A multi-tenant WebGIS system

Honglei He

School of Information Engineering, Lianyungang Technical College, Lianyungang 222006, China;

He_hong_lei@163.com

Abstract

WebGIS has a wide range of application requirements, but development and application require high labor and resource costs. In order to solve this contradiction, a multi-tenant WebGIS system is proposed. Through the statistical summary of various application scenarios, the system provides a variety of geographic information atomic services to choose from. Tenants can freely combine these services to implement a WebGIS system that meets their needs. The user interface is implemented by an online HTML editing system, and the data structure is self-made through a metadata service, and the business logic is self-made through service assembly and business rules. Business rules are ultimately done through event processing by client-side JavaScript. Through the establishment of an instance, the feasibility of multi-tenant WebGIS system is verified, and users can build WebGIS system according to their own needs.

Keywords

SAAS; WebGIS; service composition; online editing; JavaScrip.

1. Introduction

With the development of the Internet, the application of many information systems has also developed rapidly. WebGIS is one of them. Users can use geographic information services through browsers, such as browsing, querying, analyzing and modifying spatial data. WebGIS has greatly improved the sharing scope and operational efficiency of spatial information, so this application has been rapidly popularized and rapidly developed in various industries, including transportation, weather, logistics, electricity, agriculture, tourism, and community. Security, campus management, etc.

At present, many enterprises and institutions hope to have their own WebGIS system, which requires the development of a WebGIS system. At present, the software commonly used in development systems: ARCGIS, mapinfo, etc. in foreign countries, Topmap, supermap, etc. in China. These development software are expensive and require a professional developer. After the development is completed, it is also necessary to set up a special operating environment for the system. For most organizations, it is a big burden.

In order to solve the contradiction between this broad demand and high cost, this paper proposes a WebGIS system based on SAAS mode. The system adopts B/S mode, the development platform is on the server side, the data is stored on the server, and the running environment is on the server. Users do not need to install software, just use the browser to complete the customization and use of the WebGIS system. The huge cost of purchasing professional software is eliminated. In addition, the article proposes a WebGIS template development technology suitable for various industry applications, and gives the creation and use methods, making WebGIS application development simple and fast, reducing the threshold of use.

2. WebGIS system structure

WebGIS refers to a geographic information system based on WWW (World Wide Web) technology running on the Internet. Through the World Wide Web technology, the use of WebGIS is very convenient and popular. Any user on the Internet can use geographic information services through browsers, such as browsing, querying, analyzing, and modifying spatial data. Compared with

traditional GIS, WebGIS greatly improves the sharing range and operational efficiency of spatial information.

WebGIS uses World Wide Web technology to transfer information, so it has some features of the World Wide Web. For example: with platform independence, users can use the browser to access various types of GIS services on various server sides; with distributed source data, WebGIS data can be distributed on multiple servers, and WebGIS data can be compared with other Webs. Information integration; B/S mode, WebGIS decomposes the geographic information function on the server side and the client side, and can distribute the load reasonably.

WebGIS can be divided into the following architectures according to the distribution of work tasks on the server side and the browser side [1]:

(1). Centralized, all data processing is placed on the server side, the browser only displays the result, the result is that the server load is too heavy, this way can not adapt to the current multi-user multi-tasking.

(2). two-tier, data processing distribution is done collaboratively at both the server and the browser. The server side is mainly data logic, the browser is mainly representation logic, and the business logic is completed by the server side and the browser side.

(3). three-tier, on the basis of the two-tiered add a middle layer responsible for business logic, making the server and browser side transparent and functional.

The most mainstream way is the layered tile map represented by google and Baidu. The client uses the scripting language Javascript supported by the browser by default, without the need to install any plug-ins, and is universal and easy to use. The server provides a Web Map Service (WMS) that returns the corresponding map tiles based on the user's request.

Therefore, the SAAS mode WebGIS system adopts the system structure shown in Figure 1. The client runs a JavaScript-based program to request data from the server, then builds a map and supports various map operations: zooming, dragging, marking, etc.; the server executes the client's request and sends the result back to the client over the network, including the map tile. And spatial data. The browser side obtains the data of the background server through the AJAX engine and loads it, and then realizes mutual calling of the webpage element and the map object through the DOM document object model.

