ARCHITECTURAL FRAMEWORK FOR DEVELOPING COMPONENT BASED GIS SYSTEM

Dinesh Singh¹ and Aman Jatain²

¹Department of Computer Science, M. D. University, Rohtak
²Department of Computer Science, Amity University, Gurgaon

ABSTRACT

Component Based Software Engineering has one main sole motive of making the development process of software systems as easy as possible and to achieve this objective work needs to be done in previous systems to identify the concerns and limitations which can be overcome using this software engineering based approach. In this paper to support concept of component based system a domain is chosen that covers the GIS systems. GIS (Geographic Information Systems) are commonly used for development of map based applications these systems are widely used across the web and in various organizations. With the development and deepening of GIS, traditional GIS systems showed the challenges on isolation, sealing, interoperability and the limitations, thereby hindering further development and application of GIS technology. In this paper framework for component based GIS system is proposed. This framework is having rich graphical interface and user data can be easily retrieved from the connected database and displayed in the browser.

KEYWORDS

Architecture, Component, Geography Markup Language, Geographic Information System.

1. INTRODUCTION

Software reuse is the use of already developed software system to construct new software systems. It is one of the upcoming fields in the area of software engineering because it increases the productivity by reducing the cost and time in developing a software system from scratch [11]. Software reuse enhances the quality by making the software system more reliable and flexible and thus developed software system using this technology is easy to maintain and upgrade [19]. The goals of software reuse are realized by Component Based Software Engineering (CBSE), which is also known as Component Based Software Development (CBSD). CBSE focuses on the development of software system using existing reusable software components. The basic idea in this approach is to develop a software system by selecting reusable components from repository and then assemble them into a well-defined architecture. The main motive behind component based development is to develop a set of standardized software components that can be fit into any architectural within application domain and application is then assembled using these prebuilt components. A software component can be software, a web service, a web resource or a unit that constitute a set of related functions or data. In this paper domain chosen for implementing the concept of component based system is GIS systems. The need is to develop a desktop like GIS application that provides a rich user interface and visualization. The user can view data stored in database and can view that data on a map. The technologies used in the proposed system support easy access with database and integration with other frameworks.
In literature many techniques using component based GIS architecture are discussed to build Geographic Information System (GIS). In GIS architecture, the GIS system is divided into various levels that make development easy. In each level there are different components connected through interfaces through which interaction takes place among these levels. Component based GIS system ComGIS is divided into three levels: the view level, the implement level and the control level. In some approach, the GIS is also modelled using UML, that defines the various attributes of GIS and open GIS system architecture is used for developing component software environment of GIS. Each service entity represented as a component unit and assembled as part of GIS system. In this architecture a component is defined as a collection of several essential service entities. Each service entity is corresponds to one service in the GIS system. The components in this architecture are modelled using UML and are developed in object oriented computing environment, COM (Component Object Model) and Visual C++. A particular application domain needs a vast amount of data to identify components and a domain oriented approach is used for GIS component selection. Here GIS component information is managed by collecting and comparing information from various sources. The methodology used in this framework is consist of entities which correspond to the main concepts of GIS component domain. The structure of component based Enterprise is based on GIS spatial database engine SuperMap, SDX+ and implements database into SQL server and Oracle relational database and design integrated spatial database.

An implementation of WebGis is a prototype called FreeXGIS which is based on Distribute components describes a development approach of the system using EJB, Distributed Component Object Model (DCOM) and XML. FreeXGIS is a hierarchical architecture and architecture is divided into four layers: database layer, component management and presentation layer. The main objective here is on integration of components that are likely to have developed separately and even independently. Component identification, customization and integration are crucial activities in the life cycle of component-based systems. The application is designed using a java based AJAX technology called Google Web Tool Kit or GWT, which is a set of open source tools. In addition to this an open-source library, Openlayers is used to display a map in this GWT application. The tools used in this system can be easily accessed and used in a single IDE (Integrated Development Environment).

2. ARCHITECTURE OF GIS SYSTEMS

Depending on the way how the project is developed different architectural patterns might be used. One of the patterns that have been depicted in Figure 1 is the Model View Controller. This pattern is mostly used on many different types of platforms including Java. All the three parts of this model are strongly interconnected with each other and communicate with each other by sending requests, data or commands. This pattern provides the possibility to split parts of application logic and the interface. The open source tools that have been used in this project are called GWT and they are based on Java. This pattern supports this set of tools. In case of the GWT application, Model is partly responsible for handling the business logic and also responsible for handling calls to the server using GWT-RPC. GWT-RPC is an important feature of GWT as this feature supports communication with the server using RPC (Remote Procedure Calls). Thus providing the advantage to the user to develop rich web based internet applications where user can integrate the application with other java frameworks and server side solutions.

