

# EJB 기반 Directory Web Service

김재철, 허태욱, 김성수, 김광수, 박종현, 이종훈  
한국전자통신연구원  
e-mail : kimjc@etri.re.kr

## Directory Web Service based on EJB

Jae-Chul Kim, Tae-Wook Heo, Sung-Soo Kim, Kwang-Soo Kim, Jong-Hyun Park,  
Jong-Hun Lee  
LBS Team, Spatial Information Technology Center, ETRI

### 요 약

본 논문은 Directory Service(white page, yellow page, green page)를 분산환경에 적합하고 시스템 유연성을 고려하기 위해서 EJB(Enterprise Java Bean)기반 Web Service 로 구현한 방법론 및 그 시스템에 관한 논문이다. 기존 개발된 Directory Service 가 대부분 폐쇄적 네트워크를 사용하고 있으므로 시스템 자체에 유연성이 없는 정적인 시스템으로 구현이 되어있고, 플랫폼에 의존하는 아키텍처로 설계되었다. 따라서 본 논문은 이러한 서비스 비개발 특성을 극복하기 위해서 개발 아키텍처를 웹서비스(Web Service)환경을 기본으로 하고 분산컴퓨팅환경(Distributed Computing Enviroment)의 특성을 고려하기 위해서 EJB 로 개발이 되었다.

### 1. 서론

현재 국내 많은 포털사이트(portal site)들이 Directory Service 를 제공하고 있으며 대부분이 웹페이지 형태의 제한된 서비스 구조로 되어있다. 이러한 기존의 서비스는 폐쇄적이면서도 소프트웨어가 단단히 묶여져 있는(tight couple) 고정적인 시스템이고, 개발 언어가 플랫폼 의존성함으로 인해서 외부의 이질적인 어플리케이션간의 통합을 제한하는 문제점을 내포하고 있다.

최근 XML 기반의 웹서비스가 기업과 개인, 기업과 기업 사이의 차세대 e-비즈니스로 급성장하고 있으며, 분산시스템의 통합의 대안으로 제시되고 있다. 이러한 웹서비스는 디바이스와 플랫폼에 제한을 받지 않으며 소프트웨어 구조가 유연한(loose couple) 구조를 가지므로 상호 데이터를 유연하게 통합해 준다[1].

본 논문은 기존 서비스의 폐쇄성과 플랫폼 의존성을 극복하기 위해서 웹서비스 기반 Directory Service 를 제안한다. 또한 개발모듈의 개선된 분산컴퓨팅환경을 구현하기 위해서 EJB 기반으로 시스템이 개발되었다.

표준들은 XML(Extensible Markup Language)[2], UDDI(Universal Description Discovery and Integration)[3], WSDL(Web Service Description Language)[4], SOAP(Simple Object Access Protocol)[5]등이 있다. 웹서비스의 기본적인 데이터를 정의하는 XML 스키마는 각 객체의 연관관계를 정의하여 이질적인 데이터의 상호 호환을 가능하게 해 준다. UDDI 는 웹서비스의 디렉토리 서비스에 해당하며 외부에서 웹서비스를 검색하는데 사용이 된다. WSDL 은 웹서비스의 서비스를 정의하는 언어로 프로그램이나 인터페이스를 정의 한다. SOAP 은 분산된 환경의 정보를 교환하는 통신 프로토콜로서 서비스 사용자가 정보를 교환할 수 있는 통신의 역할을 담당하고 있다. 현재 MS 와 IBM 주도로 표준이 진행되고 있다.

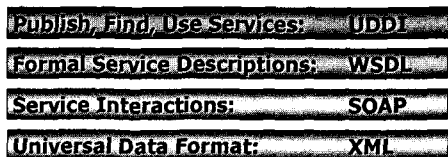


그림 1. 웹서비스의 기본 구조

### 2. 웹서비스(Web Service)

현재 W3C 가 제시하고 있는 웹서비스의 기본적인

### 3. 개발 시스템의 구조

#### 3.1 시스템 구성

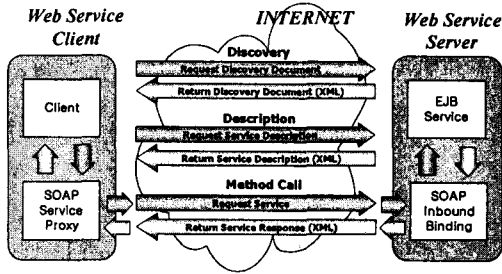


그림 2. 개발시스템의 웹서비스 흐름

개발 모듈의 전체 흐름도는 그림 2 와 같으며 웹서버에 Deploy 된 모듈은 EJB 에서 전개된 형태를 가진다. 개발에 사용된 개발도구는 WSAD(WebSphere Studio Application Developer) 5.0 이며, 웹서버는 WAS(WebSphere Application Server) 5.0 이 사용되었다.

