

Mashup-Technology in Web-Based GIS Design

P. V. Senchenko, Yu. B. Gritsenko, O. I. Zhukovskiy, Yu. P. Ekhlakov

Department of Data Processing Automation, Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia.
Email: pvs@tusur.ru, ubg@muma.tusur.ru

Received April, 2013

ABSTRACT

The paper proposes an architectural model of web-based geographic information system based on mashup-technologies. Represented system is designed using the programming patterns and is used to control access rights to various functions or objects in the system.

Keywords: Geographic Information Systems; Web Technology; Service-Oriented Architecture

1. Introduction

Enterprises involved in the operation of distributed engineering systems have a number of complex issues. Some of these problems are the lack of reliable information on the geographical location of objects subordinate engineering network, and the lack of information about the objects of utility networks belonging to other organizations.

This information is necessary, first of all, for rapid localization of accidents and for preventing their further development (damage of other communications). In addition to these tasks, employees of enterprises solve other problems, such as account, control and analysis of objects, network design and the choice of optimal configurations. Currently, high-quality solution of the above problems without new technologies, which, of course, include geoinformation technologies, seems impossible. Meanwhile, the information that operates with spatial data is usually restricted.

The use of geographic information systems (GIS) in industrial plants that operate utilities, involves the introduction of access control to spatial and attribute data describing objects of engineering infrastructure.

2. Description of Solving Methods

In the publications [1,2] the authors present the efficiency of the web-server approach for the creation of electronic systems of master plans of enterprises that operate engineering infrastructure.

Designing the architecture of a web-based GIS authors have opted for the service-oriented approach [1]. Service-oriented architecture (SOA) suggests the separation of the components, or “services” that are consistent with common interfaces, use common rules for determining

how to invoke services, and how they will interact with each other. “Service” is a software component that can be accessed through a computer network. It implements a function with a well-defined interface.

The authors have designed architecture of corporate web-mashup-GIS which uses service-oriented principles (mashup or mash-up - application that combines content from different sources).

Mashup-application combines data from multiple sources into a single integrated tool. For example, map data of virtual globes (Google Maps, Virtual Earth) can be used as a substrate, KML-flow data providing specialized map service can be used as thematic layers, and for the addition of arbitrary attribute data it is necessary to refer to the management of information services descriptions. The result is a new, unique web service that was not originally proposed by any of the sources.

Mashup-application’s architecture always has three parts:

1) the provider of content is data source. The data is available through an application programming interface (API), and a variety of Web protocols, such as RSS, REST and Web Services;

2) mashup-site is a web-based application that offers a new service using data source that does not belong to it;

3) the client’s browser is the user interface of mashup-application.

Mashup-basis of the application is a set of components – widgets. They are small software modules that have a graphical user interface and application programming interface (API), that support the event. The main objective of the widget is to integrate web-based service to the application (do mixing). Integration of mapping services via widgets is the most effective approach to implement Web-based corporative GIS within the SOA.

The architecture of the developed system is based on the several programming patterns:

- **Model-View-Controller.** The server side is committed to MVC. View and Controller components are incorporated in the controller. The model is divided into several levels [3].
- **FrontController.** According to the pattern a unique entry point to the application is set-/index.php (an alternative approach is the use of a separate script - login.php, order.php etc for each specific operation) [4].
- **Intercepting Filter.** Package FilterChain LIMB is an implementation of “Intercepting Filter” pattern. It allows to concentrate all the logic in one place. In conjunction with the pattern Front Controller all work of applications confined to a chain of filters. The filters are registered as “handle”, which become real objects only when it is their turn. “Handle” is a proxy object that acts as an object it replaces and which loads (initialize) proxied object at the first real need. Each filter decides whether to pass control to the next filter or not.
- **Dynamic Service Locator.** The essence of the pattern is that there is an easily available object, which is a common place to access to all the popular objects (services) and the capabilities of this object can be easily expanded.

3. The Results Obtained

Developed by the authors generalized architecture of web-based geographic information system is shown in **Figure 1**.

