Implementation Method of OGC Web Map Service Based on Web Service

Anthony Wenjue Jia*,Yumin Chen *,Jianya Gong *
***Wuhan University**

ABSTRACT

The most important advantage of web service is the multi-platform, multi-architecture and multi-program-language interoperability on Web, which provide efficient solution for GIS interoperability. 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 article gives a method and key techniques to implement OGC Web Map Service based on Web Service technique.

KEY WORDS: Web Service, OGC OWS, Web Map Service, WMS

1. INTRODUCTION

With the rapid development of GIS, lots of independent GIS software and data have emerged, so that providing a way for the interoperability among various GIS becomes an urgent demand. Since the Internet has grown web into a new platform for distributed applications, this then provides a technological support to realize GIS interoperability. Traditional Distributed Object Technology, such as CORBA, DCOM and EJB etc., allows distributed objects to invoke with one another. They are relatively more mature technologies that provide some advanced services, like security, transaction management and so on. In a homogeneous LAN, They have advantages on effectiveness and efficiency. However, since Internet is a tremendous distributed heterogeneous network, it’s too difficult to deploy a CORBA- or DCOM- or EJB-based large-scale network application. It may be a nightmare for IT engineers to do so. Furthermore, almost all solutions based on above technologies depend on the implementation of some single provider. Nowadays, though there are some protocol-transformation bridges realizing the interoperability among kinds of distributed object technology, whether this kind of interoperability can be extended into those high-leveled services, like security, transaction etc. and all the special optimizations made by different provider may be lost. The birth of web service provides the ability of crossing platform, architecture and language, 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 path for realizing GIS interoperability.

Open GIS Consortium (OGC) is one of several standard organizations in the field of spatial Information. Over recent ten years, it has made great efforts in seeking a way to effectively integrate the GIS, 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 the Portrayal service of the framework of OGC Spatial Information Service. This paper introduced the detailed process and key issues when implementing OGC Map Services using Web Service technology.

2. OGC Web Service model and Web Map Service

2.1 OGC Web Service model

OGC Web Services Initiative (OGC,2001) 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 bases 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 (OGC, 2001).

The basic services in OWS include Application Clients, Registry Services and Encoding, Data Services, Portrayal Services and Processing Services (OGC,2002). Figure 1 illustrates the OWS service framework. In OWS, data service provides basic service on spatial data, and the access capability to data sets. The Web Map Service (WMS) is a kind of Portrayal Services(Huang xiaobin, 2002).

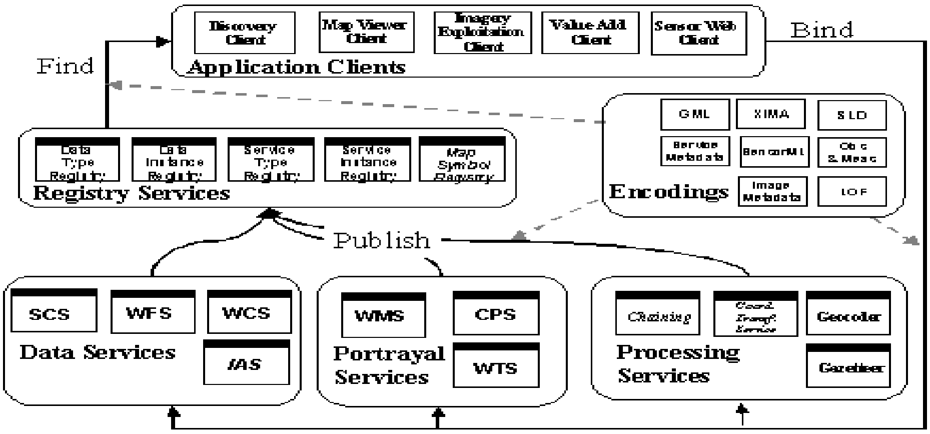

Figure 1. The OpenGIS® Service Framework

2.2 OGC WMS

OGC WMS is one of OGC Portrayal Services. It produces maps of georeferenced 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 spatial 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(Jeff de La Beaujardière, 2002).Figure 2 is the workflow of WMS:

