Design of Service-Oriented Architecture for Spatial Data Integration and Its Application in Building Web-based GIS Systems

SHA Zongyao, XIE Yichun
International School of Software, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

© Wuhan University and Springer-Verlag Berlin Heidelberg 2010

Abstract In this paper we propose a service-oriented architecture for spatial data integration (SOA-SDI) in the context of a large number of available spatial data sources that are physically sitting at different places, and develop web-based GIS systems based on SOA-SDI, allowing client applications to pull in, analyze and present spatial data from those available spatial data sources. The proposed architecture logically includes 4 layers or components; they are layer of multiple data provider services, layer of data integration, layer of backend services, and front-end graphical user interface (GUI) for spatial data presentation. On the basis of the 4-layered SOA-SDI framework, WebGIS applications can be quickly deployed, which proves that SOA-SDI has the potential to reduce the input of software development and shorten the development period.

Keywords spatial data integration; web-based GIS; service-oriented architecture; software

CLC number P208

Introduction

A large volume of data has been accumulated along with the advances made in information technology. The spatial data, as an important part of available data resource, has played critical roles in practical problem-solving and decision-making applications in various fields. With spatial data physically distributed at different locations, a critical challenge arises from the need of integrating the available data sources, analyzing the data, and representing the results. At the same time, the rapid development of computer hardware and software has stimulated an explosion of mapping applications on the World Wide Web (WWW or the web). Many GIS applications, once focusing on data and tools implemented with client-server (C/S) architecture or “single machine” model, are now migrating to a web-based model or WebGIS[1]. WebGIS systems become popular because they offer more potential benefits of flexibility, ubiquity, and reduced costs and risks of obsolescence and isolation, making remote spatial data easily accessible by an exponentially growing, diverse user community over computer networks[2].

While a wide range of approaches have been proposed to build web-based GIS programs, we focus on building web-based GIS systems by integrating different spatial data sources with a service-oriented architecture. To achieve this objective, we try to address the following two issues: (1) design of ser-
vice-oriented architecture for spatial data integration (SOA-SDI); (2) development of web-based GIS systems based on SOA-SDI. As examples, we developed two WebGIS applications based on our proposed architecture, proving the flexibility and value of the proposed architecture.

1 Design of SOA-SDI

The architecture of SOA-SDI is logically composed of four hierarchical layers (components): front-end interface, layer of a set of backend services, layer of data integration, and layer of data provider services (see Fig.1). The bottom layer, i.e., the layer of data provider services, is a set of remote data provider services for data sharing and integration. Each data provider service offers a set of interfaces through which client applications can pull remote data in and manipulate the data over the network. The data integration layer logically combines all the available data provider services, passes data request from client applications to data provider services, and conversely send back data response from data provider services to client applications. The third layer is a set of backend services that communicate with multiple data sources via data integration layer, and interact with end users to analyze and manipulate data coming from data provider services. As an example, we mainly focus on a set of map sketching tools which is located within this layer. Lastly, the most upper layer is the front-end interface layer which serves as a graphic user interface (GUI) to present the result of spatial data, allowing the end users to interact with the backend services. Front-end interface is application-specific in terms of the layout of the interface elements. In summary, the architecture connects multiple spatial data sources, coordinates various spatial data sources, integrates backend services or tools for data processing and analyzing, and presents the data in desired formats through Web-based GIS systems over the Internet, Intranet or Wide Area Network (WAN).

1.1 Multiple data provider services

The proposed architecture fully takes the layer of data provider services as a key component because, (1) a large amount of spatial data, sitting on numerous servers residing in different organizations or locations, has been accumulated, (2) data acquisition is
one of the most time-consuming and expensive input in building GIS applications. Therefore, making use of available spatial data for GIS applications is very helpful and cost effective. However, due to technical limitations such as the heterogeneous data model adopted by different data sources, as well as social reasons (e.g., data security consideration), direct data access to different data sources proves to be difficult. Therefore, SOA-SDI becomes an alternative for sharing spatial data among organizations and applications. In SOA-SDI, client applications access remote data sources through a set of interfaces (services) provided by the data provider. OGC (Open Spatial Consortium) OpenGIS®, for example, has proposed various specifications for spatial data sharing and interoperability based on service-oriented strategies (http://www.opengeospatial.org/). Among those, WMS (or Web Map Service) and WFS (or Web Feature Service) have been widely adopted by web-based GIS applications to access spatial data in different data sources over the Internet. OGC OpenGIS also specifies WFS-T (WFS-Transactional) that supports transactional operations for the target spatial data, e.g., adding, deleting, or updating spatial features. Any software platforms are regarded as compliant with OGC OpenGIS specification if it implements the required operations of that specification. WMS, for instance, must have GetCapabilities and GetMap interfaces implemented according to the OGC OpenGIS WMS specification (http://www.opengeospatial.org/standards/wms). Based on such software, spatial data can be published as spatial data provider services and consumed by other client applications over the Internet. Generally, the consumption process involves two steps, request and response. The first step is to make a request from the consumer applications to the data provider service in the form of XML request format. The response, also in XML format, returned from the data provider service will be passed to the consumer applications. Other data provider services, though not fully compliant with OGC OpenGIS specifications, are capable of providing similar interfaces as those provided by OGC OpenGIS compliant ones in terms of spatial data request and response mechanism[4]. In SOA-SDI-based applications, all available spatial data are accessed through data provider services, and the multiple data provider services can be setup through which remote data are requested and returned.