The Model notifies any changes that have been made in the data to observer. The View is the user interface that has been built up using GWT widgets or other similar libraries like GWT-Ext or
Smart GWT. It reacts to any changes that are made in the Model and dispatches the user actions to the Controller. The Controller is responsible for handling events sent from the View and according to them it can modify the Model. Each part of this pattern communicates with each other by sending messages requests or commands. The data can be exchanged by issue of commands or sending request to the respective part which is responsible for handling that data.

Another architectural model as shown below in the Figure 2 which can be used is Model View Presenter. It is a modification of MVC. In this model instead of the Controller, the Presenter is the used in the middle of Model and View. The Presenter gets the all the events from the View, and according to which it performs the required actions on the Model, and then updates the View according to those changes in the Model.

In this type the model is strictly a domain model as all the responsibility for handling the presentation logic is managed by the Presenter. Here the presenter acts upon the model and the view.
2.1 Components of GIS Systems

The best representation of a GIS approach can be achieved by breaking application functionality into modular and flexible components each component providing some different services. These services can be integrated, reused and invoked independently of the technology that is being used. Also they can be used to create more sophisticated services. The proposed approach towards interoperability is a GIS application. GIS application is having three tier architecture which consists of presentation tier, service tier and database tier. In this architecture a deployment style is used where the functionality is separated into layers.

A) Presentation Tier

The presentation tier is at the topmost level of the application and provides the application user interface. It accepts user inputs and displays results to the user. This tier is usually represented by a Web browser or mobile application. The user interface is composed of widgets, various controls such as text box, buttons, drop-down lists and a “slippy” map that is embedded in the application. This tier communicates with other tiers and retrieves the result for clients. With the help of the GWT a rich interactive user interface can be designed by using the GWT widgets and each button textbox or other GWT widgets have events associated with them which are fires when any activity is seen like clicking a button or entering data in textbox or selecting an item in a list. The presentation tier can be mode more attractive by use of CSS (Cascading style sheet) which is by default included in a GWT application when user creates any GWT application. The CSS follows a format which is used to make the buttons, labels, dialog panel and other GWT widgets more attractive to look on the browser. Each GWT application has a HTML and CSS file included with it which helps the user to customize the application to appear according to the user and each widget is displayed according to as designed by the user.

B) Service Tier

The service layer is represented by the client-server application architecture and consists of a Web server. It controls the application functionality by taking user requests, performing processing and pulls out results to the clients in the form of tables and a map. It receives input information from the user, connects to the database tier, retrieves and processes information then sends it back to presentation tier. The service tier is responsible for communication between presentation and database tiers. The data is retrieved from the database with help of Hibernate framework. In the applications GWT integration with hibernate framework is used to retrieve data from database. The Hibernate configuration file includes the necessary information to connect to the database like the JDBC driver to be used and the username or the password for the database. This file gives information that the application has to connect to which database and the name of column in which the data is stored.

C) Database Tier

The database layer consists of data stored by the user in form of tables, in this application database HSQL (HyperSQL Database) or MYSQL can be used. The advantage of HSQL is that it is written in java. It incorporates the advantage of a small, fast multithreaded and transactional database engine and supports embedded and server mode. It also consists of a powerful command line SQL tool and simple GUI query tools. We have used HSQL and MYSQL. According to the user needs any database can be used but mostly the HSQl database is used because it stores tables in memory which can be accessed quickly but users who are more familiar with MySQL or
other databases like Oracle or PostgreSQL can use that database also. Hibernate framework is used to manage all the intricate details of connection with the database so as user need to remember the steps to connect to database and access data.

3. RESEARCH METHODOLOGY

The proposed work is about implementing a GIS system based on opensource technologies and integrating these technologies to retrieve data from database and display it to the user. This system employs the use of GWT Framework in Java and an IDE (Integrated Development Environment) where these technologies are integrated.

3.1 GWT

Google Web Toolkit (GWT), a set of development tools, programming utilities, and widgets that allows the user to create rich Internet applications that are very different from the web applications that are built using javascript or other different tools. The major difference between GWT and all the other frameworks that are being used is that using GWT the user can easily write the browser-side code in Java instead of using JavaScript, thus providing the advantage to java developers to code in a language they are quite familiar with and in any IDE using different frameworks. Thus users that have an understanding of Java can easily build Ajax based internet applications without the need to learn the traditional style of JavaScript coding. Thus the user gets access to the Java development tools that are already available.