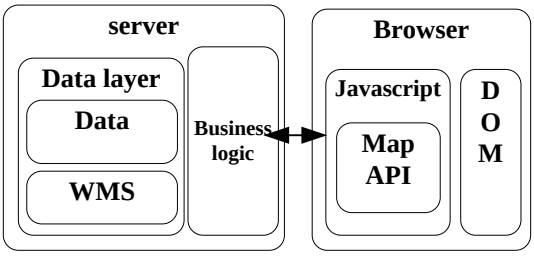

Fig. 1 Architecture

3. system design

3.1 SAAS mode design

SAAS mode WebGIS systems need to be customizable. Because the system has to support a large number of users at the same time, and the needs of a large number of users will not be exactly the same, the system must provide customizable features. Tenants can customize business logic, user interface, data structure, etc. according to their needs. According to this requirement, the SAAS model of WebGIS system is designed (Figure 2).

The system uses SOA (Service Oriented Architecture), and the bottom layer is the service entity layer, which provides various atomic services. The service assembly layer implements system functions by combining services. Since services are independent of each other, different functions can be implemented in different combinations [2]. Tenants can choose their existing services to assemble or

customize their own service assembly to implement their own business logic functions. At the same time, tenants can also customize the user interface and data structure. End users can choose to use a variety of tenant-customized instances.

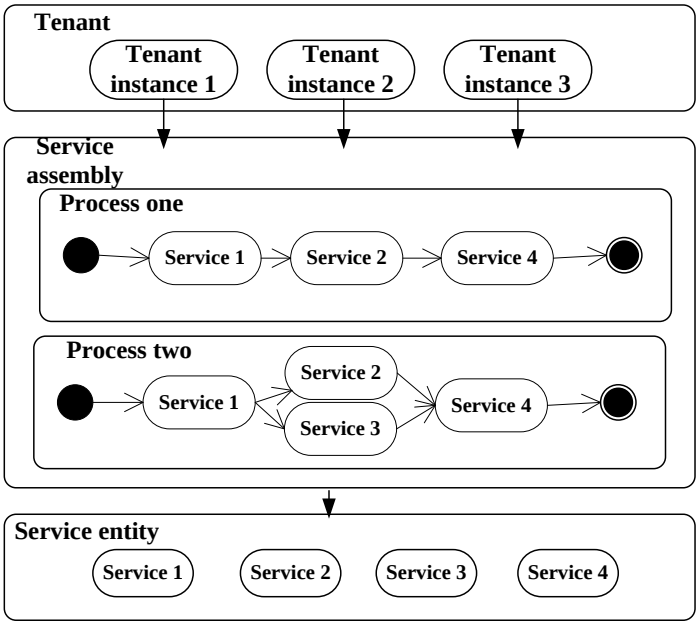

Fig. 2 Model structure

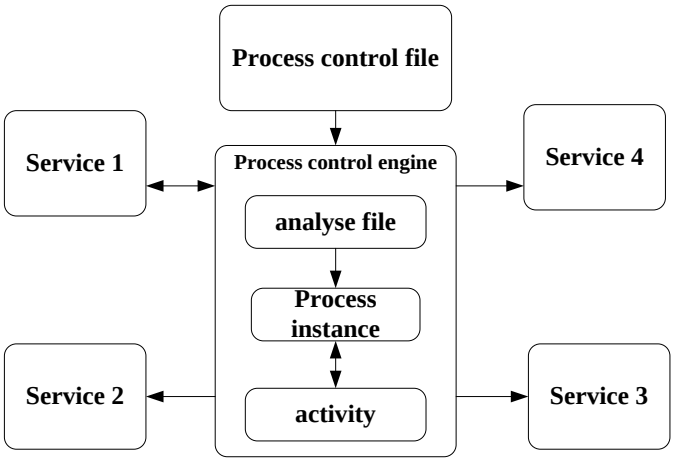

Fig. 3 Service assembly

Therefore, the main points of SAAS mode WebGIS design are: user interface, user data structure, service assembly (business logic) customization.