The system is designed as a three-tier Client-server architecture: client, Web application server, database server. Users of the system can use popular web browsers: Internet Explorer, Opera, Google Chrome, Mozilla Firefox. The user-http-request is received from the client to the web application server.

All requests to the web application server are redi-

rected to the file Root/index.php. This creates an object of class WebApplication. The object has methods of supply chain management filters that control the execution of the request: registration (adding) of the filter, the call of the first filter in the chain, the call of the following filters.

When an instance of WebApplication is being designed, filter chain is initialized, and control is passed to the first filter (**Figure 2**).

Using a filter mechanism allows to control user access rights to various functions or objects in the system. **Figure 3** shows a flow diagram of client requests with the ordered list of filters:

- 1) exception handling;
- 2) creation of the session and connection the Driver Database;
- 3) checking the existence of the requested module and the right to access it;
- 4) review of the http-request (choice of the controller and its actions to process the request, creation of an instance of the controller);
- 5) performance of the selected action of the controller;
- 6) reply generation;

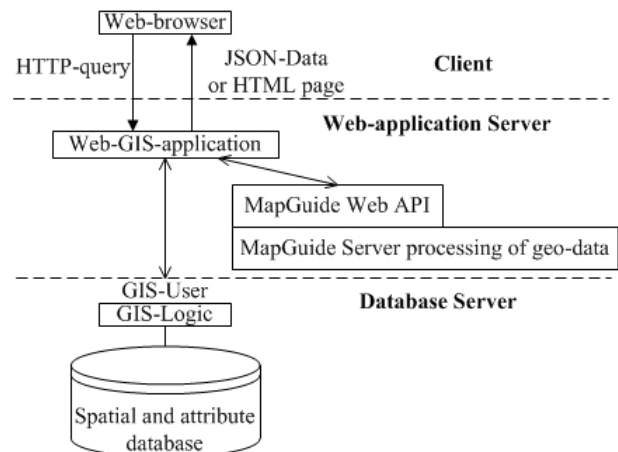


Figure 1. Web-GIS generalized architecture.

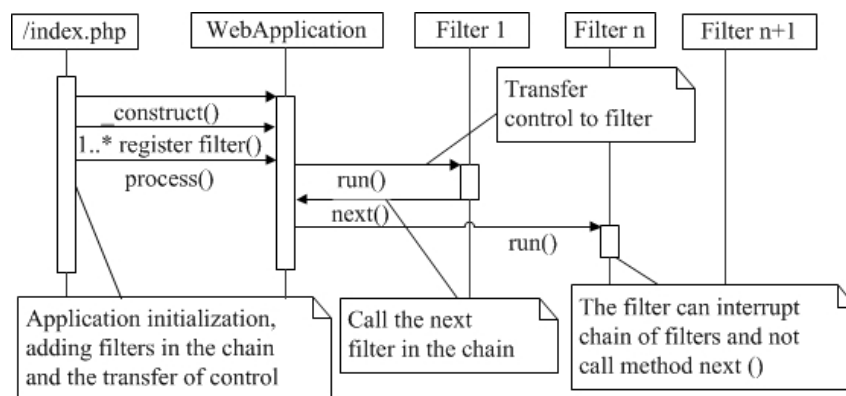


Figure 2. Demonstrates Implementation pattern “filter chain” in Web-GIS.

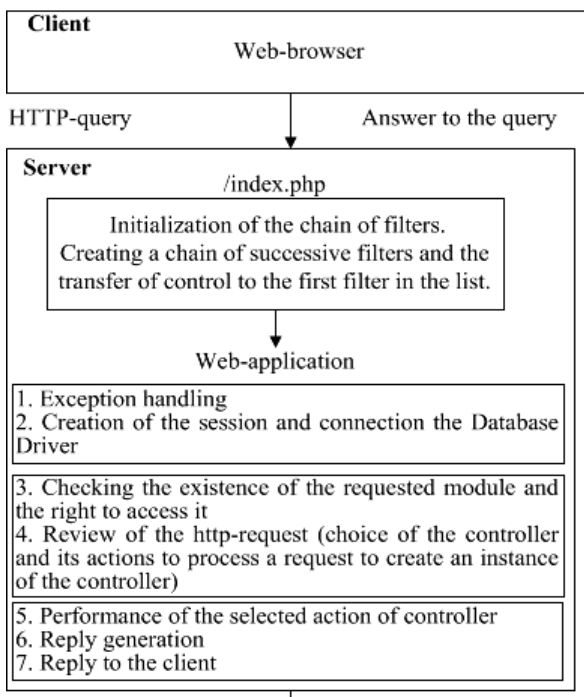


Figure 3. Client request processing scheme in Web-GIS.

