DEVELOPMENT OF THE INTEGRATING AND SHARING PLATFORM OF SPATIAL WEBSERVICES

Lan Xiaoji[1,2] Lu Guonian[1] Zhang Shuliang[1]

Shi Miaomiao[1] Yin Lili[1]

1. Jiangsu Provincial Key Lab of GIS Science, Nanjing Normal University,Nanjing,China
2. Environment and Architecture Department, Southern Institute of Metallurgy,Ganzhou,China

ABSTRACT

OGC Web Services specifications are the standard interfaces realized by more and more geo-processing software vendors. However, the different implementing technologies of software vendors bring inconveniences to web service users. This paper focuses on design and development of the integrating and sharing platform of Spatial Web Services (SWS). The paper first discusses why to establish the SWS integrating and sharing platform, then introduce basics of OGC Web Services and main technologies of implementing the SWS sharing platform. In section 4.0, design and development of SWS sharing platform are described in detail. Java technologies are widely used in web application development. In this paper are described approaches (such as servlets, JSP, JavaScript) to develop Java-based web applications of the SWS integrating and sharing platform.

KEY WORDS: spatial data sharing, spatial web services, WMS, spatial data interoperability, servlets, JSP, applet, JavaScript

1. INTRODUCTION

The Internet and especially the World Wide Web (WWW) provide a promising approach to make all kinds of data and information available publicly and privately (using an Intranet). There is a growing awareness that the WWW has changed forever the way data, information and knowledge is collected, analyzed and distributed. Using the WWW together with other elaborate distributed computing techniques, the general public will be

able to access, retrieve, merge, and analyze complex sets of geo-spatial information and data. As the trend of accessing GIS data via OpenGIS standards continues, the "spatial Web" will become as open as the Web itself [1]. Web users will easily find, view, overlay, and combine spatial information for a given region.

By far, more and more Commercial and non-commercial geo-processing software support OGC Web Services(OWS) specifications for interoperability and information exchange. Each OGC web service interface consists of a series of required and optional parameters used to communicate with the spatial web server. For different vendors, their web interfaces of OWS are realized as different objects, for instances some as Java servlets(e.g. ArcIMS, deegree), and others as CGIs, etc. ESRI's ArcIMS implements the interface of OGC's WMS as a Java Servlet: com.esri.ogc.wms.WMSServlet in WMS connector (com.esri.esrimap.Esrimap in servlet connector) [2].

If you want to get capabilities of one WMS of ArcIMS through WMS connector, you should fill out the address column of IE browser with some URL like the following:

http://<hostmachine:port>/ ogcwms/servlet/

com.esri.ogc.wms.WMSServlet?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities

If you want to get map from one WMS of ArcIMS through WMS connector, you should fill out the address column of IE browser with some URL like the following:

http://<hostmachine:port>/ogcwms/servlet/com.esri.ogc.wms.WMSServlet?Version=1.1.0&REQUEST=GetMap&Layers=cntry%2Crivers%2Ccities&SLD=&SRS=EPSG%3A4326&BBOX=-180%2C-90%2C180%2C90&WIDTH=600&HEIGHT=400&FORMAT=JPEG&TRANSPARENT=false&BGCOLOR=0xFFFFFF&EXCEPTIONS=application%2Fvnd.ogc.se_xml&URL=&SERVICENAME=wms_world&SERVICE=WMS&WMTVER=1.1.1

If you also install degree web map server and want to get map from this server, you should fill out the address column of IE browser with some URL like the following:

http://<hostmachine:port>/deegreewms/wms?service=WMS&VERSION=1.1.1&REQUEST=GetMap&LAYERS=europe:country&STYLES=default&SRS=EPSG:4326&BBOX=-10,40,12,60&WIDTH=640&HEIGHT=410&FORMAT=image/png&BGCOLOR=0xfff8ff&TRANSPARENT=true&EXCEPTIONS=application/vnd.ogc.se_inimage

From above statements, we can see: in order to share the spatial resources on the internet or intranets, users must acquaint themselves with the spatial web server software and OGC web services specifications. The objective of the research presented here is to establish one Visual integrating and sharing platform of Spatial Web Services (SWS) for web users to access spatial web services from multiple GIS vendors and to integrate multiple spatial web services in their spatial web applications.

