OpenLS Directory Service Architectures and Implementation based on Web-Service

Jae-Chul Kim, Sung-Soo Kim, Tae-Wook Heo, Jong-Hyun Park
LBS Team, Spatial Information Technology Center, ETRI
161 Kajong-Dong, Yusong-Gu, Daejon, Korea
kimjc@etri.re.kr

Abstract: In this paper, we developed those Directory 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 other OpenLS (Open Location Service) Core Services (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 Directory Service with OpenLS Core Services is proposed and tested in OpenLS Core Services environments.

Keywords: Web-Service, Directory service, OpenLS.

1. Introduction

There is a need to further structure large applications into building modules in order to use well-defined components within different business processes. A shift towards a service-oriented approach will not only standardize interaction, but also allows for more flexibility in the process. A service-oriented architecture thus has to focus on how services are described and organized to support their dynamic, automated discovery and use. If the services become more and more complex, the basic mechanism of request-response is hardly applicative anymore. A couple of services include mid or even long term (trans-) actions that demand a functionality to establish an asynchronous communication between a user and the corresponding service, or two services respectively. The Web Service is self-contained, modular applications that can be described, published, located, and invoked over a network. Web-Services perform encapsulated business functions, ranging from simple request-reply to full business process interactions [1].

The proposed architecture is proposed a new Directory Service based on Web-Service to overcome the platform dependency, closed system characteristics, and distributed computing environment.

2. Web-Service

In present, W3C proposed the Web-Service standard as follow :

XML (eXtensible Markup Language) is the markup language that underlies most of the specifications used for Web-Services. XML is a generic language that can be used to describe any kind of content in a structured way, separated from its presentation to a specific device [2].

SOAP (Simple Object Access Protocol) is a network, transport, and programming language-neutral protocol that allows a client to call a remote service. The message format is XML [3].

WSDL (Web-services description language) is an XML-based interface and implementation description language. The service provider uses a WSDL document in order to specify the operations a Web-Service provides, as well as the parameters and data types of these operations. A WSDL document also contains the service access information [4].

UDDI (universal description, discovery, and integration) is both a client-side API and a SOAP-based server implementation that can be used to store and retrieve information on service providers and Web-Services [5].

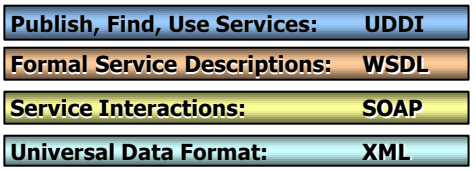

Fig. 1. Structure of Web-Service.

3. Architecture of the developed system

1) System overview

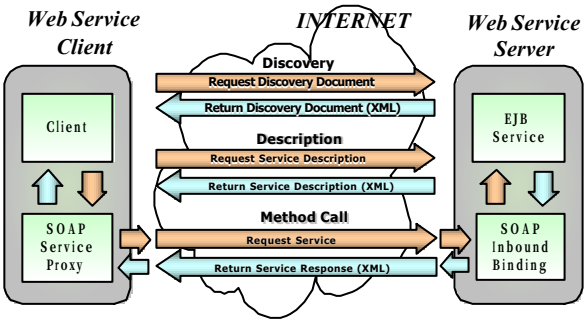

Fig. 2. System architecture.

As shown in Fig. 2, the proposed system consists of Web-Service server and Web-Service client developed from EJB (Enterprise Java Beans).

A developing tool is WSAD(Websphere Studio Appli-

cation Developer) 5.0 and WAS(Websphere Application Server) 5.0 is used in web server.

The interface of service platform considering interoperability is implemented on the basis of OpenLS(Open Location Service:http://www.openls.org) [6]-[8] Spec. v.o.2.

2) Directory request and response

The definition of request/response pairs defined below will encompass the requirements for both Pinpoint and Proximity Directory Services. The two usages of the Directory Service will be affected by selecting the appropriate optional parameters in the XML schema. The Pinpoint query will use parameters that will uniquely identify the Point of Interest. At least one of the following options may be used to identify the target location. The Proximity query will use parameters that will identify a shortlist of candidate locations by distance from a location that may itself be specified by a Pinpoint query. This may involve a combination of Proximity parameters to spatially constrain the search.

