

An e-Learning System Based on GWT and Spring MVC

Bo Song

College of Software, Shenyang Normal University
Liaoning Shenyang 110034, China
e-mail: songbo63@126.com

Yejun Zhang

College of Software, Shenyang Normal University
Liaoning Shenyang 110034, China
e-mail: yejun1108@126.com

Abstract—A solution is presented which integrate the GWT and Spring MVC in the e-Learning system in this paper. The client-side is provided with widget support based on Ajax by using GWT, and the Spring MVC layer the project, make the project easy to maintain. Hibernate using the database configuration information to provide persistent services to application, and the persistent object. Google Web Toolkit (GWT) and Spring MVC are two frameworks that used in the web application. GWT is one of the sharp spike for Web UI development. Spring MVC and Hibernate as one mature rapid business application development architecture free US from the complicated business issues. Finally, the practice shows that the method presented by this paper enable the architecture to effectively improve the real-time requirements and human-computer interaction experience of e-Learning system.

Keywords- *e-Learning; Web 2.0; GWT; Spring MVC; hibernate; framework*

I. INTRODUCTION

For e-Learning system, the main benefit of Ajax is a greatly improved user experience [1]. Although JavaScript and DHTML — the technical foundations of Ajax — have been available for years, most programmers ignored them because they were difficult to master [2]. Although most of the Ajax frameworks available today simplify development work, you still need a good grasp of the technology stack. So, if you're planning to use Ajax to improve only your application's user experience — So, if you're planning to use Ajax to improve only your application's user experience — if you're not also using it as a strategic advantage for your business — it may be unwise to spend a lot of money and time on the technology [3].

Existing systems focus on the business data processing. It is difficult to adapt to variable business rules, business rules are difficult to express it. More is a simple test and data storage, display, cannot assist business logic to handle. Due to the system level is not clear, there are some disadvantages, poor reuse of software components, development, maintenance process is complicated; Operator switching between different systems and the operating interface is not friendly [4]. Aiming at the above-motivated problems, a solution of e-Learning system is proposed in this paper. Using the foreground and background interactive mode called by Remote Procedure Call (RPC) [5] that is offered by GWT, the new solution carries on the decoupling between rich client and server-

side, thus forming a kind of hierarchical and extensible e-Learning system framework. On this basis according to the integration of GWT and Spring MVC, not only are the cost and complexity of e-Learning system reduced significantly but the human-computer interaction experience and real-time requirements are improved effectively.

II. GWT AND SPRING MVC

Google offers another way to help Java developers create Ajax applications more productively. This new framework, called GWT. It uses Java as a single programming language for both the client and server sides. The GWT eliminates the complexity of hand coding HTML, JavaScript, and other elements that bind the JavaScript actions triggered in an HTML page (client-side code run within the user's browser) to the controlling layer executing within the Java EE application server (server-side, where the database resides). When developed within a good IDE, the GWT can provide an immeasurable advantage over other frameworks and considerably lower the cost and complexity of the project. And the relative simplicity of the language helps traditional developers become accustomed to the latest technologies [6].

Spring's Web MVC framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale and theme resolution as well as support for upload files. The default handler is a very simple Controller interface, just offering a ModelAndViewhandleRequest (request, response) method. This can already be used for application controllers, but you will prefer the included implementation hierarchy, consisting of, for example AbstractController, AbstractCommandCon-troller and SimpleFormController. Application controllers will typically be subclasses of those. Note that you can choose an appropriate base class: if you don't have a form, you don't need a form controller. This is a major difference to Struts [7].

Spring MVC integration in Spring Web Flow. The Spring framework provides a full-featured MVC module to build Web applications. Using the Spring MVC architecture which can be inserted .Through policy interface, the Spring framework is highly configurable, and contains a variety of view technologies, Spring MVC separates the controller, the model object, dispatcher and the role of handler object. This separation makes them easier to customize. MVC is the core part of the WEB

project development, as the decomposition of the three words, C (Controller) separate V (View, Client) and M (model, business) constitute MVC. The Spring annotations will make our work easier, by using the reflection mechanism of annotation is to solve the problem of a large number of configuration. In the configuration file with simple statement, can easily deal with all the request control project work. When using Spring MVC to do Java Web project, using the Spring framework's built-in filter CharacterEncodingFilter, to solve the problem of the garbled [8].

Dispatcher servlet binds Spring MVC together. This XML configuration file will be the one that register as our servlet in web.xml, and it's the one that will receive all the HTTP requests to the site. Its responsibilities are as follows. Firstly, read the incoming request and delegate it to the appropriate Controller. Secondly, the controller resolve the view, means turning a request for the index view. Finally, the controller will then tell the resolved view to render the output by passing the model to the view.

