Implementation of OGC Web Map Service Based on Web Service

JIA Wenjue CHEN Yumin GONG Jianya

ABSTRACT OGC Web Map Service is one kind of OGC Portrayal Services belongs to OGC Web Service model and it provides multi-platform interoperability of spatial data set. This paper presents a method for implementing OGC Web Map Service based on Web Service technique and introduces the detailed process.

KEYWORDS Web Service; OGC OWS; Web Map Service; WMS

CLC NUMBER P208

Introduction

The most important advantage of web service is the multi-platform, multi-architecture and multi-program-language interoperability on Web. Therefore, it is a good approach to realizing GIS interoperability.

Open GIS Consortium (OGC) is one of standard organizations in the field of spatial information. In the latest decade, researchers have made great efforts in seeking a way to effectively integrate GIS with distributed processing technology, object oriented technology, database design and real-time information processing method, and realizing the interoperability on distributed heterogeneous platform. The emergence of the web service technology overcame the limits of traditional distributed object technology, and furnished a new chance for GIS interoperability over the Internet. OGC Web Services Initiative project presented a framework of web-service-based Spatial Information Service, and a series of abstract specifications and implementation specifications. OGC Web map service is one kind of Portrayal service in the framework of OGC Spatial Information Service.

1 OGC Web Service model and Web Map Service

1.1 OGC Web service model

OGC Web services Initiative is a research project that studies how to solve the problem of GIS interoperability with Web service technology and related techniques. The aim is to put forward an extensible framework, which is based on various standards and can integrate kinds of online spatial processing and location service seamlessly, i.e. OGC Web services (OWS). By XML, OWS enables the distributed spatial processing system to interact with the HTTP technique, and provides a framework of interoperability for the web-based discovery, access, combination, analysis, exploitation and visualization of various online spatial data resources, the information from sensor, spatial processing service and location service.

The basic services in OWS include Application Clients, Registry Services and Encoding, Data Services, Portrayal Services and Processing Services. Fig. 1 illustrates the OWS service...
framework. In OWS, portrayal service provides presentation manners of spatial data, and the access capability to data sets. The Web map service (WMS) is a kind of portrayal services.

1.2 OGC WMS

OGC WMS is one of OGC Portrayal Services. It produces maps of geo-referred data. In the specification of WMS, a map is defined as the visual presentation of the geographical data. The outcome generated by WMS is not map data, but the image of the map. There are three operations defined in this specification: GetCapabilities returns meta-data on service level. The meta-data, expressed in XML, is the description of the content of service information and the required parameters; GetMap returns a map image according to the parameters in the request. The geographical parameters and the size parameters are explicitly defined. The returned map image may be in the format of GIF, JPEG, PNG or SVG; GetFeatureInfo returns the information of some special elements on the map. This operation is optional. Fig. 2 is the workflow of WMS operations.

2 Characteristics and architecture of Web Service technology

2.1 Understanding of Web Service technology

Web service is to establish a new platform for interoperable distributed applications. In fact, web service is a suite of standards which define how applications interoperate through the web. Some key techniques of Web service, such as SOAP, WSDL (Web service description language) and UDDI (universal description discovery and integration), endow web service with natural capability of cross-platform interoperation. Web service is a self-contained, self-described and modularized application, and it can be discovered, located and invoked through the web.

2.2 Characteristics of Web Service technology

In addition to the natural capability of cross-platform interoperation, the service also has following characteristics:

- sharing both program and data
- easy to be integrated with other program
2.3 Web service architecture

Web Service architecture describes a framework, in which sorts of services are dynamically described, published, discovered and invoked. This architecture defines the standardized operations of that how service provider, service requester and service registration center interact with one another. Finding the appropriate service, we can describe the file invocation service by the WSDL offered by the service registration center. Fig. 3 presents the Web service architecture.

3 Web-service-based method for the implementation of WMS

From the Web service architecture, we can see that service is the core, because all operations such as publishing, searching, and binding must take service as the foremost concern. OGC WMS is an instance of service. There are three steps to implement WMS:

1. Create WMS application;
2. Deploy WMS application, and enable it to provide service;
3. Publish WMS to registration center for service user to search;

Next, we will analyze every step of creating, deploying and publishing WMS and the key techniques in detail, and validate them by implementing a client instance.

3.1 Create WMS application

The lifecycle of web service defines four different solutions to implement Web service. When implementing the OGC WMS, due to the explicit interface definitions of WMS, we can adopt the Top-to-Bottom solution. From existent descriptions of web service interface, this solution creates applications that can implement the above-mentioned interface.

From the view of technical implementation, we can regard web service as an application, which exposes interfaces that can be invoked through web, and allows itself to be called by any program that runs on any platform, any system and is coded by any language. This application can be implemented with all sorts of existent programming language. Our WMS implements the defined operation interfaces in OGC WMS specification. To create WMS application, the most important thing is to implement the operation interfaces and make it satisfy the function requirements of the operation interfaces. Using GIS components to implement WMS application can greatly reduce the workload of development, and it is helpful for code reusability. We employed the components of GeoSurf and GeoStar to implement OGC WMS under Java and .NET platform, respectively. By choosing different platform and language for software development, we validate the cross-platform interoperation capability of WMS.

