

웹서비스 기반 개방형 핵심공통기술의 설계 및 구현

김재철, 박종현, 이종훈
한국전자통신연구원 공간정보기술센터

Architectures and Implementation of OpenLS Core Service based on Web-Service

Jaechul Kim, Jonghyun Park, Jonghun Lee
Spatial Information Technology Center, ETRI
E-mail : kimjc@etri.re.kr

Abstract

In this paper, we developed those OpenLS(Open Location Service) Core Services based on the Web-Service, because Web-Service environments provide a suitable method to gather requested information in an appropriate way. The proposed architecture cooperate with OpenLS Core Services(Directory Service, Location Utility Service and Router Determination service) and is an interoperability one — it identifies those global elements of the global Web-Services network that are required in order to ensure interoperability between Web-Services. In this paper, a new architecture of OpenLS Core Services is proposed and tested in OpenLS Core Services environments.

I. 서론

현재 국내 많은 포털사이트들이 디렉토리, 맵, 라우터 서비스를 제공하고 있으며 대부분이 웹페이지 형태의 제한된 서비스 구조로 되어있다. 이러한 기존의 서비스는 폐쇄적이면서도 소프트웨어가 단단히 묶여져 있는(tightly couple) 고정적인 시스템이고, 개발 언어가 플랫폼 의존성함으로 인해서 외부의 이질적인 어플리케이션간의 통합을 제한하는 문제점을 내포하고 있다.

이러한 문제를 해결하기 위해 최근 XML 기반의 웹서비스가 기업과 개인, 기업과 기업 사이의 차세대 e-비즈니스로 급성장하고 있으며, 분산시스템의 통합의 대안으로 제시되고 있다. 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 OpenLS Core Services를 제안한다. 또한 개발자들의 개선된 분산컴퓨팅환경을 구현하기 위해서 EJB 기반으로 시스템이 개발되었다.

1. 웹서비스(Web Service)

현재 W3C 가 제시하고 있는 웹서비스의 기본적인 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.

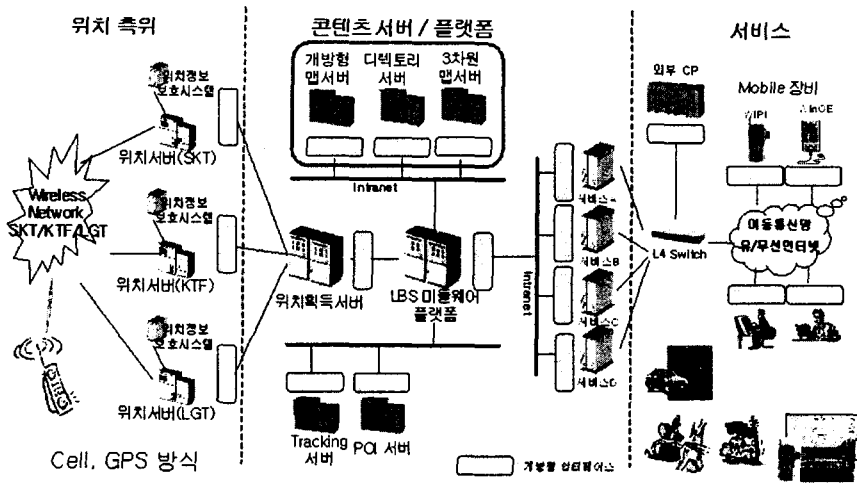


그림 1. 개방형 LBS 연계기술

II. 개발 시스템의 구조

2.1 시스템 구성

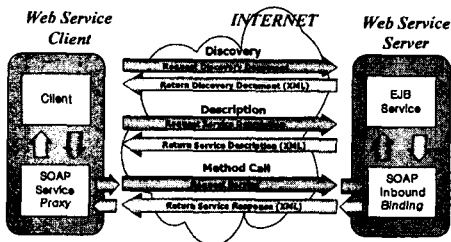


그림 2. 개발시스템의 웹서비스 흐름

그림 1 에서와 같이 유·무선 사용자가 핵심공통 모듈 (OpenLS Core Service)를 요청하기 위해서 플랫폼에 개방형 인터페이스(웹서비스)를 통하여 요청을 하면 플랫폼이 해당 요청에 대하여 서비스를 위한 인증을 하고 나서 핵심공통 모듈에 서비스 요청을 하고 그 결과를 사용자에게 전송하게 된다. 여기에서 사용자와 플랫폼 그리고 핵심공통 모듈은 웹서비스를 통하여 요청과 응답이 이루어지며, 서비스에 대한 명세는 플랫폼이 WSDL을 UDDI 서버에 등록함으로써 사용자가 서비스를 요청할 수 있게 된다.

핵심공통 모듈의 전체 흐름도는 그림 2와 같으며 웹 서버에 Deploy 된 모듈은 EJB 에서 전개된 형태를 가진다. 개발에 사용된 개발도구는 WSAD(Websphere Studio

Application Developer) 5.0이며, 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.opensls.org>)[6][7][8]에서 정의된 권고안(Spec. v.0.2)을 기준으로 작성되었다.

2.2 요청(Request) 및 응답(Response)

핵심공통 모듈의 하나인 Directory Service 에 대한 요청(Request)와 응답(Response)의 스키마는 그림 3 과 그림 4 와 같으며 각각의 스키마는 기본 Abstract Data Type 을 포함하고 있다.

