

A WFS BASED GEOGRAPHICAL FEATURES SERVICE APPROACH FOR INTERNET GIS

Penglin Zhang^{a,*}, Feng Zhang^b, Bolin Ruan^b and Zhiyong Lv^a

^aSchool of Remote Sensing and Information Engineering, 430079 Wuhan, China - zplcy@263.net

^bTianjin Land & Resources & Housing Management Research Centre

WG II/6

KEY WORDS: Web GIS, Online Map, Web Service, Web Feature Service, GML

ABSTRACT:

While Web Service based Web GIS architecture has undergone the significant development in recently decade years, most of them, for instance, Google Maps, Yahoo Maps and other online maps or web-based GIS are based on the Web Map Service (WMS) standard interface and return only raster data which can not be edited or spatially analyzed. Therefore, the OGC membership defined and maintains the Web Feature Service (WFS) specification, the OpenGIS WFS interface allows requests for geographical features across the web using platform-independent calls. In WFS specification, XML-based geographical markup language (GML) is the default payload encoding for geographical features transportation. However, although the OGC claim that non-GML format, such as shapefiles can also be used for transportation, an un-ignorable fact is that the data organized in non-GML format must be pre-reorganized in XML Schema before using it. This must increase the difficult for non-GML format data distribution and share via WFS-based system. In this paper, we propose a WFS-based approach that not only the GML encoding can be transported, but other formats such as shapefiles also can be transported online without any pre-encoding. In the proposed approach, we propose to use an online component which set on the server side to parse the non-GML geographical features and transform it into GML schema and provide standard interface based on the WFS specifications. To test the validity of the proposed approach, we developed simple web-based online map, and applied it to the actually shapefiles. The experiment results show the proposed approach is valid.

1. INTRODUCTION

It has been predicted that the web and distributed mobile and wireless networks may eventually become the dominant form for accessing GIS (i.e. GIS will be freed from the desktop). For instance, web GIS currently has the higher potential user base and the lowest cost per user [1]. Especially, with the incredibly rapid proliferation of web applications, Internet commerce, wireless communication, and small electronic devices, it will be important to investigate how to further improve the integration of geospatial analytical functionality with these technologies [2].

It is obvious that Web GIS and online maps are important research branches in GIS field. Just as introduced in literature [3], through the desktop PC, Client/Server and the most popular Browser/Server computation pattern, it is lately developing into the service-oriented computation pattern (also called GIS Web Services), which is base on the Web Services technology. GIS Web Services is a branch of Web Service. It integrated GIS, XML, Web Service and so on. Because the advantages of data sharing, function and services integration, GIS web service becomes an important kind of distributed and interoperable GIS, and is supposed to promote the development of Web GIS in the following years.

Presently, the international organizations, such as ISO and OGC, already have the corresponding abstract standards about how to provide Web GIS services, and the OGC even has set down a series of implementation specifications, which include three basic geographic information services: WMS (Web Map

Service), WFS (Web Feature Service) and WCS (Web Coverage Service), besides some other services like geographic coding and encoding services which can return spatial coordinates references. Moreover, the WFSs are able to provide all kinds of spatial vector data services via internet. The OGC WFS implementation specifications [4] support add, delete, edit and such operations on the spatial vector features, and also the locking ability to avoid operating conflicts, so they can solve problems related spatial vector data sharing and interoperating commendably. Even though, some shortages appear in WFS implementation specifications: First, the "standard" data (GML data) which correspond to OGC specifications are needed for the WFSs, however, lots of "non-standard" data (non-GML data) exist; Second, there isn't a interface with spatial coordinates extent as parameters to return spatial vector data to the requestors, while such a interface is needed practically.