개발 플랫폼의 인터페이스는 현재 개발 중인 서비스들과 연동 및 호환성을 고려하여 OpenLS(Open Location Service: <http://www.openls.org>)[6][7][8]에서 정의된 권고안(Spec. v.0.2)을 기준으로 작성되었다.

#### 3.2 요청(Request) 및 응답(Response)

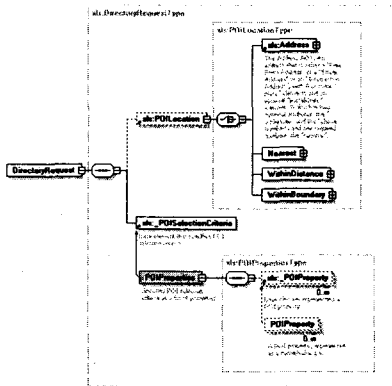


그림 3. Directory Request 의 스키마

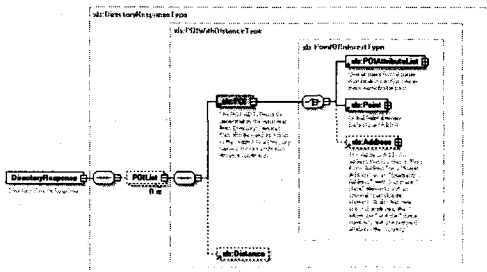


그림 4. Directory Response 의 스키마

Directory Service 에 대한 요청(Request)와 응답(Response)의 스키마 는 그림 3 과 그림 4 와 같으며 각각의 스키마는 Abstract Data Type, Geometry 를 포함하고 있다. 개발은 OpenLS 에 정의된 스키마를 Java Bena 으로 변환을 하고 각 변환된 Java Bean 을 객체로 사용하여 요청에 대한 응답을 하게 된다.

웹서비스에서는 기존의 XML 전송방식의 DOM 파싱(parsing)을 Proxy Class 가 담당하게 됨으로 추가적인 작업을 요구하지 않는다. 따라서 개발된 시스템의 최종 외부 노출 인터페이스는 하나의 WSDL 로 만들어지며 클라이언트는 이러한 WSDL 을 근거로 해서 코드를 작성하게 된다. 현재 IBM 과 MS 양측의 주도로 개발된 웹서비스에서 그 호환성이 부분적으로 문제가 되고 있다[9].

#### 3.3 Use-Case

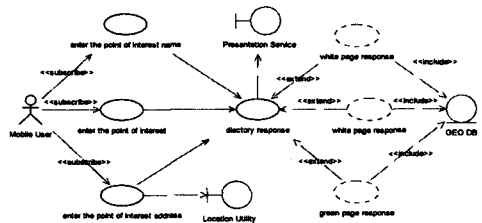


그림 5. 개발 시스템의 Use-Case 다이어그램

개발 시스템은 Directory Service 를 White Page Service, Yellow Page Service, Green Page Service 로 구분하였으며, Use-Case 는 그림 5 와 같이 구성이 된다. 이러한 개방형 서비스는 입력과 출력이 특정한 디바이스와 환경에 제한을 두지 않고 Client 가 Server 로부터 획득한 요청과 응답이 모두 기술된 WSDL 에 근거해서 유연하게 다른 모듈과 통합을 가능하게 해 준다.

그림 5 에서 사용자가 Mobile User 라고 가정했을 때 요청의 형태는 그림 3 의 요청 값에 대한 응답이 그림 4 의 응답 스키마에 맞게 작성되어, 그 결과 값이 리턴되게 된다. 이러한 동작이 가능하게 되는 것은 입출력에 대한 정의와 서비스 서버에 대한 정보 및 인코딩/ 디코딩 방법에 대한 내용이 WSDL 에 모두 정의되어 있기 때문이다.

Directory Service 를 위한 DB 에서 요청(Request)에 대한 응답(Response)를 구하기 위해서 공간연산이 필요하며 본 시스템에서는 Oracle 9i (SDO)가 사용이 되었으며, 사용된 DB 의 공간연산이 본 시스템의 거의 모든 공간연산을 지원하였다.

질의 속성으로 사용되어야 할 POIAttributeList 에서 Referencetype 이 미국 분류체계를 기준으로 만들어졌으므로 스키마를 국내 실정에 맞는 분류체계로 수정하였다. 스키마 내부에 정의된 추상(abstract)화 된 부분은 본 OpenLS spec. v.0.2 의 특성을 고려하여 재정의 를 통해서 DirectoryRequestType 과

DirectoryResponseType 이 재정의 되었다.