The remainder of this paper is structured as follows. In the next section, we introduce basic knowledge of OGC Web Services. Main technologies of implementing the SWS sharing

platform is described in section 3.0. Design and development of the SWS integrating and sharing platform are described in section 4.0. Finally, conclusions and future work are presented in section 5.0.

2. BASICS OF OGC WEB SERVICES

The Open GIS Consortium, Inc. (OGC) is a not for-profit international industry consortium of more than 200 companies, government agencies and universities participating in a consensus process to develop publicly available geo-processing specifications. These specifications provide a vendor-neutral interoperability framework for web-based discovery, access, integration, analysis, exploitation and visualization of geo-data sources, sensor-derived information, location information, and geo-processing capabilities.

2.1. WMS

A Web Map Service (WMS) produces maps of geo-referenced data. A "map" is a visual representation of geo-data rather than the actual data itself [4]. WMS specification specifies interfaces and schemas that enable developers to enable users to dynamically integrate geographic information from several different WMS services, no matter what underlying technology is used.

WMS implementation specification defines three operations [4]: **GetCapabilities** returns service-level metadata, which is a description of the service's information content and acceptable request parameters; **GetMap** returns a map image whose geo-spatial and dimensional parameters are well defined; **GetFeatureInfo** (optional) returns information about particular features shown on a map.

2.2. WFS

The OGC WMS allows a client to overlay map images for display served from multiple Web Map Services on the Internet. In a similar fashion, the OGC Web Feature Service (WFS) allows a client to retrieve geo-spatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services.

WFS implementation specification defines the following WFS operations [5]:

GetCapabilities

A web feature service must be able to describe its capabilities. Specifically, it must indicate which feature types it can service and what operations are supported on each feature type.

DescribeFeatureType

A web feature service must be able, upon request, to describe the structure of any feature type it can service.

GetFeature

A web feature service must be able to serve a request to retrieve feature instances. In addition, the client should be able to specify which feature properties to fetch and should be able to constrain the query spatially and non-spatially.

Transaction

A web feature service may be able to service transaction requests. A transaction request is composed of operations that modify features; that is create, update, and delete operations on geographic features.

LockFeature

A web feature service may be able to process a lock request on one or more instances of a feature type for the duration of a transaction. This ensures that serializable transactions are supported.

The OGC divides WFS implementations into two categories:

- Basic WFS ----implements the required operations (a read-only implementation), and;
- Transaction WFS ----adds the Transaction capability, and possibly the LockFeature capability, to the Basic implementation.

2.3. WCS

The Web Coverage Service supports the networked interchange of geo-spatial data as "coverages" containing values or properties of geographic locations. Unlike WMS, which filters and portrays spatial data to return static maps (server-rendered as pictures), the Web Coverage Service provides access to intact (unrendered) geo-spatial information, as needed for client-side rendering, multi-valued coverages, and input into scientific models and other clients beyond simple viewers.

The Web Coverage Service consists of three operations [6]: GetCapabilities, GetCoverage, and DescribeCoverageType. The GetCapabilities operation returns an XML document describing the service and the data collections from which clients may request coverages. Clients would generally run the GetCapabilities operation and cache its result for use throughout a session, or reuse it for multiple sessions.

The GetCoverage operation of a Web Coverage Service is normally run after GetCapabilities has determined what queries are allowed and what data are available. The GetCoverage operation returns values or properties of geographic locations, bundled in a well-known coverage format. Its syntax and semantics are similar to the WMS GetMap request, but several extensions support the retrieval of coverages rather than static maps.

3. MAIN TECHNOLOGIES OF IMPLEMENTING THE

INTEGRATING AND SHARING PLATFORM

In developing Web applications, in order to build customized content based on the user's input and to enhance the simplicity of using HTTP to request information, there are many technologies that can be used. These are described below.

3.1. Extensible Markup Language (XML)

The OGC web services specifications are based on W3C's XML specification. Understanding it requires understanding the basics of XML. Overall, you should familiarize yourself with the following [7]:

- Creating an XML document instance
- Rules for XML well-formalness
- Distinction between well-formed and valid documents
- Internal and external subsets
- Elements and attributes
- Entities (parameter, general, internal and external, parsed and unparsed)
- Notations
- Parsing/Validating a DTD and XML document

XML is best thought of as a language or encoding for data description. More correctly, XML is a language for expressing data description languages. XML however, is not a programming language. There are no mechanisms in XML to express behavior or to perform computations. That is left for other languages such as Java and C++. XML provides a means of describing (marking up) data using user defined tags. Each segment of an XML document is bounded by starting-tags and end-tags.