1.2 Data integration

Data integration is located on the upper layer of the data provider services that coordinates multiple data provider services for different applications. The main role of this component is to build up data request in a proper format which will be sent through Internet/Intranet/WAN to specific data provider services, and in return, parse the responses returned from those services. Accordingly, the layer of data integration provides a bidirectional gate for accessing and integrating data from a variety of provider services for applications and parsing the response. For any request from client applications, the data integration component is responsible for sending a well-formatted request (in the format of XML) that the destination remote data provider services understand. For each response, the data integration is responsible to extract interested data for its upper layer, backend services. Some important parameters, such as map coordinate transformations and reprojections if required, are also included and formatted by the data integration component. With the layer of data integration, different applications do not need to specify which server the interested data is on but leave this decision by such a layer. In this sense, the layer of data integration is also a common gate for all client applications.

1.3 Backend services

The layer of backend services is composed of a series of services or components on which WebGIS applications are built. Note that the components can be dynamically manipulated (updated, removed, or added). As an example, we present here a set of map sketching tools that can sketch different spatial objects on a digital map. Other backend services, such as spatial data editing, map rendering and web page pushing services, are also included in this layer, which will be reported separately.

The traditional spatial data models employing either vector model or raster model have limitations. For example, in vector data model, the real world is digitalized as point, line and polygon features. Every feature is logically organized within a particular fea-
ture layer for storage and presentation. In such a data model, a layer can only represent one theme, e.g., river layer, land parcel layer, etc. The limitation of this model is that it cannot represent multiple-type features in one theme layer. When considering a theme layer with multiple-type features and map annotations, map sketching tools are preferred to meet the requirement of combining different feature types on one layer over the Internet. Map sketching tools work on particular spatial data provider services that differ from others in that the target data layer of data provider services supports the multiple-type features. Freehand drawing tools consisting of different sketching elements (symbols), along with designed icons and pictures, are provided as map sketching tools to work on digital maps to represent different types of features.

As shown in Fig.2, the map sketching tools have such elements as adding text (map labeling) feature, point feature, line feature and polygon feature. In addition, icon and picture features are also supported by map sketching tools designed to combine pictures or icons on a map. The icon feature and picture feature have similar properties except that an icon feature usually refers to an image with smaller size while a picture feature is much larger. All the sketching elements can be “glued” together in a thematic layer which, unlike general GIS layers that are organized by unique feature type (polygon, line or point), proves easy to control (e.g., turning layer on or off). All of the layers with various sketching elements are serialized (saved) in a sketched feature pool for storage and retrieval purpose.

The sketching elements exist in two forms, screen form and storage form. The former is for visualization purpose which is shown on GUI for representation while the latter records the physical coordinates of the elements in the sketched feature pool (a place for storing the elements). The map sketching tools gather sketched elements in screen form and convert them into storage form before sending them to the feature pool via the data provider service. On the other hand, any returned sketched elements from the feature pool, once a request has been processed by the data provider service, is again parsed by the map sketching tools before it is shown on data viewer (e.g., web browsers) in screen form. This two-way conversion ensures that, the sketched elements, along with all the other integrated data, are under the same spatial reference system. To display the sketched features on GUI from the feature pool, a series of parameters are predefined, out of which spatial references (coordinate and projection systems) are required. For example, if the data provider service of map sketching receives a data request with spatial reference specified, the returned features will be processed by map sketching tools according to the spatial reference information and thus the features can be reprojected in the right way to align (overlay) well with spatial features pulled from other data sources.

![Functionalities and general workflow of map sketching tools](attachment:image.png)
The digital map can be sketched or annotated by the sketching tools (text tool, point tool, line tool, etc.). For example, a text (label) can be created (through “add new” operation) and saved to the sketched feature pool (SFP). All the elements in the SFP can be displayed properly with specified displaying parameters. Fig. 2 shows that the map sketch tools support operations like “add new”, “delete”, “update” and “style setting”. For each operation, the feature type can be text, point, line, and etc. All features stored in the sketched feature pool can be displayed under limited coordinate systems and the preferred one must be specified when making request from client applications.