III. IMPLEMENTATION OF THE E-LEARNING SYSTEM

As Fig .1 showed, a kind of hierarchical and extensible e-Learning system is proposed in this paper. In this Fig .1, the rich client is deployed and run in the user's browser, and the layers of server-side is deployed and run in the Java EE application server, and the communication mode is Ajax between the two. The rich client is provided with widget by GWT based on Ajax. The framework proposed by this paper makes full use of the characteristic of GWT technology and obtains further depuration and encapsulation, which enable the framework more suitable for the design and development of specific e-Learning system.

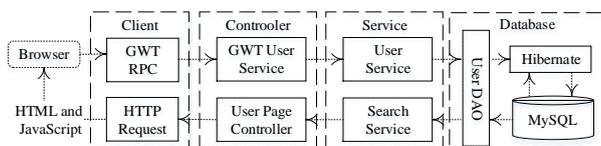


Figure 1. The framework of the e-Learning system.

In Fig .1, it's the framework of e-learning application. The GWT services straddle the line between controllers and services, since they basically provide the HTTP processing capability of a controller and perform some GWT serialization work on the results from the service layer. The DAO layer contain the code make the results of our business processes persistent and query the database.

All business logic should be kept in the service layer because this is the place where it will be most reusable. Service objects are simple POJOs. The HTTPRequest-object used in a controller. Similarly, to keep the object relationship mapping (ORM) code in a DAO layer. This serves two purposes. First, it improves the testability of service layer. Accessing the database directly from within the business methods, we need to access the database when we unit test these services, which will complicate our unit tests considerably. With an injected DAO layer, to inject mock DAO objects instead, which will make unit tests far

easier to do. Additionally, if we ever need to switch database or decide to use a different ORM tool, we'll have an excellent buffer to isolate the effect of these changes.

A. Rich UI Client-side

The client presentation layer is in charge of downstage interface display of e-Learning system, the implementation of which includes two parts: one is HTML, CSS, photograph and etc. that do not need GWT compiling; the others are Java interface codes that need be translated into JavaScript. The former is similar to the development of B/S mode, and the latter is mainly responsible for the management and use of GWT UI widget.

In order to embody the technical features of rich client-side, the functions of data cache and logic calculation should be provided in the client logic layer of client-side. Making full use of the computing capability of client-side can improve the response time of e-Learning system and reduce the load of server-side. The reusability and maintainability of client-side application can be improved significantly by designing the tool class of e-Learning system to implement the above functions.

Service calling layer is responsible for calling service offered by the service layer of server-side and returning the result to client logic layer and client presentation layer. In other words, the service calling layer decouple between foreground logic and background logic. According to each YourService in this layer, a YourServiceCaller class is designed. Both the construction of the service object and the access of operating the service object are encapsulating, which enable client-side to call the service by YourServiceCaller class. In doing so, it is implemented to centralized control of background service access in order to maintain the code.

B. Service Layer

The class diagram of the GWT RPC service extended framework about e-Learning system is shown in Fig .2. By extending the RPC framework of GWT, e-Learning system not only implements the communications between the service calling layer and the server-side, but accomplishes the uniform control of the whole system service. According to the GWT RPC extended framework, as shown in Fig .2, the sequence diagram (Fig .3) below presents a typical flow of calls between the GWT RPC mechanism and MySQL Database.

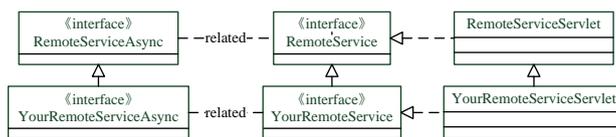


Figure 2. The class diagram of the extended framework about GWT RPC.

C. The interaction in the system logic framework component

In the design of the architecture, the business developer only needs to develop five types of code: UIController, UIWidget, Action, DAO and Entity. UI using the GWT framework, provides the UI development "UI management components" and "UI Widget components". The UI

management components have cache management. The object that mainly manage is: the current user information, system constants and the dynamic parameters that system commonly used and so on, to avoid the frequent transfer data in the foreground and the background. "UI Widget components" has tree components, dialog operation, paging box components and drawer components, etc. Developer's use these components can quickly develops business module, not only improves the speed of development, also is beneficial to the accumulation of component in the future, reuse and maintain. "UI Widget component" is mainly responsible for rendering and input control interface, it does not call the background service. "UI Controller components" is mainly responsible for interactions with the data of the background services, management and scheduling Widget component.