(1) User interface customization

Customization of the user interface can be achieved through a metadata service. Tenants customize their own interfaces, such as icons, forms, text, backgrounds, table layouts, etc., and store customized interface data in metadata tables. When the user accesses the system, the system displays the interface according to the data of the metadata table.

(2) Customization of user data structure

If each user has a unique database or each user has a unique set of tables in a database, the customization of the user data structure is very simple, and the user-defined table structure is fine.

Considering the effective use of system resources, a multi-user shared a table set of a database, using a metadata service. The method is to keep the expandable field in the shared table set. This field does not directly determine the meaning and data type of the field, but requires a metadata table to

determine. Because metadata can be used as data to describe data, users only need to customize the metadata to customize the data structure.

(3) Service assembly customization

The service assembly can be customized online by the tenant. The tenant first customizes the process control file and then passes it to the process control engine. The process engine parses the file generation process instance and controls the assembly of each atomic service [3]. The process engine is implemented by a Javascript program. Figure 3

A process control file consists of multiple business rules, each of which includes the target object condition, event, and action component. When the target object satisfies the event, the action part will specify what service will be called [4]. After the customization is completed, the process control file is saved to the metadata in the form of an XML file.

In order to simplify the tenant's customization process as much as possible, and to prevent errors. The system is customized by the tenant through the selection of the business rule template. The business rule template is based on several scenarios commonly used in the WebGIS system, and the commonality and variability are identified and generated on this basis.

3.2 System function design

Through analysis, common WebGIS system requirements include: user management; geographic information classification; geographic information addition, deletion, query; map type switching; map tile switching; path calculation; charting; timing update; alarm; UI element management; Customization, etc. [5-6]. Among them, user management, geographic information classification, and geographic information management belong to common needs, and other parts are differentiated needs.

The system uses, for example, as shown in Figure 4, for a total of three roles. The system administrator is responsible for managing the entire system and reviewing the operations of the tenant. Tenants can customize the system functions they need, manage geographic information, and audit user actions. Users can use the system and manage geographic information within their own permissions.

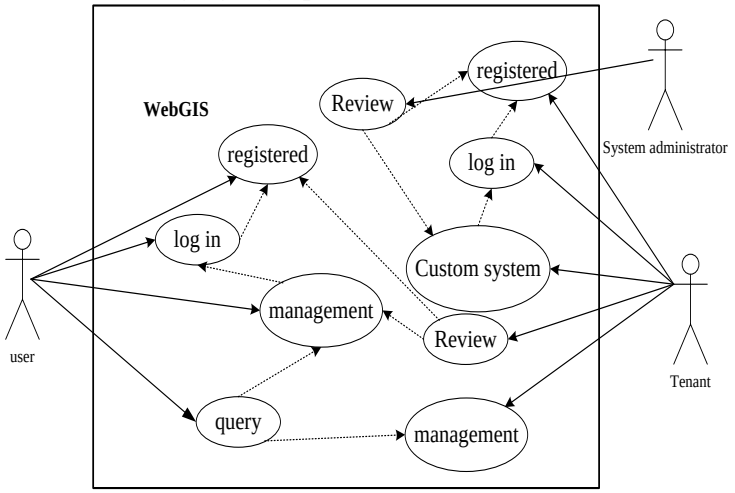

Fig. 4 Service assembly

3.3 Database and tenant customization

In addition to meeting the management of tenants and the management of geographic information, the WebGIS system database also satisfies the customization of tenants. So the metadata service was used at the time of design.

The UI field in the tenant metadata table is to save the tenant-made user interface file; the rules field is a process control file that saves the tenant customization, and the file is a business rule set in XML format.

The Data_str field is a data structure that holds tenant customizations, which is a custom rule set for data structures in XML format. The method is to keep the expandable field in the shared table set. This field does not directly determine the meaning and data type of the field, but requires a metadata table to determine. For example, you can customize the "custom 1" field of the data table "geographic information table" to save the "polygon" field and save a polygonal geographic area.