To overcome the limitations mentioned above, in this paper, we propose an approach based on online component which run on the server side to transform the non-GML spatial data to the GML spatial data and provide the standard SOAP/HTTP access interface which satisfy to the WFS specification. Concisely, in order to support the non-GML spatial data in the system based on WFS, we propose to design an online component run on server-side. The server-side component server as a real time interpreter between non-GML and GML spatial data, and output the standard WFS interface after transforming non-GML to GML data. Namely, the designed online server-side components act as not only an interpreter of non-GML spatial

* Penglin Zhang. Ph.D, Associate professor of Wuhan University, my interest research areas include network GIS and computer vision. Email: zplcy@263.net

data, but a service provider. In Section 2, we describe the overview of the proposed solution; Section 3 gives the implementation of the solution and gives the experiments and preliminary results in section 4. In the final section, we present a discussion and conclusions. Finally, in section 5, the conclusions and further works are given.

2. OVERVIEW OF THE APPROACH

Similar with the basic architecture of web service architecture [5], the architecture of proposed solution is an extension triangle, shown as Fig.1

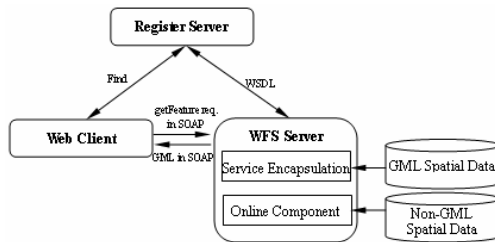


Figure 1: Architecture of the proposed method

In Fig.1, different from the standard architecture of OGC WFS, we set a component (also called Service Component) on the server side to implement the parse non-GML spatial data, transform it to the GML schema and encapsulate it as a service with standard SOAP/HTTP interface. WFS feature server is an OGC compatible Web Service based Web Feature Server. Here, it is used to provide web feature service satisfy to the OGC WFS specification. And its service expressed using Web Service Define Language (WSDL)-Web service Interface Description Language encoded in XML. Where, the service component acts as the critical role in transformation between non-GML descriptive spatial data to GML descriptive spatial data. Requests and returned GML data from WFS are put into SOAP envelope. SOAP is an XML based message exchange protocol used in Web Services interactions. Register server used to distribute services defined in WSDL. Web clients should prepare client stubs to be able to invoke the Web Services defined in WSDL.

3. IMPLEMENTATION OF SERVICE COMPONENT

The WFS uses GML [7] to encode the Simple Feature abstraction [6] and the OGC Filter Encoding Specification [8] to allow the formation of complex queries. Therefore, for those features organized using traditional formats, such as .shp file, must be pre-organized according to GML specifications before sharing in Web Service based architecture. This must bring much difficult for sharing and distributing of existed non-GML spatial data. The purpose of proposed components is to encode non-GML data to GML data online and provide standard SOAP/HTTP interface for web users.

3.1 What Components?

To implement shapefile to GML online encoding on server side, in this paper, we define three server side components: App_Code, shp2gml and gml2svg. Fig.2 shows the relationship between three components.

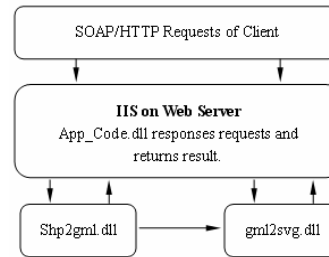


Figure 2 relationships between proposed service components

App_Code

App_Code is a component set on IIS server and used to deal with requests from client. It is the interface between client and the other components shp2gml and gml2svg. In implementation, to implement the defined functions, App_Code provides 17 interfaces, shown as following list:

- ◆ ExtractAllGMLDataURL(), Get URL of GML documents of all overlapped layers.
- ◆ ExtractAllSVGDataURL(), Get URL of SVG documents of all overlapped layers.
- ◆ ExtractGMLData(), Get GML document of appointed range of current layer and return XmlDocument.
- ◆ ExtractGMLDataURL(), Get URL of the appointed range of current layer.
- ◆ ExtractSVGData(), get SVG document of appointed range of current layer and return XmlDocument.
- ◆ ExtractSVGDataURL(), get URL of SVG document of current layer.
- ◆ GetAllEnvelope(), get the maximum coordinate range of all layers.
- ◆ GetAllGMLData(), get GML document of all overlapped layers and return XmlDocument.
- ◆ GetAllGMLDataURL(), get URL of GML document of all overlapped layers.
- ◆ GetAllSVGData(), get SVG document of all overlapped document and return XmlDocument.
- ◆ GetAllSVGDataURL(), get URL of SVG document of all overlapped layers.
- ◆ GetEnvelope(), get the coordinate range of current layer.
- ◆ GetFileNames(), get data information can be provided in server side.
- ◆ GetGMLData(), get GML document of current layer and return XmlDocument.
- ◆ GetGMLDataURL(), get URL of GML document of current layer.
- ◆ GetSVGData(), get SVG document of current layer and return XmlDocument.
- ◆ GetSVGDataURL(), get URL of SVG document of current layer.