"Action components" provide the business logic implementation, such as: the saving of announcement information, save announcement information first, and then submit for review. Action must use the interface to implements the separate design patterns, and to follow the AJAX of GWT calls specification, provide an asynchronous interface. Need to register in the Spring configuration file.

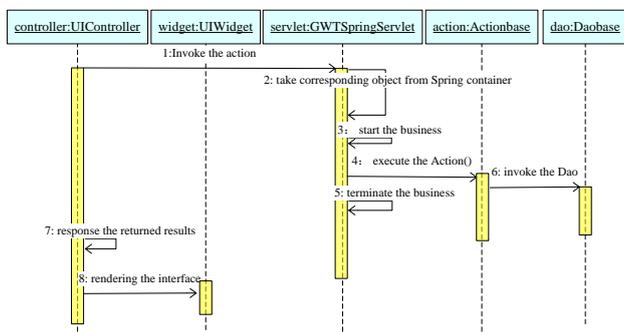


Figure 3. The sequence diagram of the e-Learning system.

"GWTSpring integration components" implemented will GWT integration with Spring framework. In the Fig .3, GWTSpringServlet is "GWTSpring integration components", it as a WEB application service unified listening Servlet, UIController calls to back-end services are made by GWTSpringServlet route by and forward. GWTSpringServlet contains calls to the Spring container packaging, in a WEB application server startup, read the Spring configuration file, to initialize the Spring container.

"GWTSpring integration components" implements the integration between the GWT and Spring framework. GWTSpringServlet is "GWTSpring integration components", it is a unified listening Servlet in WEB application service, UIController calls to back-end services are made by GWTSpringServlet route and transmit. GWTSpringServlet contains call and encapsulation to the Spring container, when the WEB application server startup, read the Spring configuration file, to initialize the Spring container. When UIController send the asynchronous AJAX request GWTSpringServlet inform the Spring container and instantiate the relevant Action .When the spring initialize a Bean instance, according to the Spring configuration file structure an instance, inject the dependent objects, such as: DAO or other Action. Before the call Action method, the Spring container will

determine to start a transaction according to the configuration.

After the call method of Action terminate, Spring container will determine to submit an effective transaction or rollback transaction according to the configuration and code execution. Therefore, it is not necessary to write any sentences about transaction in the business development code. After UIController send the Action call request, the user can operate other interface, browser receives the results returns from Action, automatically send to the UIController. UIController getting the results, rendering the UIWidget component interface.

D. Hibernate and GWT Serialization

Hibernate is an open source object-relational mapping framework, It has a very lightweight object encapsulation for JDBC, Java programmer can manipulate the database with object programming thinking. Hibernate can be used in any situation which use the JDBC .It can be not only used in Java client program, but also in the Servlet/JSP Web application. The most revolutionary is that Hibernate can replace the CMP in the J2EE application architecture which apply EJB, to complete data persistence [9].

Using XML mapping document for each Java class. Within that document, specify the name of the database table to map to and the columns that our object properties will be saved in. Hibernate use these mapping to automatically convert our Java objects into SQL select, insert, and update commands. For basic object properties, this is simple. Things get a little more complicated (but also more valuable) when we start letting Hibernate take care of persisting more complicated relationships, such as those for Map and Set.

The GWT class named *ServerSerializationStreamWriter* class has an enumeration that matches incoming objects to the appropriate serialize. Listing 3-1 shows the first bit of this from the basic GWT class.

```

CLASS_TO_VALUE_WRITER.put(long.class, new ValueWriter() {
    Public void write(ServerSerializationStream Writerstream,
        Objectinstance) throws SerializationException {
        stream.writeLong(((Long)instance).longValue());
    });
  
```

This Map collection is responsible for writing out the serialized value of incoming objects to the stream. So we can't extend it. Being reduced to reworking the class is unfortunate but it works. Adding a method such as the one in Listing 3-2.

```

public void setValueWriter(Class<?> clazz, ValueWriter writer) {
    CLASS_TO_VALUE_WRITER.put(clazz,writer);
}
  
```

This method change ValueWriter into one of our own that executes our HibernateFilter. GWTCController code in Listing 3-3.

```

private ServerSerializationStreamWriter1941 getWriter(Serialization
Policy:serializationPolicy) {
    ServerSerializationStreamWriter1941 writer=new ServerSeria
lizationStream Writer1941(serializationPolicy);
    writer.setValueWriter(Object.class, new ValueWriter() {
    publicvoidwrite(ServerSerializationStreamWriter1941 stream,
  
```

```

Objectinstance)throwsSerializationException {
    stream.writeObject(HibernateFilter.filter(instance));
}
});
return writer;

```

When GWT Controller is asked for *ServerSerializationStreamWriter*, create one of our own design, and set the *Objectstream* writer to one that enacts *HibernateFilter* before the objects are serialized. That's all need to do to prevent GWT deserialization from exploding when it deals with Hibernate objects.