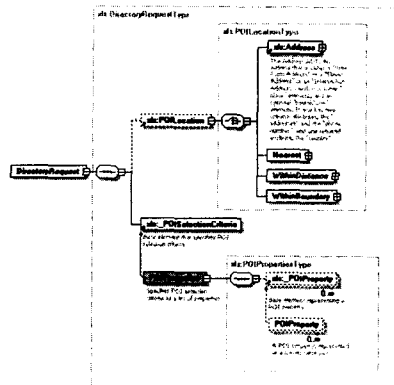


그림 3. Directory Request 의 스키마

III. 실험 결과

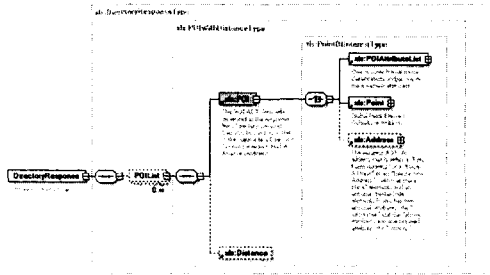


그림 4. DirectoryResponse의 스키마

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

2.3 EJB 기반 개발 컴포넌트

EJB는 JAVA언어로 작성되는 분산 객체 형태의 비즈니스 로직을 구현하는 어플리케이션을 개발하기 위한 표준 컴포넌트이며, 한가지 볼에 종속적인 것이 아니라, 여러가지 볼들을 이용할 수 있으며 볼들을 통하여 컴포넌트를 조합하여 어플리케이션을 작성할 수 있다. 트랜잭션, 보안, 패일오버, 멀티스레드, 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 EJB의 특성을 고려해 볼 때, 개발 모듈이 개방형 시스템과 분산환경을 목표로 설계되었으므로 이러한 EJB의 장점을 수용하였다. 단순히 개발 모듈을 JAVA Bean으로 개발하여 Web Service로 전개시킬 경우 분산환경 지원 및 컴포넌트 재사용이 어렵게 된다. 따라서 개발 시스템은 EJB 형태의 컴포넌트로 개발이 되었다.

```
<?xml version="1.0" encoding="EUC-KR" ?>
<DirectoryRequest >
  <POILocation>
    <Nearest>
      <Position>
        <Point>
          <Coord ts=" " decimal="." cs=",">198393.007,452232.309</Coord>
        </Point>
      </Position>
    </Nearest>
  </POILocation>
  <POIProperties directoryType="Yellow Pages">
    <POIProperty>
      <KSICS category="패스트푸드" subType="음식점" type="생활편의"/>
    </POIProperty>
  </POIProperties>
</DirectoryRequest>
```

위의 질의로부터 응답은 아래의 XML 형태의 결과값이 나오게 되며 실제 질의(요청)는 본 시스템 내에서 JAVA Bean에서 만들어진 DOM 구조로 생성되게 된다. 핵심공통 모듈의 입출력은 OpenLS v.02에 정의된 인터페이스를 기본으로 하였다.

```
<?xml version="1.0" encoding="EUC-KR" ?>
<DirectoryResponse>
  <POIList>
    <POI ID="1" POIName="KFC공평동매점">
      <Point>
        <Coord cs="," decimal="." ts=" " >198393.00,452212.30</Coord>
      </Point>
      <Distance distanceUnits="M" value="22.360295180477"/>
    </POIList>
  </DirectoryResponse>
```

본 논문에서는 핵심공통 모듈의 성능평가를 위해서 KTF WIPI 에뮬레이터로 테스트하였다. WIPI 자체에서 XML 처리는 독립적인 Gate-Way 서버를 통하여 실행하고, WIPI 에뮬레이터에서는 HTTP 방식을 이용하여 Gate-Way 서버를 통하여 개방형 플랫폼에 서비스를 요청하게 된다.

그림 5는 에뮬레이터에서 현재의 위치를 프리젠테이션 서비스를 요청하고, 위에서 언급한 현재위치에서 가장 가까운 패스트푸드점을 찾는 것을 도시화한 것이다.

프리젠테이션 서비스와 연계된 디렉토리서비스의 요청

참고문헌

및 응답 예:

- (1) 현재위치에 대한 지도 및 위치정보 요청
- (2) 서비스 요청 기준점 설정
- (3) 디렉토리 카테고리 설정
- (4) 최단거리/반경/화면영역 설정
- (5) 요청결과 확인
- (6) 요청결과에 대한 위치정보 표시

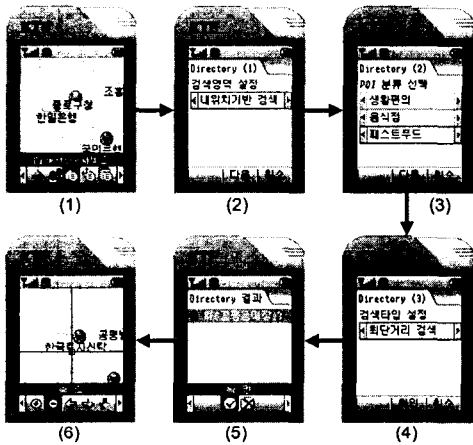


그림 5. 디렉토리 서비스 요청

- [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 제 14 권 7 호, 2002, pp. 23-37.
- [10] P G Sarang, Kyle Gabhart, Andre Tost. Professional EJB. Wrox Press Inc. 2001. ISBN: 1861005083

IV. 결론 및 향후 연구과제

본 논문에서는 기존 서비스의 폐쇄성과 플랫폼 의존성을 극복하고 분산컴퓨팅환경 특성을 고려한 EJB 기반을 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 Java가 플랫폼 의존하지 않고 수행이 가능하다는 장점이 있지만 실험결과 .NET으로 구현된 모듈에 비해서 속도가 느려지는 경향이 있었다. 이러한 문제는 접속 사용자의 수와 병렬구조의 서버를 고려한 추가적인 실험을 통해 정밀한 검증이 필요할 것으로 고려된다.