The architecture of WMS implementation is shown in Fig. 4. The system contains three parts: client, application server and data server. On the application server, WMS on Java and .NET platform calls GeoSurf and GeoStar components to implement WMS interfaces, access data server and file server through these components, and then obtain the spatial data.

3.2 Approach to WMS deployment

The difference between web service and other applications is that web service assures the characteristic of cross-platform interoperability by several standard protocols SOAP, WSDL and UDDI. Therefore, how to deploy, describe, transmit and register a web service is the crux in its implementation. After finishing the develop-
To implement WMS application, we need to apply it on application servers, and make it provide service by the standards of SOAP, WSDL. Since SOAP, WSDL and UDDI are standards, different provider may implement different products with them. For instance, many companies like SUN, APACHE, IBM and Borland etc. have presented Java-based web service packages one after the other. .NET platform presented by Microsoft is also a prominent example. To implement web service, all of them provide tools that do not depend on providers so that the complexity in developing web service is greatly decreased. .NET platform provides an integrated development environment for developing the web service application. It is easy to deploy service over the .NET platform. Taking APACHE AXIS as an example, we will introduce the WMS-deployment process on the Java platform as follows.

APACHE AXIS is the third generation SOAP server of APACHE with very good performance. It is coded by Java language and fully opens the source code, which can be downloaded from the website (http://www.apache.org). AXIS is not able to work independently. Before deploying, publishing WMS by AXIS, we need to configure it on some application server (like Tomcat). The configuring process is as follows (take Tomcat as an example); first, add all jar files under the directory \webapps\axis\WEB-INF\lib into the classpath of Tomcat, then set the jar files of AXIS to the system classpath; Launch Tomcat after above configuration, and open the test paper http://localhost:8080/axis to confirm whether the configuration is successful or not.

To deploy WMS application on the AXIS server, we also need a deployment file that provides a mechanism AXIS server for establishing a mapping between the name and the implementation of web service. The deployment file is described as follows.

```xml
<deployment
    xmlns="http://xml.apache.org/axis/wsdd/
    xmlns:ns="http://wms"
    xmlns:java="http://xml.apache.org/axis/wsdd/providers/java" >
  <service name="WMSBeanService" provider="java:RPC">
    <parameter name="className" value="wins.WMSBean"/>
    <parameter name="allowedMethods" value="*"/>
  </service>
</deployment>
```

Once WMS is deployed on the AXIS server, it can be called by other applications. Web service describes its concrete content by WSDL. WSDL uses XML to describe the service information. Client calls a service according to the methods and parameters etc. gained from the WSDL description file of the service. When the information of the service interface is changed, WSDL file of the service will be changed accordingly. By WSDL, we can realize the loose coupling between the client and server.

AXIS server can expand WSDL file automatically for web service, and provides a command-line tool Java2WSDL for generating WSDL description file. Compared with automatically expanding, Java2WSDL has more flexible functions.

### 3.3 Method for WMS registration

Once WMS is deployed on the application server, other applications and services may call the service through its address. In order to make more service requestor can get the basic information and address of WMS, we need register
WMS into the UDDI registration center. UDDI of web service provides several functions for web service, such as dynamically publishing, searching, binding etc. Using UDDI, Web service provider can register its service into service registration center. It can also visit the website of UDDI registration center to register its service. Service requestors, or other web services or applications search demanded services by UDDI, and then get its WSDL description file for calling the service.

Since AXIS server does not support UDDI nowadays, we can adopt UDDI4J, a java class library proposed by IBM. It provides the API for interacting with UDDI registry. Using UDDI4J, we can register WMS to existent registration center, such as IBM registration center or Microsoft registration center. We can also create our own registration center for service requestors.

3.4 Client instance of WMS

After the deployment of WMS, applications or web services on any platform can call it. WMS client can be coded in any language. We present an instance of WMS client in this section, which is coded in JSP. In this instance, by querying the UDDI registration center, client finds the WSDL address of WMS, invokes Getcapabilities and GetMap operations of the service, and then gets the image of the map (Fig. 5).

4 Conclusion

WMS is the most fundamental portrayal service in OGC Web Service Model. This paper presents the method and key techniques for the implementation of the OGC WMS by web service technology. We have primarily studied some issues of the web-based GIS interoperation. In WMS, Map data in same scale and projection from different map server can be directly overlapped, however, for those data in different scale and projection, map server need to perform some conversion before sending them back to the client.

REFERENCES

1 OGC (2001) OGC Web services initiative. http://opengis.org/ows/index.html
2 OGC (2002) OGC request for quotation and call for participation in the OGC Web Services initiative initial operating capability and demonstration Annex B. OWS1.2 architecture http://opengis.org/ows1.2/docs/020314_OWS1_2_Annex_B.doc
3 Huang X B (2002) Research on spatial information service and application integration based on GeoAgent, Ph. D dissertation. Beijing: Peking University. (in Chinese)
4 de La Beaujardière J (2002) Web map service implementation specification. http://www.opengis.org/techno/implementation.htm

Fig. 5 Returned map image