Almost every data model that can be expressed in an XML document can be represented as a tree, where the root is the top element and the branches are the other elements nested in it. Elements have relationships with each other that will be familiar to anyone who has worked with object-oriented programming. In terms of the tree, as the three branches out, the leaves and twigs become *children* of the branches. Elements that are on the same level in the documents are called *siblings*.

Each element has an element type name and zero or more attributes, and each attribute consists of a name and a value. Each element consists of two tags: a start tag and an end tag. An element can have attributes, which are written inside the start tag. Each attribute has a name and a value.

There can be multiple attributes of one element, and the content of the element and the values of the attributes are to a very large degree interchangeable, as follows:

```
<ARCXML version="1.0">
 <REQUEST>
  <GET_IMAGE>
   <PROPERTIES>
     <ENVELOPE minx="-122.5" miny="37.8" maxx="-122.4" maxy="37.9"/>
   </PROPERTIES>
  <GET_IMAGE>
 <REQUEST>
</ARCXML>
```

The following technologies and tools that are parts of the XML family that are relevant to spatial web service:

DTD & XML Schema

XSL (XSLT, XPath, etc)

XPointer, Llink

XMLNS (Namespaces)

DOM and SAX

3.2. ArcXML

ArcXML is the protocol for communicating with the ArcIMS Spatial Server [8]. An ArcIMS Spatial Server is the backbone of ArcIMS and provides the functional capabilities for accessing and bundling maps and data into the appropriate format before sending the data back to a client. In order to understand ArcXML, it is first necessary to understand how configuration files, ArcIMS services, requests, and responses relate to each other and how they interact with the ArcIMS Spatial Server.

ArcXML, a derivative of XML, a relative of HTML, and a subset of SGML, is a meta-markup language that describes structured data content rather than display information. ArcXML files are text files with an .axl extension that can be edited. ArcXML defines content for MapServices as well as for requests from clients, responses by ArcIMS, and communications between the business logic tier and servers. By manually inserting ArcXML elements and appending element attributes in the map configuration file, additional rendering and labeling options, data subsets, tabular joins, and other capabilities, not available through the ArcIMS Author interface, can be accessed.

3.3. Servlets [9][10]

Servlets are generic extensions to Java-enabled servers. Their most common use is to extend Web servers, providing a very secure, portable, and easy-to-use replacement for

CGI. A servlet is a dynamically loaded module that services requests from a Web server. It runs entirely inside the Java Virtual Machine (JVM). Because the servlet is running on the server side, it does not depend on Web browser compatibility.

3.4. JSP and JSTL [9][10][11]

Two technologies or approaches to developing Java-based Web applications that utilize Web services are Java Server Pages (JSP) and JSP with the use of tags from the JSP Standard Tag Library (JSTL).

JSP is based on Java servlet technology and allows the insertion of Java code directly into static HTML. A JSP page is basically an HTML page with inline Java code that manipulates dynamic content and specific tags in addition to the regular HTML tags. The embedded Java code is expressed using script elements. A JSP page is processed by a JSP container. When the Web browser makes a request for a JSP page, the JSP container in the Web server first compiles the JSP page into a servlet, which is a Java program. Then the JSP container compiles the servlet with more business logic Java code (i.e. JavaBeans), and finally the JSP container executes the compiled servlet class to produce a generated Web page to the Web browser. Therefore, a JSP page is actually another way to write a servlet without having to be a Java programming expert.

JSTL is based on JSP, but uses standard functional HTML-like tags instead of Java code to manipulate dynamic content in a JSP page. JSTL includes tags for many common tasks such as looping over data, performing conditional operations, importing and processing data from other Web pages, simple XML manipulating, database accessing, and text formatting. JSTL also supports an Expression Language (EL) that is inspired by both ECMAScript (JavaScript) and the XPath expression languages. This makes JSP pages have easier access to the data required.

3.5. Applets

Applets are Java-based GUI components that typically execute in a Web browser. Applets can provide a powerful user interface for Web-based distributed applications. Applets have access to all the features and advantages of the Java platform technology.