The different formats can be employed to represent and manipulate the returned sketched elements depending on the types of data viewer (web browser)[5]. The popular web friendly and text-based markup languages, SVG (Scalable Vector Graphics) and VML (Vector Markup Language), are selected as candidates to represent spatial features due to their advantages in spatial data representation[6].

1.4 Front-end interface

The front-end interface provides an interactive communication between the Web-based GIS systems and end users. It is composed of GUI elements (e.g., buttons, tabular forms, and map images), which provide the gateway for the end users to interact with applications. For WebGIS applications, a map-centered frame is usually preferable due to its easy-to-understand property for showing spatial data.

2 Implementation

The proposed architecture provides a key software infrastructure on which different applications can be quickly built. As examples, two applications, Safe Route-to-School (SR2S) and Public Facility Management (PFM), are deployed to demonstrate the potential value of the infrastructure. Both applications share the available data provider services although different applications have variations in terms of the data use. The multiple data provider services include WMS services, Google map, ArcIMS (See http://www.esri.com) services, and Pictometry (See http://www.pictometry.com) image service, etc., which are setup within a local network or available on remote servers over the Internet. For the layer of backend services, the map sketching tools, map rendering service, map editing tools, etc., are integrated in the proposed SOA-SDI.

2.1 Data provider services

Four categories of data provider services are taken to visualize spatial data for the above mentioned applications: Google map, WMS services, ArcIMS services and Pictometry image service. Google map has an attractive map with plenty of information that allows public users to take the advantage of it in their routine works, e.g., finding a location and making queries on route information. More importantly, Google map has a set of APIs that allow it to be integrated in other web applications. The most prominent functionalities of Google map include, if not all, routing, geocoding and high resolution of aerial photos for urban areas, and attractive visualization of maps. By applying Google map as a data provider service, the features that Google map possess can be taken over to other applications. The second data provider service is WMS services, which support interfaces such as “GetMap” and “GetFeatureInfo” to retrieve data based on the OGC specification. “GetMap” allows the web applications to specify distinct layers, the spatial reference system, the geographic area and other parameters regulating the returned map format. “GetFeatureInfo” provides feature information by identifying a point on a map based on its pixel location, with required parameters for each request. WMS services can be called over the Internet no matter where the data request comes from. Once a WMS service is setup, it can serve various applications. However, one limitation of WMS services is that the map style is predefined, meaning that it does not support customized map rendering. Correspondingly, the third kind of services, ArcIMS image services, is created to allow end users to customize the map style according to individual application requirement. Compared to other data provider services, ArcIMS image services are more open in terms of customization capability. Lastly, a Pictometry image provider service is developed and setup to provide the
applications with a detailed 3-D data view for interested regions where high spatial resolution images are required. Pictometry image service, as a data provider service, is a web service that accepts requests from remote applications and return 3-D images from a Pictometry image warehouse. The integration of 3-D images and other data from various data provider services offers a wonderful data displaying for decision making. The returned images can be viewed from different angles as specified in the request (see Fig.3).

![Fig.3 Example of the high resolution and 3-dimention Pictometry image powered by Pictometry image data provider service](image)

Functional sections in the user interface include: 1-setting image scale; 2-moving or center image; 3-setting image type (either setting viewing angle for oblique image by clicking N-North, E-East, W-West or S-South, or ortho image by clicking cross icon in the center of the circle); 4-the returned image center location; 5-setting image size (three levels, big, standard and small, are defined).

Data provider services can be categorized into different data groups based on the roles that the services play. Flexible access to data services by different applications is possible because data provider services can be easily adjusted (added or removed) for different applications. For example, the Google map data provider service can be used to locate an address, to find a route or to show the surrounding environment for a building. Some WMS services can be integrated to map the static spatial features, which mainly serve as reference data. ArcIMS services are more suitable for representing thematic layers when an interactive rendering is preferred to emphasize particular features, or when spatial query and the spatial analysis are required. Meanwhile, the Pictometry image service focuses on representing locations where detailed information about the local environment are desired. The component of data integration enables all of the pulled-in data from the data provider services georeferenced and overlaid, so that all the data can be overlaid, visualized and compared within the web-based data presentation interface.

### 2.2 Backend services

We present here the map sketching tools mentioned previously. In the following applications, the map sketching tools use VML for displaying spatial features, and save the elements to a data storage system with the current map coordinate and projection system as the spatial reference. Conversely, all sketched elements can be retrieved from the data storage system and shown on GUI with the same spatial reference system as they are created. The integration of sketching map within traditional GIS data in web-based applications proves to be an ideal way to realize data sharing and integration, since different users can update the sketched pool, and the updated data can be reflected simultaneously by other users in a multiple user application.