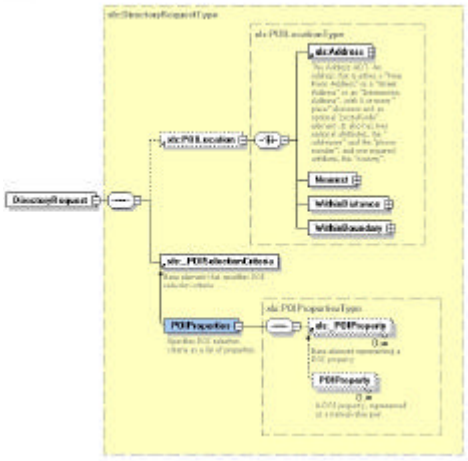

Fig. 3. Schema of directory request

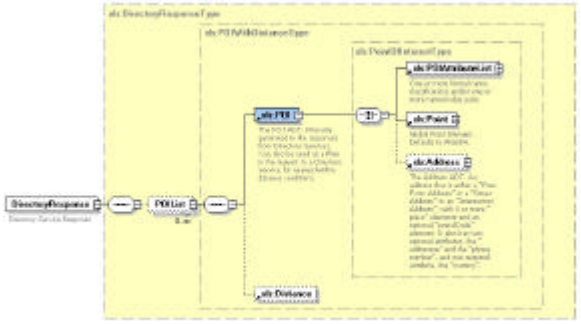

Fig. 4. Schema of directory response

The schema of request and response on Directory Service is described in Fig. 3 and Fig. 4. And each schema includes Abstract Data Type, Geometry. In development, skeleton class (Java Beans) is made from OpenLS Directory Service schema and we can access requests and responses using those classes. In Web-Service Implementation, DOM (Document Object Model) parsing contrast to previous XML transfer method depends on the proxy class. Thus an additional process is not required.

In consequence, the published interface of processing module is made in a WDSL document, and the service client makes the proxy interface on the basis of the published WDSL document. Currently, Web-Service developed between the two opposing sides (IBM and Microsoft) has a problem that its interoperability can not be applied [9].

3) Use-case

The user of the Directory Service wishes to locate a particular Place, Product or Service. The user may constrain the request by specifying parameters that filter out candidate places/products/services according to some identifier, attribute, or location. Use-case for Directory Services is displayed below (Fig. 5).

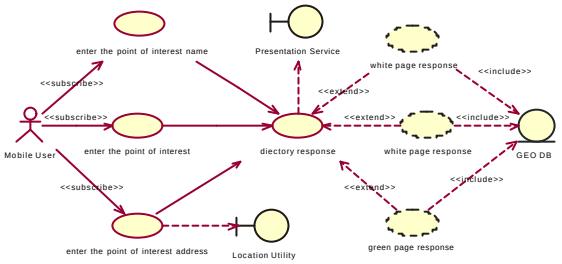

Fig. 5. Use-case Diagram

We need a spatial operation to execute the Directory Service that searches for nearest POIs, within distance, within boundary. In this system, Oracle 9i (SDO) is used and it assists with Directory Service spatial operation to search the interest point.

As POIAttributeList in OpenLS directory Service Schema is made on the basis of NAICS (North American Industry Classification System), we defined a new Korea Standard Industry Classification System.

4) System architecture based on EJB component

The Enterprise JavaBeans architecture is component architecture for the development and deployment of component-based distributed business applications. Applications written using the Enterprise JavaBeans architecture is scalable, transactional, and multi-user secure. These applications may be written once, and then deployed on any server platform that supports the Enterprise JavaBeans specification [9]. In brief, EJB is designed to address issues involved with managing distrib-

uted business objects in three-tier architecture.