From above interface list, we note that the App_Code can return not only GML related information, but also SVG related information. Those interfaces related to SVG can further transform GML data to SVG in server side. The SVG data will be transported to the client side for display.

shp2gml

Functions of component shp2gml include:

- (1) Read shapefile and encode it to GML document according to the defined GML3.x-based SHP2GML.xsd.

- (2) Transform GML format to SVG format via calling component gml2svg.dll.
- (3) Crop map according to the requested range.

In shp2gml component, we defined an interface named ImportSHP, this will be called to read shapefile when component App_Code initialization. Moreover, other interfaces, for instance, ExportGML, GetGMLData, ExtractGML and so on are defined to deal with different requests come from client.

gml2svg

Compare with the shp2gml, the functions of gml2svg is relative simply. The main function of gml2svg is parse GML document and create SVG file. Fig.3 shows the main functions of component gml2svg.

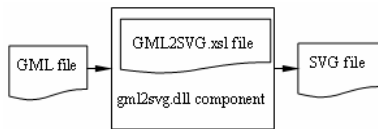


Figure 3 the function of component gml2svg.dll

3.2 GML Schema of This Paper

In order to define the structure of GML documents, a schema definition file (.xsd file) needs to be defined. In this paper, to simply problems, we only explain the process from ESRI shapefile to GML document. To implement the GML encoding of point, polyline and polygon features in shapefile, we define a schema definition file named SHP2GML.xsd, which based on GML3.x (GML3.0 and GML3.1.1). The main contents of SHP2GML.xsd shown as:

```
<? xml version="1.0" ? >
<? xs:schema
xmlns="http://www.whu.edu.cn/berlinr"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.whu.edu.cn/berlinr"
attributeFormDefault="unqualified"
elementFormDefault="qualified" version="1.0">
<xs:import schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"
namespace="http://www.opengis.net/gml" />
<xs:complexType name="shp2gml Type">
<xs:complexContent mixed="false">
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1"
name="Feature_ID"
nillable="true">
<xs:simpleType>
<xs:restriction base="xs:double">
<xs:totalDigits value="18" />
<xs:fractionDigits value="5" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="1"
name="Feature_NAME"
nillable="true">
<xs:simpleType>
<xs:restriction base="xs:string">
```

```
<xs:maxLength value="254" /> </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="1"
name="TheGeometry"
nillable="true" type="gml:GeometryPropertyType" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="shp2gml"
substitutionGroup="gml:Feature"
type="shp2gml_Type" />
</xs:schema>
```

where notation "Feature_ID" represents the identification of the object, "Feature_NAME" represents the name of the object, and "TheGeometry" represents the geometry information of the object.

3.3 Non-GML Features to GML Features

After obtaining the schema definition file, the next work is to transform non-GML file to GML-based file according to the structure defined in the schema definition file. In order to provide the standard web feature service, the proposed component also define standard web feature service interface using WSDL. In this paper, we only introduce how to transform ESRI shape file data to GML document. In general, this process includes two steps. First, parse shape file and obtain the real spatial data. Then, create construct GML document according to the GML specifications and GML schema definition file SHP2GML.xsd defined in section 4.3.