국내에서 Directory Service 를 제공하기 위해서는 OpenLS 스펙만으로는 이상과 같은 문제점 및 주소체계와 같은 문제점도 발생함으로 국내 표준안의 도출이 시급하다.

### 3.4 EJB 기반 개발 컴포넌트

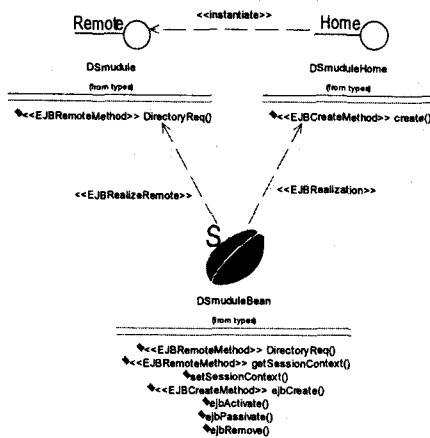


그림 6. EJB 다이어그램

EJB 는 JAVA 언어로 작성되는 분산 객체 형태의 비즈니스 로직을 구현하는 어플리케이션을 개발하기 위한 표준 컴포넌트이며, 한가지 틀에 종속적인 것이 아니라, 여러가지 틀들을 이용할 수 있으며 틀들을 통하여 컴포넌트를 조합하여 어플리케이션을 작성할 수 있다. 트랜잭션, 보안, 패일오버, 멀티쓰레드, 커넥션풀링 등과 같은 기능들을 개발자가 고려하지 않아도 개발이 가능하다. 그리고 EJB 컴포넌트를 개발하는 단계와 EJB 컨테이너에 배치하고 실행하는 단계는 명확하게 구분되며 EJB 어플리케이션 기존의 JAVA API 등을 이용할 수도 있으며 다른 언어로 개발된 컴포넌트와 함께 동작할 수 있다[10].

이러한 EJB 의 특성을 고려해 볼 때, 개발 모듈이 개방형 시스템과 분산환경을 목표로 설계되었으므로 이러한 EJB 의 장점을 수용하였다. 단순히 개발 모듈을 JAVA Bean 으로 개발하여 Web Service 로 전개시킬 경우 분산환경 지원 및 컴포넌트 재사용이 어렵게 된다. 따라서 개발 시스템은 그림 6 과 같이 EJB 형태의 컴포넌트로 개발이 되었다.

그림 7 은 개발된 EJB 로부터 웹서비스로 전개하여 작성된 WSDL 을 도식화한 것으로 서비스 클라이언트 작성시 참조된다. 실제 이러한 웹서비스는 UDDI 서버를 통하여 서비스를 찾고 찾아진 서비스로부터 WSDL 문서를 받음으로 해서 구현이 가능하다.

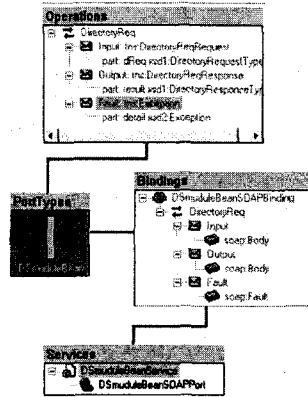


그림 7. 서비스 WSDL 의 다이어그램

### 4. 실험 결과

사용된 POI 는 서울, 경기지역의 388,576 개의 POI 이며 디렉토리 구분은 대분류, 중분류, 소분류의 세 단계로 나누었다. 그림 8 에서 붉은 점은 실험에서 사용된 POI 정보이며 배경의 경계는 시·군 행정경계를 나타낸다.