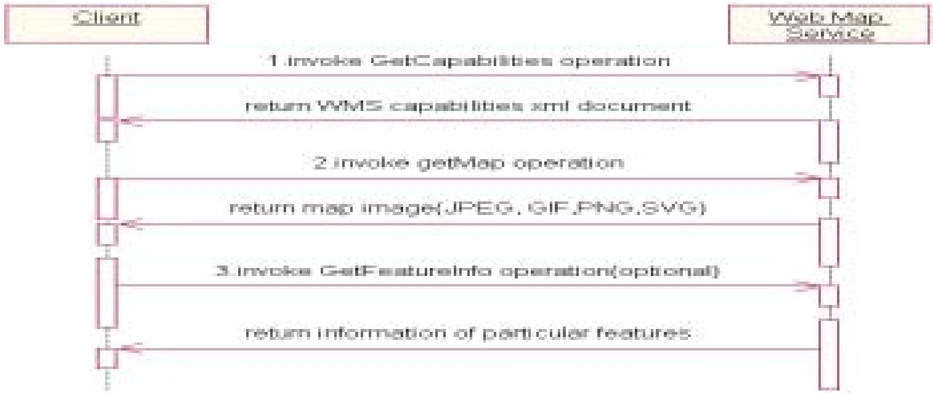
Client
Web Map Service
1.invoke GetCapabilities operation
return WMS capabilities xml document
2.invoke getMap operation
return map image(JPEG, GIF,PNG,SVG)
3.invoke GetFeatureInfo operation(optional)
return information of particular features

Fig.2 OGC Web Map Service work flow chart

3. The characteristics and architecture of Web Service Technology

3.1 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 defined how applications interoperate through the web. The best advantage of web service is that it allows applications, which run on different platforms, using different architecture and coded by different program languages, to invoke one another, but it does not need complicated configuration and deployment. 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 self-contained, self-described and modularized applications, and it can be discovered, located and invoked through the web.

3.2 The characteristics of web Service Technology

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

1) Sharing not only program, but also data;
2) Easy to be integrated with other program;
3) Having the capability of going through the fire wall;
4) Beneficial to software reusability;
5) Simplicity in configuration and deployment

3.3 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. After found the appropriate service, we can describe the file invocation service by the WSDL offered by the service registration center. Figure 3 presents the Web Service Architecture.

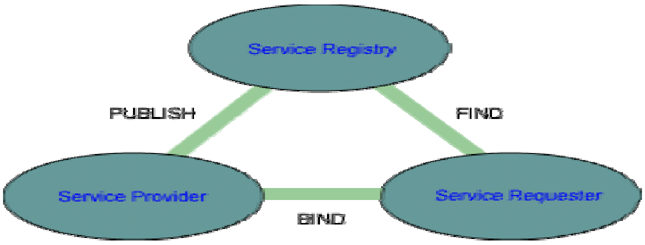

Fig.3 Web Service Architecture

4. Web-Service-Based Method for the Implementation of WMS

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

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

4.1 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 fore-mentioned interface.

From the view of technical implementation, we can look web service as an application, which exposes interfaces that can be invoked through web, and allow 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 the WMS specification. To create WMS application, the most important thing is to implement the operation interfaces and make it to satisfy the function requirements of the operation interfaces. Using GIS component 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 validated the cross-platform interoperation capability of WMS.

The architecture of WMS implementation is shown in figure 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.