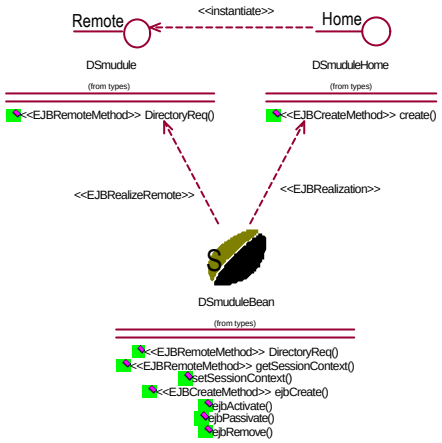

Fig. 6. EJB component diagram

The system architecture developed from Java Bean can not be reused on distributed environment and other component module. For this reason, the proposed architecture has an EJB structure (Fig. 6). Fig. 7 is directory Web-Service WDSL diagram and this WSDL document is referenced on deploying eb-service. As a matter of fact, it is possible WSDL document to be searched in UDDI server and Web-Service to publish the WDSL document.

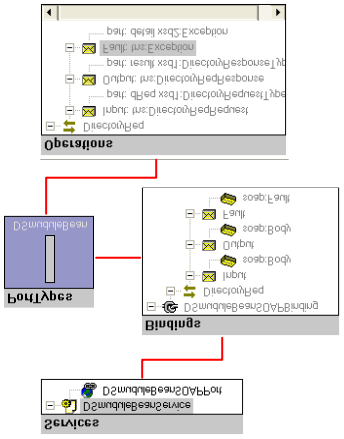

Fig. 7. Directory Service WSDL diagram.

4. Conclusion

In this paper, Directory Web-Service based on EJB is proposed to overcome a platform dependency and to enhance the distributed computing performance.

The proposed Directory Web-Service is independent of a server platform. And due to the Web-Service system architecture, client is not limited to program languages. However Java based Web-Service is a little slower than .NET Web-Service in test-bed. More simulation is need in parallel computing environment and multi-user request. In future works, we will consider the Korea Standard Industry Classification and test the system performance in the various environments (OS).

```
<?xml version="1.0" encoding="euc-kr"?>
<DirectoryRequest xmlns="http://www.opengis.net/xls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/xls DirectoryService.xsd">
    <xls:POILocation>
        <xls:WithinDistance>
            <xls:Position>
                <xls:Point>
                    <xls:Coord ts="198102.79 452263.92"/>
                </xls:Point>
            </xls:Position>
            <xls:MaximumDistance value="300"/>
        </xls:WithinDistance>
    </xls:POILocation>
    <xls:POIProperties>
        <xls:POIProperty>
            <xls:KSICSType category="Post Office" subType="Public Agency"
type=" Public Service "/>
        </xls:POIProperty>
    </xls:POIProperties>
</DirectoryRequest>
```

Fig. 8. Directory Request use-case.

```
<?xml version="1.0" encoding="euc-kr"?>
<DirectoryResponse xmlns="http://www.opengis.net/xls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/xls DirectoryService.xsd">
    <xls:POIList>
        <xls:POI ID="2" POIName="JongNo 2 Ga PostOffice ">
            <xls:Point>
                <xls:Coord ts="198338.49719 452482.63754"/>
            </xls:Point>
        </xls:POI>
        <xls:Distance distanceUnits="M" value="321.551304959814"/>
    </xls:POIList>
    <xls:POIList>
        <xls:POI ID="3" POIName="KwangHwaMoon PostOffice">
            <xls:Point>
                <xls:Coord ts="197981.06303 451950.5568"/>
            </xls:Point>
        </xls:POI>
        <xls:Distance distanceUnits="M" value="336.175475517775"/>
    </xls:POIList>
</DirectoryResponse>
```

Fig. 9. Directory Response use-case.

References

[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] Technical Report: SOAP Version 1.2 Working Draft. http://www.w3.org/TR/soap12

[4] Technical Report: Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl

[5] Universal Description, Discovery and Integration. http://www.uddi.org

[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] Sun Microsystems . Enterprise JavaBeans™ Specification, v2.0, Copyright 2001 by Sun Microsystems, Inc.