In a heterogeneous Web environment, it is especially important that client-side components be portable. For the protection of the client machine, it is important to be able to place security restrictions on these components and detect security violations. Java applets serve both these needs.

3.6. JavaScript

A JavaScript is lines of executable computer code, which are interpreted by the browser at runtime and scripts are usually embedded directly in HTML pages. All major browsers,

like Netscape and Internet Explorer, support JavaScript.

When a client request an HTML page with a script, the browser reads it from the top and executing Javascripts instruction as they occur and instantly displaying the result. These statements can react to some user driven events, such as mouse clicking, page navigation, inserting text into forms etc.

Core JavaScript contains a core set of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects.

4. DESIGN AND DEVELOPMENT OF THE SWS INTEGRATING AND SHARING PLATFORM

4.1. Design of the SWS Integrating and Sharing Platform

The intention of developing the SWS sharing platform is to enable web users who have little knowledge about OGC web services' specifications and spatial web server software to easily integrate geographic information from several different web services of different vendors, no matter what underlying technology is used. With the visual interface of the SWS sharing platform, users need not to know the details of OGC web services specifications, and can share spatial resources from different vendors. Figure 1, below, shows the architecture of application of the SWS sharing platform.

In this Architecture, when a client request is made, it is first handled by the SWS sharing platform to decide to which spatial server the requested service belongs, and then forwarded to the spatial server to get the map layers needed. The hierarchy model of the SWS sharing platform is portrayed in Figure 2 below.

The SWS sharing platform wrapped the components that fulfill the following functions:

- CatalogServices: manage the all kinds of services from all spatial web servers;
- GetServices: get all services from multiple spatial web servers;
- GetLayers: get all layers from a specified service;
- OverlayMaps: overlay all maps from different Services or Servers.

The components composing the SWS sharing platform is illustrated in Figure 3.