In implementation, according to a pre-defined rule, the content of shapefile is re-encoded and result the GML document which satisfies to the GML 3.x specifications. Concisely, the corresponding rule may be described as: A shapefile will correspond to a layer (shapefile layer) in GML document. One layer contains one "gml:boundedBy" (boundary of the layer) and multi- "gml:featureMember" (objects in the layer). Each geometry objects (no matter point, polyline or polygon) in shapefile needs to be encoded to a object with shp2gml type (gml:featureMember) in GML document. And that a shp2gml object contains "Feature_ID", "Feature_NAME" and "TheGeometry". There into, the relationship between the type of geometry object in shapefile and the geometry type xsi:type of TheGeometry in GML document is shown in table 1.

Objects in shapefile	Objects in GML file
Point, Multi-Point	gml:Point
Polyline	gml:LineString
Polygon	Gml:Polygon, gml:MultiPolygon

Table 1: corresponding between geometry object in shape and GML file.

Then, according to the GML schema definition file SHP2GML.xsd defined in section 4.3 and above corresponding rule, we can generate standard GML document which satisfies to the GML 3.x specifications. The following sample is part GML segment of Layer Lake.

```
...
<ShapefileLayer LayerName="lake.shp" LayerID="1">
```

```

<gml:boundedBy>
<gml:Envelope>
<gml:lowerCorner>
<gml:X>85008.4453787739</gml:X>
<gml:Y>12591.9101117145</gml:Y>
</gml:lowerCorner>
<gml:upperCorner>
<gml:X>168713.63585673</gml:X>
<gml:Y>54493.8281027387</gml:Y>
</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<gml:featureMember>
<shp2gml>
<Feature;ID>2</Feature;ID>
<Feature;NAME>Polygon 2</Feature;NAME>
<TheGeometry xsi:type="gml:Polygon">
<gml:exterior>
<gml:coordinates>98032.8516629039,54154.1993024224
98000.750030874, 54158.7499584266...
</gml:coordinates>
</gml:exterior>
</TheGeometry> </shp2gml>
</gml:featureMember>
...

```

3.4 Webservice-based Interface of Components

From section 3.1, we note that the proposed components provide seventeen interfaces in all. There into, GetFileNames(), GetEnvelope(), GetGMLData(), GetGMLDataURL(), ExtractGMLData(), and ExtractGMLDataURL() are six basic standard interfaces based on OGC WFS specifications. Clients can request data service from server side via standard SOAP/HTTP request using these basic interfaces. Moreover, these basic OGC specifications based interfaces can act as the supplement of OGC WFS specifications in Geographical feature service. And in practice, it can server as assistant of WFS.

4. EXPERIMENTS

To test the validity of the proposed solution, we applied it on the real data of Shenzhen city. In experiment, we designed and developed a simple WFS-based online geography system. Here, the proposed components are set on the server side and used to parse the requested shapefile and re-encoded it to GML document, further to transform to SVG format for display in client side.

4.1 Configuration of server

In experiments, we choose visual studio .NET as developing environment, C# as the developing language. Moreover, we choose IIS as web server. Fig. 4 shows the configuration of servers. Virtual directory named "wfsAssistant" in the IIS server used to set the developed online components.

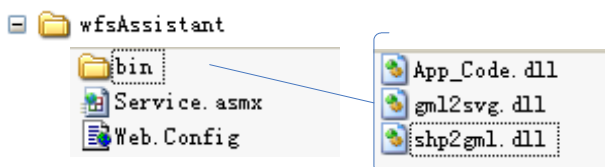


Figure 4 the configuration of server

Web.config represents the configuration file of server. It used to set source of shapefile, application schema of components run on server side, path of style file and virtual path of returned vector data. This file needs to be revised when server configuration.

4.2 Data Service Initialization

Before sending to the client, the vector data service needs to be initialized in server side (see Fig.5). In initialization, the requested data will be pre-read and placed in rand memory. The purpose is to avoid the server to read data frequently.

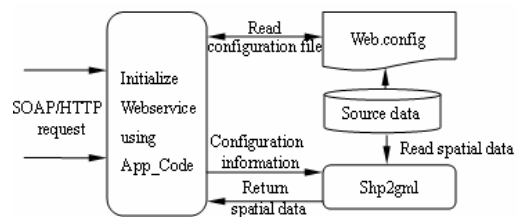


Figure 5 the flowchart for data service initialization

After finishing the service initialization, the interfaces of vector data service in server side then can be called.