### 2.3 Applications

The two WebGIS applications, Safe Route-to-School (SR2S) and Public Facility Management (PFM), are built based on the infrastructure discussed. They are introduced and accessible on the Web at http://geodata.acad.emich.edu/mayordashboard and http://geodata.acad.emich.edu/sroute for demonstration purposes. In SR2S, end users can create a safe route to school for students. The Google map image combined with Pictometry high resolution aerial photo image (from Pictometry image service) is integrated in the application and used as the reference image (Fig.4). In PFM, information of available public facilities pulled in from WMS service is visualized on the GUI, so that users can better understand the status of buildings and public facilities (Fig.5). The map sketching tools are used to sketch an easy-to-understand map. The overlay of the Google map along with Pictometry’s high resolution aerial photo images also provides a vivid visualization of data. It needs to be pointed out that the two applications can share data provider services, data integration component and backend services, though the front-end interface is
application-specific. Therefore, the main input for building WebGIS applications is the front-end interface design while all the other functions rely on the available service-oriented architecture. This greatly decreases the input of building new WebGIS applications. Therefore, the proposed architecture has great potential in creating complicated software.

The sketched elements, which are included in one spatial layer, clearly indicate the location and area of the dangerous building labeled as “immediate repairing required”. In addition, two blocked traffic shown in red circles and five enterable traffic shown by black arrows are also sketched on the map. The map with the sketched elements is assumed to be a more easy-to-understand map compared to the one without the elements.

Fig. 4 Main interface of SR2S

The application uses Google map as the base map which is overlaid by sidewalk segments (highlighted in yellow) collected by end users. Data sketching tools are integrated in the project for collecting sidewalks (line features).

3 Conclusion

Service-oriented architecture (SOA) for building software is not new itself and has been proved to be an effective structure in implementing software applications [7]. In this paper, we introduce the design a 4-layered SOA-SDI which can be adopted to build various WebGIS applications. Compared with traditional SOA, this 4-layered SOA-SDI shows more flexible and powerful when developing new GIS applications. For example, the data integration layer even encapsulates remote data provider services through which client applications do not need to specify which data provider services they are requesting. Once SOA-SDI constructed, developing new WebGIS applications will be a work of putting different services together and of designing front-end interface.
In SOA-SDI, we particularly focus on multiple spatial data integration. Multiple data source integration can provide more context information for WebGIS users than a single data source to make well-informed decisions or problem-solving. As the volume of spatial data available continues to grow, the need for extracting, analyzing and displaying the data is increasing. Since many open data sources have been available on the web, integrating application-related data by pulling data in and combining them on a same data viewer involve much less input than duplicating the data production. Thanks to the available technologies for spatial data publishing and sharing such as Open-GIS WFS, WMS, ArcIMS image services, and Google map, these technologies are becoming more popular now in developing spatial information systems.

Note that the backend services included in SOA-SDI can be flexibly combined. Depending on different applications, one or more services can be enabled without relying on the other services. As the services developed are web-based and can be called over a network (the Internet, Intranet, or WAN), this structure proves to be a solid basis for quickly building new WebGIS applications.

With the SOA-SDI proposed, we have implemented it in WebGIS applications. Our studies have shown that the different applications (e.g., SR2S and PFM) can be easily deployed while certain attention is needed only for the design of different front-end interfaces.

Acknowledgement

This work was undertaken at IGRE (the institute of geographical research and education), Eastern Michigan University. The authors appreciate the data preparation and the involvement of software development by graduate student, Guan Wang.

Reference

[1] Hecht L (2002) Web services are the future of geoprocessing [J]. GeoWorld, 15(6): 23-27
[2] Geoffrey A, Rafael M (2003) Building web-based spatial information solutions around open specifications and open source software[J]. Transactions in GIS, 7(4): 447-466
[3] Sips M, Schneidewind J, Keim D A (2007) Highlighting space-time patterns: Effective visual encodings for interactive decision-making[J]. International Journal of Geographical Information Science, 21(8): 879-893
[4] Sha Zongyao, Xie Yichun (2009) Building spatial information systems based on geospatial web services [C]. Proceedings of the 2009 World Congress on Software Engineering, Hangzhou
[5] Xi Yantao, Wu Jiangguo (2008) Application of GML and SVG in the development of WebGIS[J]. Journal of China University of Mining and Technology, 18(1): 140-143
[6] Dunfey R I, Gittings B M, Batcheller J K (2006) Towards an open architecture for vector GIS[J] Computers & Geosciences, 32: 1720-1732
[7] Kuk S H, Kim H S, Lee J, et al. (2008) An e-engineering framework based on service-oriented architecture and agent technologies[J]Computers in Industry, 59(9): 923-935