The tenant DOM object table is a DOM tree object and a service object that store the tenant's home-made interface, and is used by the tenant to select and invoke when the business rule is defined. The DOM tree is saved in Json format. The structure of some of the tables is as follows:

Table 1 Tenant Table

Field Name	Field Type	Description
id	Integer	Tenant number, primary key
name	varchar	Tenant name
password	varchar	password
tel	Integer	telephone number
mail	varchar	email

Table 2 Geographic Information Table

Field Name	Field Type	Description
id	Integer	number, primary key
lbid	Integer	Category number
xlbid	Integer	Small category number
Name	varchar	name
LatLng	varchar	Geographic coordinates
userID	Integer	User id
Zoom	Integer	Zoom level
Tenant_id	Integer	Tenant id
custom 1	Text	Customizable field
custom 2	Text	Customizable field

Table 3 Tenant DOM Object Table

Field Name	Field Type	Description
id	Integer	Number, primary key
Tenant_id	Integer	Tenant id
DOM	Text	File DOM tree in Json format

Table 4 Tenant Metadata Table

Field Name	Field Type	Description
id	Integer	Metadata number, primary key
Tenant_id	Integer	Tenant id
UI	Text	Tenant UI file
rules	Text	Tenant customized process control file
Data_str	Text	Tenant Data file

4. Instance performance study

In order to verify whether the developed multi-tenant mode WebGIS platform can support multiple tenants to execute different business process instances at runtime, and whether different process instances can be isolated, we simulate 400 tenants on the client, and the tenant sets the configuration

plan separately. It indicates that the developed multi-tenant mode WebGIS platform can derive different process instances according to the tenant configuration file, and the execution of each process instance is isolated and does not affect each other. In order to further evaluate the execution efficiency of the multi-tenant mode WebGIS platform, we respectively simulate the nodes, using the JavaScript-based multi-tenant mode and the server centralized mode, and the test instance running time is shown in the figure.