4.3 Experiment results

In experiments, we apply developed components on actual data of lakes of Shenzheng data. All the original data are stored in shape file. Fig. 6(a)~(c) show the returned GML documents and vector data of developed components according to the requested range of client-side. There into, Fig.6 (a) shows the returned GML documents of Shenzheng Lake Layer. Fig.6 (b) shows the returned SVG data correspond to the GML document shown in Fig.6(a). Fig. (c) shows the SVG result of multi-layer overlapped each other of requested range.

```

- <ShapefileLayer LayerName="深圳市湖库.shp" LayerID="1">
- <gml:boundedBy>
- <gml:Envelope>
- <gml:lowerCorner>
  <gml:X>85008.4453787739</gml:X>
  <gml:Y>12591.9101117145</gml:Y>
</gml:lowerCorner>
- <gml:upperCorner>
  <gml:X>168713.63585673</gml:X>
  <gml:Y>54493.8281027387</gml:Y>
</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
- <gml:featureMember>
- <shp2gml>

```

(a) GML document segment of Shenzheng Lake Layer



(b) SVG data of Shenzheng Lake Layer



(c) SVG data of overlapped multi-layers of requested range
Figure 6 experiment results

Figure 6 the returned GML documents and vector data of developed components

5. CONCLUSIONS AND FURTHER WORKS

This paper proposes a solution for the web feature service of traditional non-GML data. The preliminary experiment results reveal that the proposed solution is feasible. Comparison with the existed WMS-based Web GIS or online mapping, advantages of the proposed solution include:

- (1) The proposed solution may return the geographical features service which used to edit or perform spatially analysis.
- (2) The server side component is based on the WFS specification, thus, it has the strong interoperability and reusability.
- (3) It is different the traditional middle-ware, the proposed component return WFS based standard interfaces, not the data itself.
- (4) Systems constructed using the proposed approach will full take advantage of the existed resources like the traditional non GML spatial database without any WFS-based pre-organization.
- (5) The proposed approach provides a solution for addressing non-GML features share and interoperation based on WFS framework.
- (6) The proposed solution will be a good supplement to OGC WFS on dealing with non-GML-data.

Further improvements of the proposed solution include:

- (1) Increasing the compatibility for non GML profile described spatial data formats.
- (2) Improving the parsing and transport speed of spatial data by using the compress technique for geographical features and GML document.

ACKNOWLEDGEMENT

This research is partially supported by Natural Science Foundation Committee (NSFC, Code: 2005ABA031) of Hubei province, China.

REFERENCES

- [1] Longley, P.A., Goodchild, M.F., Maguire, D.J., and Rhind, D.W. (2001) Geographic Information Systems and Science. Chichester, John Wiley and Sons
- [2] Wright, D., 2002. Research Priorities of the University Consortium for Geographic Information Science. [http://www.ucgis.org/priorities/research/2002researchPDF/short term/q_pervasive_computing.pdf](http://www.ucgis.org/priorities/research/2002researchPDF/short%20term/q_pervasive_computing.pdf) (accessed 23 Apr. 2008)
- [3] Ruan, B.L., 2007. A way of spatial vector data service based on WFS, Master thesis, Wuhan, China.
- [4] OpenGIS Web Feature Service (WFS) Implementation Specification. <http://www.opengeospatial.org/standards/wfs> (accessed 10 Apr. 2008)
- [5] Web Services architecture overview. <http://www.ibm.com/developerworks/webservices/library/w-ovr/> (accessed 28 Apr. 2008)
- [6] Open GIS Abstract Feature Specification. <http://www.opengeospatial.org/standards/as> (accessed 16 Apr. 2008)
- [7] OpenGIS Geography Markup Language (GML) Encoding Standard. <http://www.opengis.org/techno/implementation.htm> (accessed 16 Apr. 2008)
- [8] Filter Encoding Implementation Specification <http://www.opengeospatial.org/standards/filter> (accessed 28 Apr. 2008)

