 UDDI에 저장한다.

이와 같은 방식으로 생성된 WSDL 문서는 WSUN의 Device/Service Registry에 저장되며, UDDI에 Business Entity에는 WSUN, Business Service에는 서비스 이름을 저장한다.

이와 같은 형태로 인하여 <그림 4>과 같이 WSUN을

이용하여 지니 서비스를 검색하여 요청 및 사용을 할 수 있다.[5]

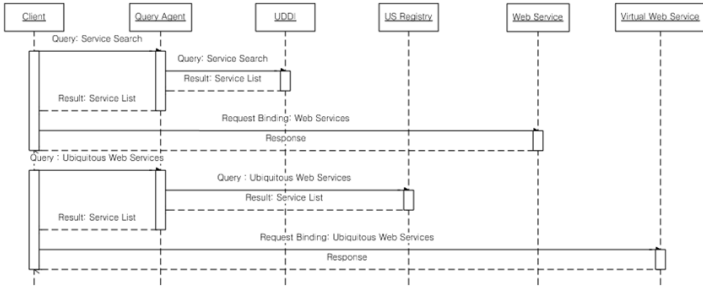


그림 4. WSUN을 이용한 서비스 검색 및 바인딩

클라이언트는 Query Agent를 통하여 UDDI에 등록된 웹서비스를 검색하게 되며, 웹서비스의 경우 바로 바인딩을 맺을 수 있다. 하지만 지니 서비스의 경우 서비스의 사용 가능 여부를 판단하기 위하여 다시 한 번의 쿼리를 거치게 되며, 사용가능한 경우 서비스와 바인딩을 맺게 된다. 바인딩을 할 경우, 웹서비스 클라이언트는 SOAP 메시지를 이용하지만 지니의 경우RMI의 메시지를 사용하여 서비스를 사용하게 된다

다음은 지니 클라이언트가 웹서비스를 사용을 하는 경우이다.

GOAL의 경우, UDDI에 등록되어있는 모든 서비스를 지니 서비스화를 하여 Lookup Service에 등록하게 된다. 하지만, 광대한 양의 웹서비스를 지니 서비스로 변경하기 위해서는 GOAL에서 제공하는 메타서비스들의 과부하가 심하다. 또한 휘발성의 메모리를 사용하는Lookup Service의 경우, 하드웨어가 재부팅되게 되면 모든 웹서비스에 대하여 다시 지니 서비스로 변환을 하여 재등록 시켜야하는 문제가 존재한다 또한 방대한 양의 웹서비스를 등록하기 위해서는 대용량의 Lookup Service가 필요로 하게 된다. 그리하여, 본 논문에서는 <그림 5>와 같이 웹서비스를 지니 서비스로 변환과정을 거치지 않고 지니 어댑터에 Lookup Service의 메시지를 처리하는 모듈과 웹서비스 클라이언트 기능을 추가하여 지니 클라이언트로부터 요청 메시지를 받게 되면 웹서비스 클라이언트를 통하여 웹서비스를 검색하고 사용하는 형태가 된다.

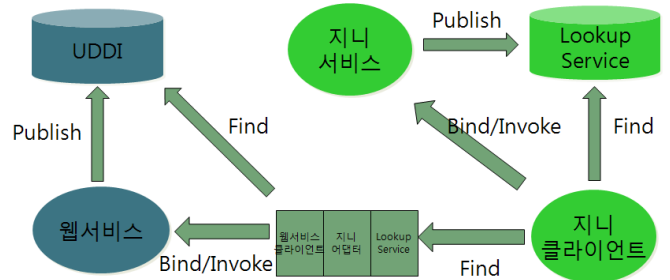


그림 5. 지니 클라이언트를 위한 지니 어댑터의 구성

기존의 지니 시스템과 웹서비스 환경이 존재하는 상황에서 Lookup Service를 확장하고 웹서비스 클라이언트 기능을 추가하여 서비스를 검색을 하게 된다

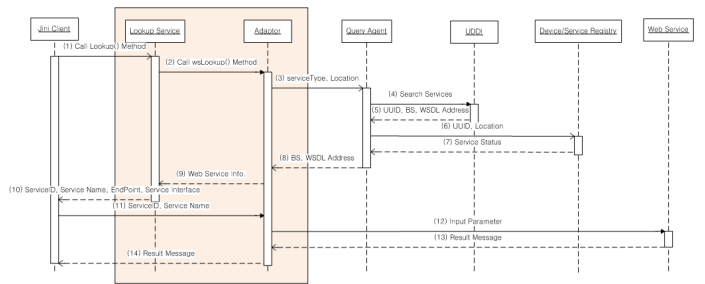


그림 6. 지니 클라이언트의 웹서비스 검색 및 바인딩

지니 클라이언트는 서비스를 검색하고 사용할 때 다음의 과정으로 처리한다

1. 지니 클라이언트는 어댑터를 찾는다
2. 어댑터에게 서비스 검색을 요청한다
3. 서비스에 대하여 어댑터에게 서비스와 요청한다
4. 지니 어댑터와 웹서비스 간에 연결된다

5. 결론 및 향후 연구

지금까지 지니 서비스를 웹서비스로 변환하여 등록 검색, 바인딩하는 것과 지니 클라이언트에서 웹서비스를 검색하고 바인딩 하는 과정을 설명하였다 이를 통하여 지니 클라이언트는 웹서비스를 지니 서비스로 변환되지 않은 상황에서 서비스를 검색하여 사용할 수 있다 또한 웹서비스 클라이언트에게 서비스의 상태와 장소를 고려하여 서비스를 제공할 수 있다 본 논문은 지니와 웹서비스 간의 상호운용을 위한 지니 어댑터를 중심이지만 다른 형태로 서비스를 기술하는DPWS, UPnP, Havi 등과 같은 서비스 디스커버리 미들웨어와 관련하여 이와 같은 어댑터를 생성하게 되면 웹서비스와 상호운용을 이룰 수 있다. 또한 이러한 서비스 디스커버리 미들웨어에서 제공하는 서비스들이 웹서비스로 변환이 됨으로써 모든 클라이언트에게 모든 서비스를 검색하여 사용이 가능

하게 된다.

참고 문헌

- [1] 임형준, 오일진, 황운영, 이경하, 이강찬, 이승윤, 이규철, “유니버설 네트워크를 위한 웹서비스 프레임워크 설계 및 구현” 정보과학회지, 2008. 4
- [2] Jan Newmarch, “Foundations of Jini 2 Programming” Apress, 2006
- [3] Graham Glass, “The Web Services (R)evolution - Applying Web Services to Applications”, <http://www.ibm.com/developerworks/webservices/library/ws-peer1.html>, 2000. 11
- [4] Franco Cicirelli, Angelo Furfaro, Libero Nigro, “Integration and Interoperability between Jini services and Web Services”, IEEE international Conference on Service Computing(SCC 2007), 2007. 10
- [5] Yun-Young Hwang , Il-Jin Oh, Hyung-Jun Im, Kyoung-Ha Lee, Kangchan Lee, Seungyun Lee, Kyu-Chul Lee, “UWS Broker for Ubiquitous Web Services Dynamic Discovery” , IPC-2007, 2007. 10
- [6] Web Services “www.w3.org/2002/ws/”