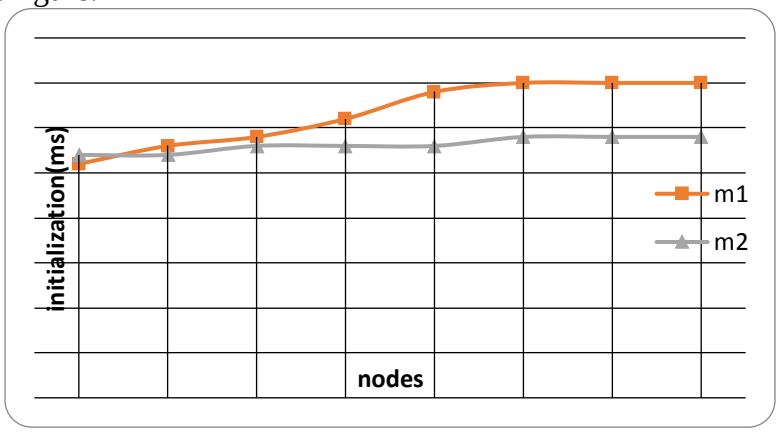

Fig. 5 Initialization time

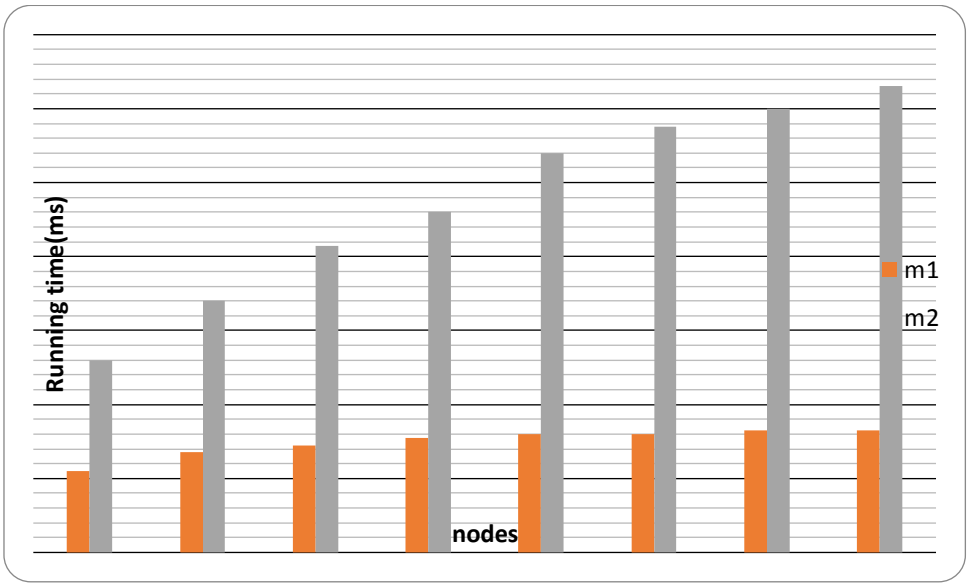

Fig. 6 Running time

According to the example data, m1 represents a multi-tenant mode based on JavaScript, and m2 represents a server centralized mode. When the system is initialized, both modes increase with the number of nodes, and m1 is slightly higher than m2. See Figure 5. After the system running time, the multi-tenant mode is significantly smaller than the server centralized mode, and as the number of nodes increases, It will not increase significantly. See Figur 6.

5. Conclusion

The contradiction between the wide application requirements of WebGIS and the higher cost can be solved by SAAS mode. SAAS mode WebGIS system, the development platform is on the server side, the data is stored on the server side, and the running environment is on the server side. Users do not need to install software, just use the browser to complete the customization and use of the WebGIS system, eliminating the huge cost of purchasing professional software.

The needs of a large number of WebGIS users will not be exactly the same, and it is impossible to design a system that fully satisfies the requirements. Through the statistical summary of various

application scenarios, the common needs and the different needs are counted. These requirements are separately decomposed into a variety of geographic information atomic services to choose from. Tenants can freely combine these services to implement a WebGIS system that meets their needs. Due to the particularity of WebGIS, some of these services are implemented on the server side, and some are implemented by JavaScript on the browser side. The control of the assembly process is implemented on the browser side, and the business rule file is executed through JavaScript. A business rule file consists of one or more business rules. Each business rule consists of "source object (service)" + "event" + "target object (service)" + "action".

The online HTML editor can solve the self-made user interface, and can freely choose to add various services. With the business rules file, tenants can easily complete the system customization.

Considering the economics of the system, choose to use the shared mode to use the database. In order to meet tenant customization and tenant management and geographic information management, the WebGIS system database was designed with metadata services in mind.

After a tenant completes a custom system, a WebGIS template is formed. These templates can be shared by tenants to others. Other tenants simply choose a suitable template and modify it with a little modification. Reduced the threshold for use.

Acknowledgements

This work is supported by CERNET Innovation Project (No. NGII20170627).

References

[1] Y.E.Wang,and H.Liu:Research on WebGIS and Its Architecture, Surveying Engineering, vol. 18 (2009), p.70-73.
[2] J. Guo,and P.Yan:Analysis and Design of Urban Tourism Information System Based on WebGIS, Energy Procedia, vol. 13(2011), No.1, p. 3794-3799.
[3] C.A.Sun,and Z.X.Zhang:A variation model-based reusable and customizable SaaS development approach,Journal of Software, (in Chinese). http://www.jos.org.cn/1000-9825/5294.htm
[4] M.S. El-Kady, and M.A.ElMesmary:Creating spatial database of the foundation soil in Aljouf area using GIS, Innovative Infrastructure Solutions, vol. 3(2018), no.1, p. 1-6.
[5] R. Krishna,J. Iqbal, and A. K. Gorai: Groundwater vulnerability to pollution mapping of Ranchi district using GIS," Applied Water Science, vol. 5(2015), no.4, p. 345-358.
[6] T.T.Wang,and Q.J.Shen: JavaScript debugger software architecture, Journal of Natural Science of Hunan Normal University, vol.37(2014), no.6, p. 88-92.