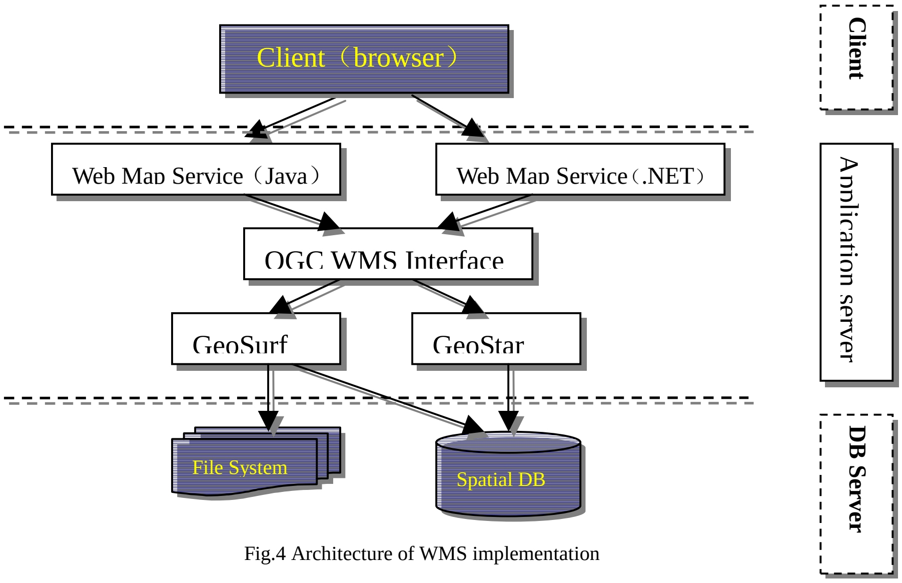

Fig.4 Architecture of WMS implementation

4.2 The approach for 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 implementing it. After finishing the development of WMS application, we need to deploy it on application servers, and make it to provide service for outside 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’s easy to deploy service over the .NET platform. Taking APARCHE 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 its 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 calsspath of Tomcat, then set the jar files of AXIS to the system classpath; Launch Tomcat after above configuration, and open the test pape 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 to establish a mapping between the name and the implementation of web service. The deployment file is as follows:

```
<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="wms.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. Comparing with automatically expanding, Java2WSDL has more flexible functions.

4.3 Method for WMS registration

Once WMS is deployed on the application server, other applications and services may call the service by 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 also can 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 doesn’t 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 also can create our own registration center for service requestors.

4.4 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, see figure 5.

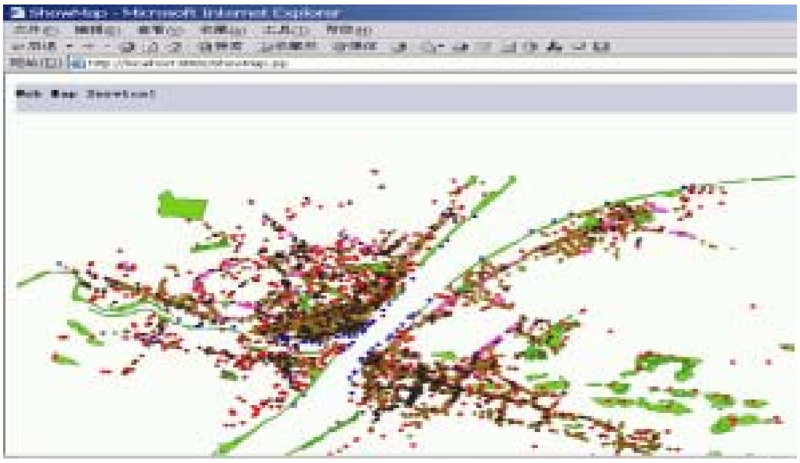

Fig5.Returned map image

5. Conclusion

WMS is the most fundamental protrayal service in OGC Web Service Model. This paper presented the method and key techniques for the implementation of the OGC WMS by web service technology. We 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

Jeff de La Beaujardière (2002). "Web Map Service Implementation Specification". http://www.opengis.org/techno/implementation.htm

Huang xiaobin(2002). "Research on Spatial Information Service and Application Integration Based on GeoAgent". Doctoral dissertation.

OGC (2001). "OGC Web Services Initiative", http://ip.opengis.org/ows/index.html.

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://ip.opengis.org/ows1.2/docs/020314_OWS1.2_Annex_B.doc.

OGC (2001), "A Request for Technology In Support of an OGC Web ServicesInitiative". http://ip.opengis.org/ows/rft/OGC_Web_Services_Initiative_RFT.doc, 2001