IV. FOOTNOTES

It is simulated that the client-side submits static and dynamic request to JavaEE application Server by browsers in real environment through the Grinder, in this paper. The testing scenario is information query in the first page of system, which can relate to view switching, curriculum screening, sorting, and pagination etc. The performance testing highlights the Java class in the lasting unit. The aggregate average response time (AART) of the system is shown as Fig .4.

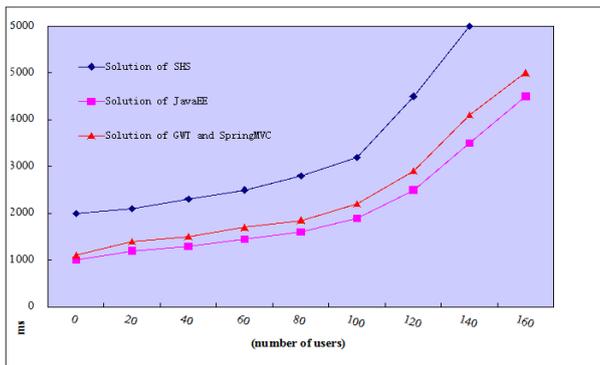


Figure 4. AART curve.

When analyzing the AART curve in Fig .4, compared with the e-Learning system solutions of [4] and [10], we can find that the system response time of the solution proposed by this paper may increase according to the curve until it reaches 100 users. After that time, the increase of the curve changes more acutely. Through analyzing the changing tendency of curves, in the performance test, the application's maximum limit on the number of users is 100, so it meets the needs of the application service. The solution of GWT and Spring MVC is close to the Solution

of JavaEE AART. This solution is a very effective solution. It has high development efficiency, and it's easy to manage and maintain the E-learning system at the same time. Finally, in this paper how to improve e-Learning system construction is also discussed which has some guiding significance for the application and development of the e-Learning system construction in the network teaching field.

ACKNOWLEDGMENT

This work was supported by the Science and Technology Project of Education Department of Liaoning Province (Research of e-Learning System of Small and Medium-Size Enterprises Based on SaaS, No. L2013417).

REFERENCES

- [1] Michelle Pieri and Davide Diamantini, "An E-learning Web 2.0 Experience," *Procedia - Social and Behavioral Sciences*, Vol. 1 (116), 2014, pp. 1217-1221.
- [2] Clyman, John and Better, "Web-App Interfaces with AJAX.; A new approach to building Web applications promises a richer experience for users," *The Independent Guide to IBM Personal Computers*, Vol24, 2005, pp. 76.
- [3] Bo Song and Jie Liu, "Implementation of J2EE data persistence tier with TopLink," *Microelectronics & Computer*, Vol.23(8), Oct. 2006, pp. 132-135.
- [4] Bo Song and Jing Zhao, "Research on network teaching system based on open source framework," *IEEE Ninth International Conference on Hybrid Intelligent Systems (HIS 2009)*, IEEE Press, Vol.1, Aug. 2009, pp. 28-32.
- [5] R. Hanson and A. Tacy, "GWT in Action, Manning Publications Co," USA, pp. 361-370, (2007).
- [6] Bo Song and Miaoyan Li, "An e-Learning system based on GWT and Berkeley DB," *Lecture Notes in Computer Science*, 7332, Vol. 2, Mar. 2012, pp. 26-32.
- [7] Praveen Gupta and M.C. Govil, "Spring Web MVC Framework for rapid open source J2EE application development: a case study," *International Journal of Engineering Science and Technology*, Vol. 2, 2010, pp. 1684-1689.
- [8] Rohan V. Thakare, Santosh Kakade and Bhushan Sapre, "Spring MVC Framework for Web 2.0," *International Journal of Engineering Innovations and Research*, Vol. 1, 2012, pp. 188-193.
- [9] Jin Hua Zhou, "Persistence Design on English Test Questions Library System Based on Hibernate," *Applied Mechanics and Materials*, Vol. 1(543-547), 2014, pp. 4573-4576.
- [10] Bo Song and Yue Zhang, "Implementation on network teaching system based on Java EE architecture," *IEEE Second International Conference on Information Technology and Computer Science (ITCS2010)*, Vol. 1, July. 2010, pp. 354-357.