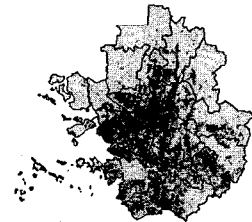


그림 8. 실험에 사용된 POI

아래의 XML 문서는 테스트 질의 문서로서 '서울특별시 종로구청'에서 가장 가까운 우체국을 웹서비스로 질의를 한 경우이다. 앞 절에서 기술한 바와 같이 POIProperty 의 ReferenceSystem 대한 국내 표준의 부재로 인하여 테스트 시스템은 NAICS(North American Industry Classification System)를 임의로 구분하여 사용하였다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<Request xmlns="http://www.opengis.net/xds" methodName="DirectoryRequest"
version="1.0.3" requestID="UseCase3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/xd
C:\OpenLS\Schema\DirectoryService.xsd">
  <DirectoryRequest>
    <POILocation>
      <Nearest>
        <POI ID="1">
          <POIAttributeList>
            <POIInfoList>
              <POIInfo name="POI Name" value="종로구청"/>
            </POIInfoList>
            <POIAttributeList>
          </POI>
        </Nearest>
      </POILocation>
      <POIProperties directoryType="Yellow Pages">
        <POIProperty name="NAICS_type" value="공공편의"/>
        <POIProperty name="NAICS_subType" value="공공기관"/>
        <POIProperty name="NAICS_category" value="우체국"/>
      </POIProperties>
    </DirectoryRequest>
  </Request>
```

위의 질의로부터 응답은 아래의 XML 형태의 결과 값이 나오게 되며 실제 질의(요청)는 본 시스템 내에서 JAVA Bean 에서 만들어진 DOM 구조로 생성되게 된다. 하지만 기본 스키마 파일로부터 만든 JAVA Bean 에 오류가 있을 경우에는 정확한 요청과 응답이 이루어 질 수 없다. 그리고 자바로 개발된 웹서비스의 경우 응답값이 자바에서만 지원하는 리턴값을 리턴할 경우에는 .NET 웹서비스에서는 프록시 클래스를 생성할 수 없게 된다. 이러한 문제는 현재 W3C 에서 논의가 되고 있으며 향후 해결해야 할 문제이다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<xs:DirectoryResponse xmlns:xs="http://www.opengis.net/xd"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:schemaLocation="http://www.opengis.net/xd
C:\OpenLS\Schema\DirectoryService.xsd">
  <xs:POIList>
    <xs:POI ID="1">
      <xs:POIAttributeList>
        <xs:ReferenceSystem>
          <xs:NAICS category="공공문의" subType="공공기관" type="우체국">
            </xs:NAICS>
          </xs:ReferenceSystem>
          <xs:POIInfoList>
            <xs:POIInfo name="POI Name" value="서울역탄회관우편취급소">
              </xs:POIInfoList>
            </xs:POIInfoList>
            <xs:POIAttributeList>
              <xs:Point>
                <xs:Coord ls="198197.99251 452266.22942"></xs:Coord>
              </xs:Point>
              <xs:Address>
                <xs:freeFormAddress>서울시 종로구 수송동 80-8
              </xs:freeFormAddress>
            </xs:Address>
          </xs:POI>
          <xs:Distance accuracy="1" distanceUnits="KM" value="0.09523"></xs:Distance>
        </xs:POIList>
      </xs:DirectoryResponse>
```

5. 결론 및 향후 연구과제

본 논문에서는 기존 서비스의 폐쇄성과 플랫폼 의존성을 극복하고 분산컴퓨팅환경 특성을 고려한 EJB 기반 Directory Web Service 를 제안하였다. 제안된 웹서비스 기반 시스템 구조가 개발된 서버의 개발 언어에 대해서 클라이언트의 개발언어가 제한되어야 한다는 단점을 극복할 수 있었고, 개방형 시스템의 구현이라는 서비스 성능과 효율이 증대되었다. 하지만 본 시스템의 개발 언어인 Java 가 플랫폼 의존하지 않고 수행이 가능하다는 장점이 있지만 실험결과 .NET 으로 구현된 모듈에 비해서 속도가 느려지는 경향이 있었다. 이러한 문제는 접속 사용자의 수와 병렬구조의 서버를 고려한 추가적인 실험을 통해 정밀한 검증이 필요할 것으로 고려된다.

향후 연구과제로 OpenLS 에서 정의된 권고안에 포함되지 않은 국내 주소체계에 대한 스키마와 디렉토리 분류체계 스키마의 정의를 통해 개선된 시스템을 구현하고자 한다.

참고문헌

[1] Scott Cable, Ben Galbraith, Romin Irani. *Professional Java Web Services*. Wrox Press Inc. 2002. ISBN: 1861003757  
 [2] David Hunter, Kurt Cagle, Chris Dix. *Beginning XML*. Wrox Press Inc. 2001. ISBN: 0764543946  
 [3] Universal Description, Discovery and Integration.

http://www.uddi.org  
 [4] Technical Report: Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl  
 [5] Technical Report: SOAP Version 1.2 Working Draft. http://www.w3.org/TR/soap12  
 [6] OpenLS Directory Service Specification, OpenGIS© Project Document (OGC 02-094), Open GIS Consortium Inc., 11 November 2002.  
 [7] OpenLS Location Utility Service, OpenGIS© Project Document (OGC 02-092), Open GIS Consortium Inc., 14 November 2002.  
 [8] OpenLS Presentation Service, OpenGIS© Project Document (OGC 02-091), Open GIS Consortium Inc., 14 November 2002.  
 [9] 정부연. “웹서비스의 개념과 관련 기업에 미치는 영향”, 정보통신정책 제14권 7호, 2002, pp. 23-37.  
 [10] P G Sarang, Kyle Gabhart, Andre Tost. *Professional EJB*. Wrox Press Inc. 2001. ISBN: 1861005083