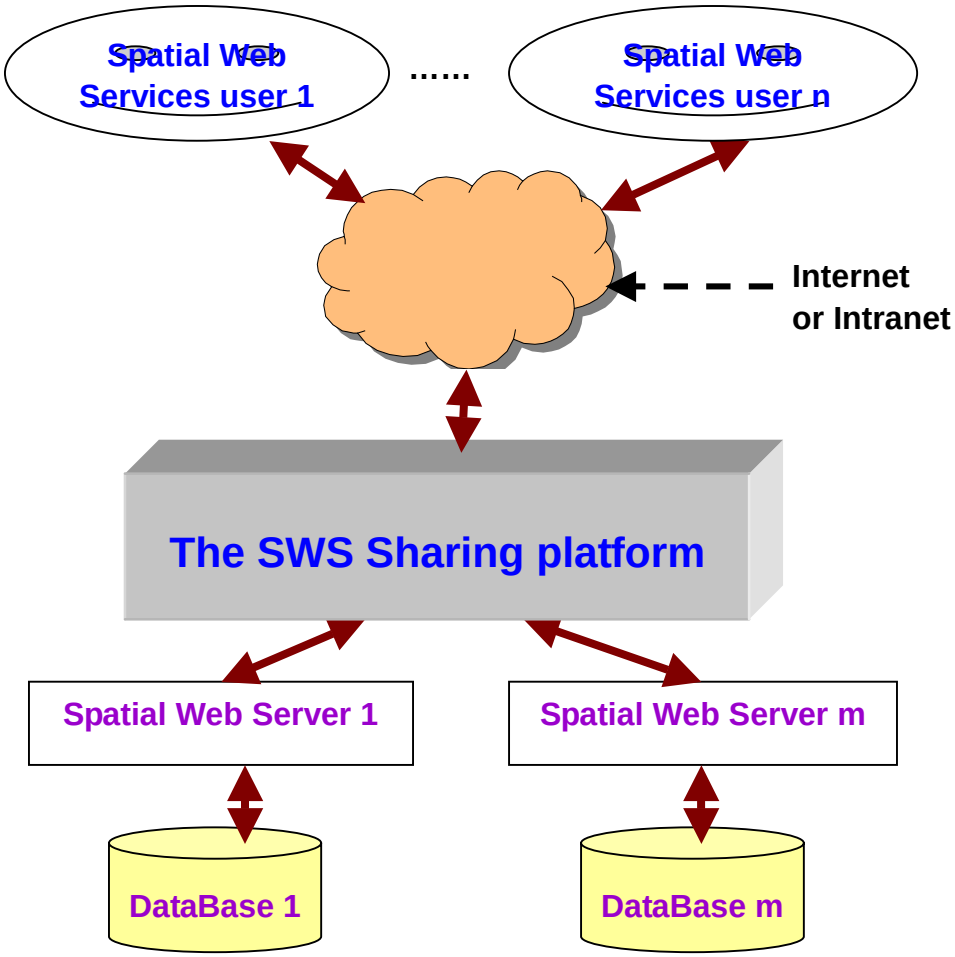

Fig.1 Architecture of application of the SWS sharing platform

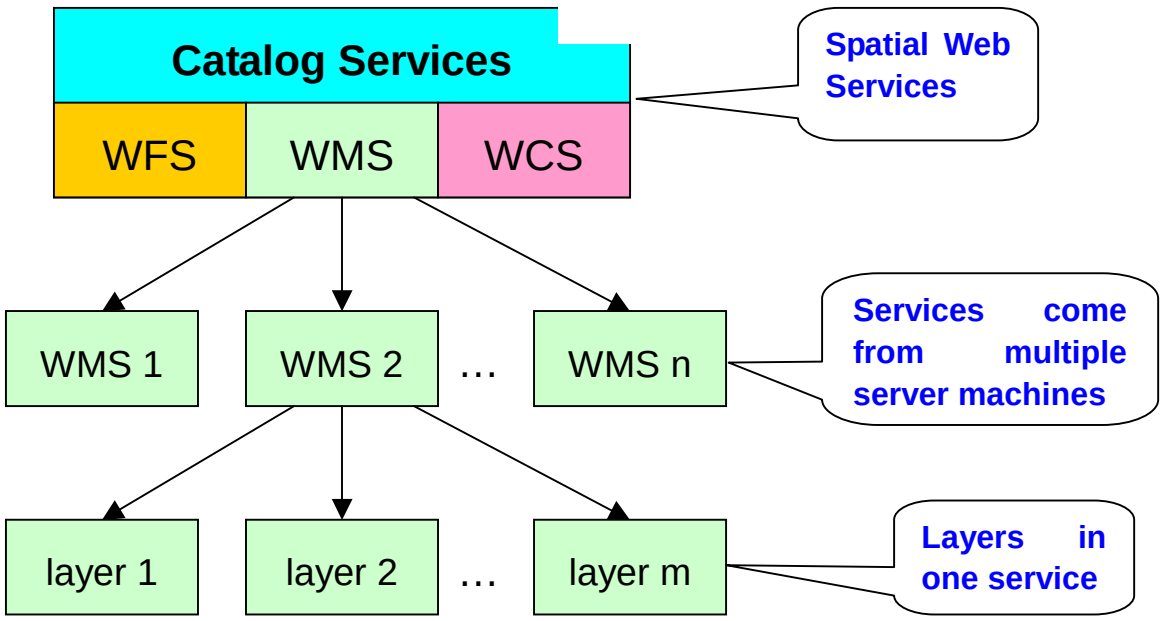

Fig.2 The hierarchy model of the SWS sharing platform

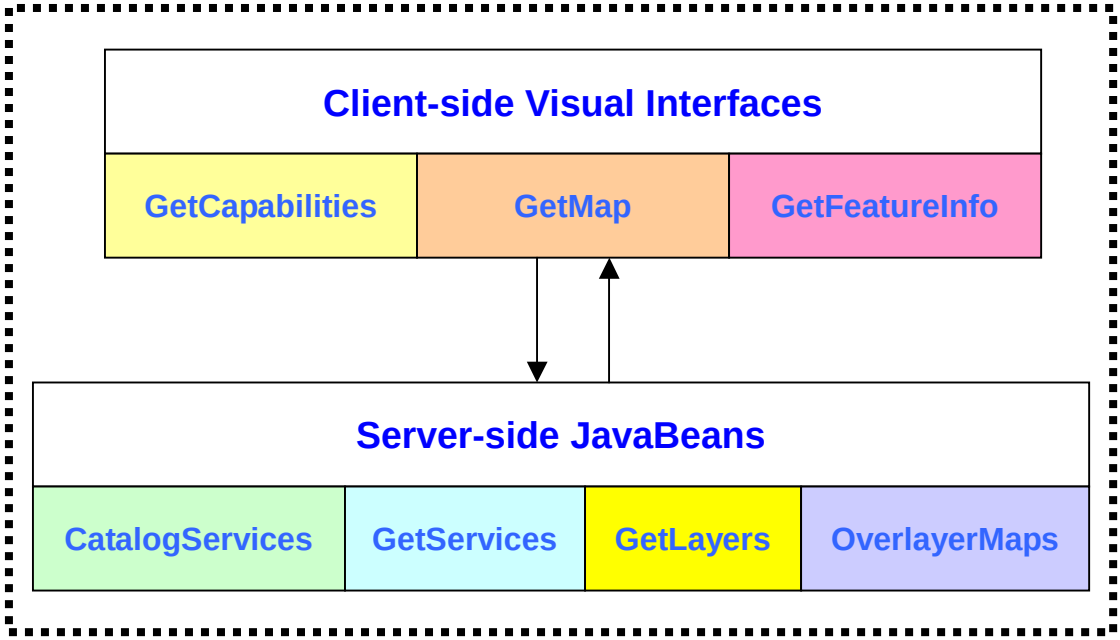

Fig.3 The components composing the SWS sharing platform

4.2. Development of the SWS Integrating and Sharing platform

4.2.1. Developing Environment

In the research experiment, we select the following developing environment:

- Operating System: Windows 2000 server/professional
- Web Server: IIS 5.0
- Servlet Engine: ServletExec 4.11
- Spatial Web Server: ArcIMS 4.01 & degree 1.1
- Developing Language: Java & JavaScript
- Developing IDE: JBuilder

In our experiment, the spatial web services come from ArcIMS 4.01 and degree 1.1 Spatial Web Server.

4.2.2. Implementation

According to the design described above, the development of the SWS sharing platform covers the server-side JavaBean components and the client-side visual interfaces.

The server-side components are realized as JavaBeans[11][12][13]: ServicesCatalog.class, GetServices.class, GetLayers.class, OverlayMaps.class.

The client-side visual interfaces are actually a series of HTML pages running inside a Web browser that can interact directly with a spatial web server via the HTTP profile of the OGC Web service Interfaces. The visual interfaces of WMS GetCapabilities, GetMap, GetFeatureInfo are showed in Figures 4,5,6 respectively.