Google provided GWT, allowing these rich web based applications to be written in Java, by just using the tools that are already existing. As the size and complexity of rich Internet applications is increasing to a large extent there is a need to develop applications easily using concepts and tools that are already familiar to the developer and the user. The large internet applications are very difficult to manage, and Java was designed to make the process of developing large applications manageable. GWT brings all the benefits of Java to rich Internet applications and it also allows the user to interact with the existing JavaScript code. If the user wants to use GWT the user can also integrate it with the javascript code if user is familiar with JavaScript coding hence making GWT flexible and also it can be integrated with the existing server-side services.

3.2 Features of GWT

GWT is a Java-to-JavaScript compiler that produces code capable of running on different web browsers like Internet Explorer, Firefox, Mozilla, Safari, and Opera. The GWT provides a compiler which converts the Java syntax to the corresponding JavaScript code, and it utilizes the JavaScript versions of the Java classes that are used commonly like Vector, HashMap, and Date. GWT also includes a large library of widgets and panels, thus reducing the effort to build a web application that looks similar to a desktop application. The widget library includes the usual suspects like text boxes, dropdown menus, and other form fields. Also it includes some complex widgets like a menu bar, tree control, dialog box, tab panel, stack panel etc. GWT provides different tools for communication with the server. Like firstly it includes several wrappers of different complexity and capability around the JavaScript XML, HttpRequest object, which is an object that is most often associated with the Asynchronous JavaScript and XML (AJAX) development. Another tool provided by GWT is a set of classes which support the JavaScript Object Notation (JSON) message format. JSON is a popular message format which is best known for its simplicity and the widespread availability.
3.3 Why GWT?

GWT also supports a tool that enables Java objects between the browser and server without the need to transform them into an intermediate message format. This tool allows the user to access the server side services that are written in any language and enable it to integrate with other main Java based frameworks: Java- Server Facse (JSF), Spring, Struts and Enterprise Java Beans (EJBs). This flexibility means that GWT doesn’t increase the amount of work; instead, it makes it possible for the user to continue to use the same server-side tools that user uses normally and that are well known to the user. But it is not very easy to write these rich Internet applications in Java. GWT also provides the support for a JUnit testing framework and a special hosted-mode browser that allows the user to develop and debug applications in Java without ever needing to deploy the code to a server. This saves the time and also the GWT-RPC (Remote Procedure Call) mechanism allows the user to send Java objects between the client and server with only a little more work on both the client and server sides. Also there is a great GWT Openlayers wrapper which allows the user to easily view the geographic data in a GWT application. The user can display and view maps in GWT applications by including the GWT openlayers jar file in their GWT applications. The javascript functionality of the openlayers API can easily be accessed throughout GWT by inheriting it in the GWT XML file.

3.4 GWT Widgets

The below figures depicting individual GWT widgets have been referenced from the official GWT website [http://www.gwtproject.org/doc/latest/RefWidgetGallery.html](http://www.gwtproject.org/doc/latest/RefWidgetGallery.html)
3. EXPERIMENTAL RESULTS

Here figure 3 shows the results in the form of screenshots which are being taken when the application is running

A) When we select a checkbox to view contact information

Figure 3 contact information
This result is displayed when for the first time application is run, in the screenshot add and delete buttons provide the ability to user to add new or delete any contacts:

**B) When we run the application a list is displaying containing all the contacts.**

![Figure 4 List of contacts](image1)

This result is displayed to display the contact data on the map provided by the OpenLayers library.

**C) When user clicks on the show button contact information is shown on the map.**

![Figure 5 Map showing contact information](image2)
4. CONCLUSION AND FUTURE WORK

This paper provides a view of the GIS systems and the need to develop a GIS application using open source tools and libraries. Thus after getting a deep understanding of the requirements of the system and to provide an interactive easily accessible understandable user interface GWT fulfils the need for a suitable application framework for a GIS system that can be divide into components. In GWT each part provides different functionality, the client and server responsibility is structured and separated hence using RPC mechanism the goals can be accomplished. Also GWT can be integrated with hibernate framework using DTO to access data stored in the databases. The open source OpenLayers library provides easy access to geographic data and data can be easily displayed in the GWT application using GWT OpenLayers wrapper library. Hence the questions which were posed while identifying the research problems are answered in this proposed system. This application can easily be built using open source technologies.

The proposed framework focuses on use of open source tools and libraries and it can be extended by integrating a spatial database having spatial datasets. This database can be populated by collecting data for a single domain like traffic data or data of pollution areas. Also an open source server like Geo server can be integrated with this database to collect data from database and display it to the user in the GWT application.

REFERENCES