7) reply to the client.

The controller divides requests into two classes: a database query or request of geoprocessing. Integrated service MapGuide Web API is responsible for processing geodata. Database query is passed to the third level - the level of the database server. Controller's operation on interaction with the model goes through a set of tools "Toolkit". The toolkit refers to the model through the objects of class Tier (data and functions of web-based geographic information system) and MgTier (communicates with the server through the MapGuide MapGuide Web-extensions).

The framework Zend_ACL (Access Control List) is used to manage privileges. For more information on using the framework one can refer to the official website Zend Framework [5].

ACL is used to control access to certain objects (roles) to other objects (resources). Roles request access rights to resources, the application performs access control by ACL.

To use the object as a resource object one must implement the interface ZendAclResourceInterface, which contains a single method - getResorcedId () - and returns a unique identifier of the resource. Similarly, the role must implement the interface ZendAclRoleInterface, containing method getRoleId ().

The system also provides base classes ZendAclRole and ZendAclResource, that implement the appropriate interfaces. All created roles are stored in an array (field _roles class ZendAclRoleRegistry).

4. Findings

Using a mashup-technology and software architecture makes it possible to develop software of web-based geographic information system with the following features:

- Keeping the rules of access to spatial data and the functional component of Web GIS.
- Organization of role regulated access to the data of master plan (objects, groups of objects, functions and territories).
- User registration in a Web GIS.
- User-defining as a group member.
- Assigning of the group members with the access to spatial data on a set of layers, on the territory of this region of the general plan, the type of engineering network.
- Assigning of the group members with the access to the technological capabilities of the system, represented as a function of Web-GIS client.
- Audit of users in the electronic environment master plan. The major controllable actions are the creation of spatial objects, view information about objects, edit spatial and attribute data, deleting data.
- Identifying the layers of electronic master plan to include in the audit mode.
- Assigning users and groups rights to user interface elements Web-GIS client: GIS layers, information reports, GIS functions.
- Two modes of operation to the parameter assignment of access rights:
- Auto assigning rights through objects. It is recommended if you want to assign multiple roles of the right to a single object.
- Auto assigning permissions through roles. It is recommended if you want to assign a role of law for several properties.
- Inheritance of access privileges on the parent role.
- Formation of a query to filter the data according to the user's session with its current access rights.
- Reporting of users access the spatial data warehouse objects and actions on these objects for a specified period of time.

The material in this article was developed with the financial support of the Ministry of Education and Science of the Russian Federation in the framework of the event 2.4 Federal Program "Research and development on priority directions of scientific-technological complex of Russia for 2007-2013", the state contract № 07.524.11.4013.

REFERENCES

- [1] Y. B. Gritsenko, Y. P. Ehlakov and O. I. Zhukovsky, "GIS Technology Monitoring Engineering Networks: Monograph," Tomsk: TUSUR, 2010, p. 148.

- [2] Y. B. Gritsenko and O. I. Zhukovsky, "Typical Architectural Solutions GIS Engineering Infrastructure of the Enterprise," Reports of the Tomsk State University of Control Systems and Radio Electronics, 2011, Vol. 2, No. 24, pp. 171-175.
- [3] Model-View-Controller. 2012. <http://ru.wikipedia.org/wiki/Model-View-Controller>
- [4] Front Controller pattern. 2012. <http://en.wikipedia.org/wiki/FrontController>
- [5] Zend Framework. Programmer's Reference Guide. 2012. <http://framework.zend.com/manual/ru/zend.acl.introduction.html>