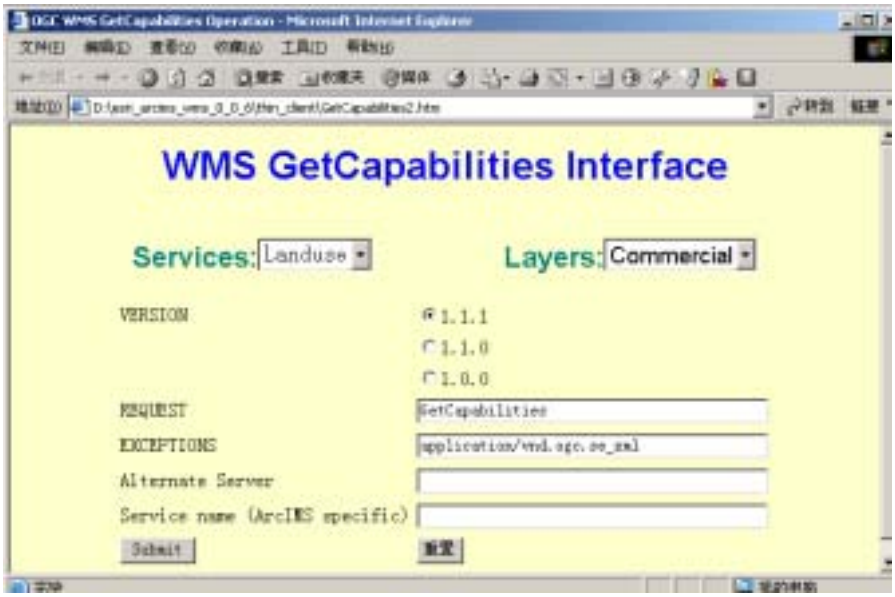

Fig. 4 The visual interface of WMS GetCapabilities

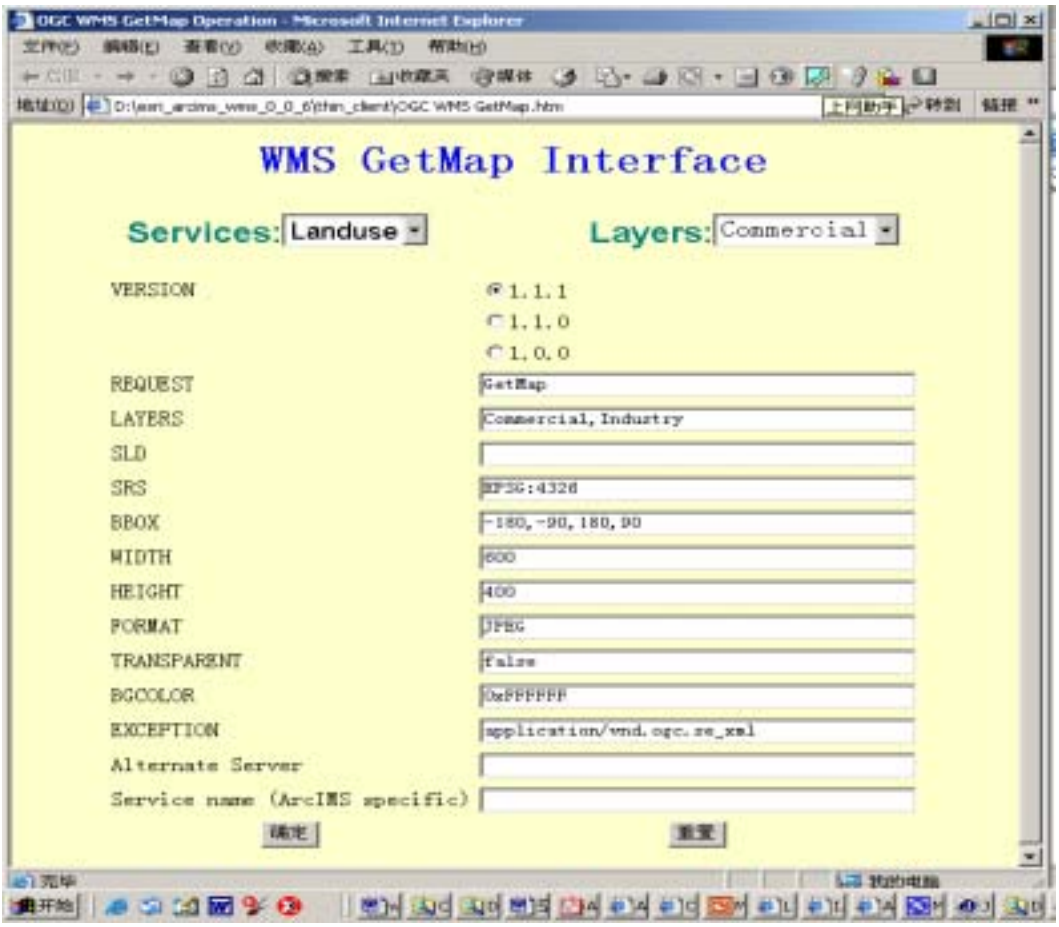

Fig.5 The visual interface of WMS GetMap

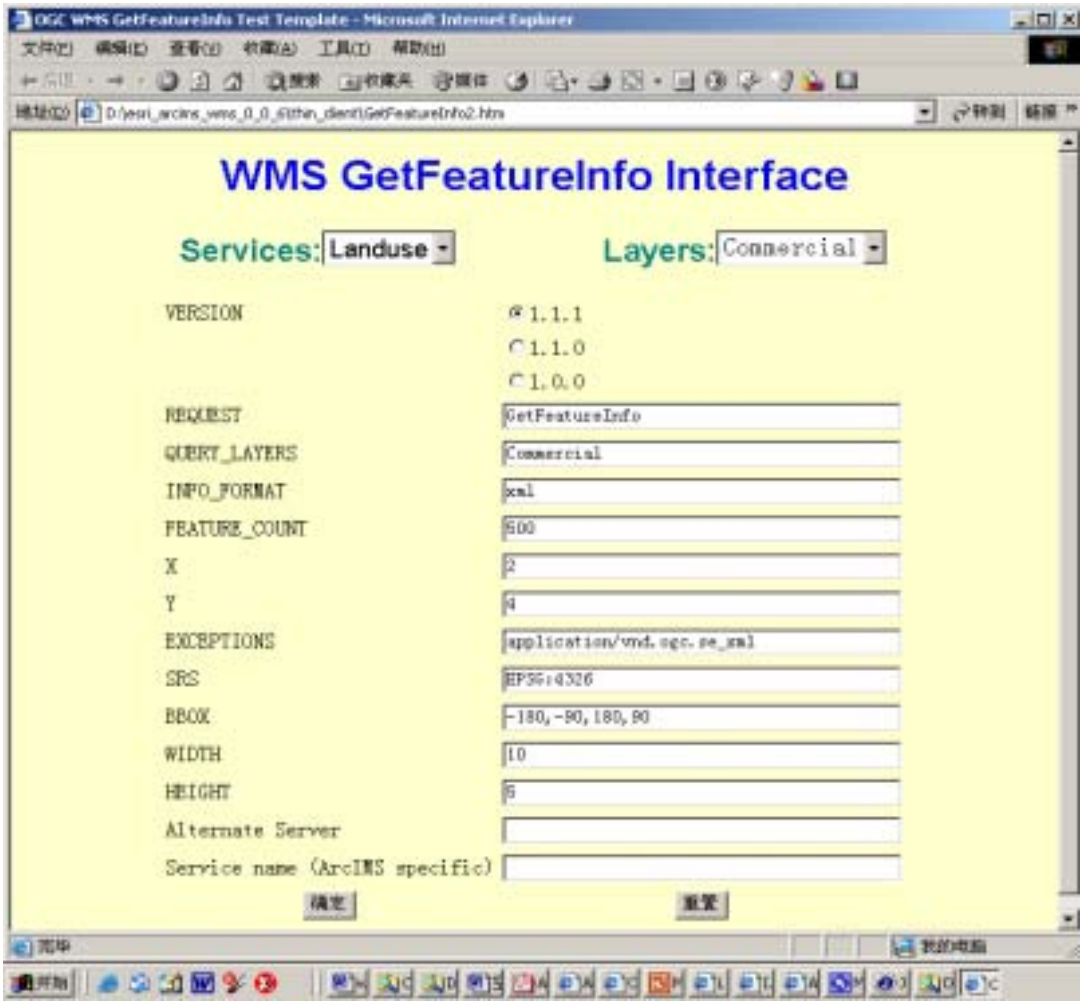

Fig.6 The visual interface of WMS GetFeatureInfo

5. CONCLUSIONS AND FUTURE WORK

One of the ultimate goals of the GIS industry is to have full *interoperability* between web-based geographic datasets enabling information stored at different locations on the web to be viewed together in single applications. In our practice, Java Servlets technique was chosen to implement the server-side component, JSP and JavaScript were used to implement the client-side visual interface. The SWS integrating and sharing platform developed in this research can easily integrate geographic information from several different web services of different vendors. Up to now, we only realized WMS sharing platform on ArcIMS and degree. The current system is still under development. A list of improvements will be made in the next stage of the project, including WFS, WCS function, and integrating more spatial web servers, such as MapXtreme, MapGuide, GeoSurf, etc.

6. REFERENCES

[1].OpenGIS Consortium (2003), OpenGIS Web Map Server Cookbook

[2]. An ESRI White Paper (2003), ArcIMS 4 Architecture and Functionality

[3]. http://www.opengis.org

[4]. OpenGIS Consortium (2002), Web Map Service Implementation Specification 1.1.1

[5]. OpenGIS Consortium (2002) , Web Feature Service Implementation Specification 1.0.0

[6]. OpenGIS Consortium (2002) , OpenGIS Web Coverage Service Implementation Specification 0.9

[7]. W3C (2000), Extensible Markup Language (XML) 1.0 (Second Edition)

[8]. ESRI document (2002), ArcXML Programmer's Reference Guide

[9]. http://java.sun.com

[10]. Sun Microsystems, Core Servlets and JavaServer Pages .

[11]. Hans Bergsten (2002), JavaServer Pages™(2nd Edition)

[12]. Sun Microsystems (2003), The J2EE™ 1.4 Tutorial

[13]. Sun Microsystems (2003), The Java™ Web Services Tutorial