Web Development Based on Struts and Hibernate Framework

Wang Huilin

Abstract. In order to achieve the efficiency management, a student management system based on Struts and Hibernate framework is designed. The paper analyzes technical features of opensource framework Struts and Hibernate which are based on J2EE technology firstly, then proposes the system construction and analysis the main realization method of the system, finally introduces the key technologies. Through practice using, this system has been proven with the features of high efficiency, convenience, better extendibility and maintainablity.

Keywords: MVC mode, Student management system, Struts and Hibernate, EJB.

1 Introduction

With the development of system structure, software structure has made rapid progresses in many fields. As the key technologies of multilayer application structure, Java and .NET have different characteristics and can be adapted to the requirement of different system. Object/relation image is a relation data model which relation data model is represented by object data model. Hibernate is a image tool of object and relation based on Java. Struts is a program structure used to construct Java Web. AJAX stands for the asynchronous of JavaScript and XML, it is not a new technology, but many integration of mature technology [1]. In the paper, we study the applications of Hibernate, Struts and AJAX in Web development, and give an example of student management system to analysis the realization of the above technologies.

The remaining sections are organized as follows: In section 2, we study the implement of persistence layer. In section 3, we study the implement of business logic layer. In section 4, we study the implement of presentation layer. Section 5 concludes the paper.

Wang Huilin
Department of Mathematics and Information Technology, HanShan Normal University, Chaozhou, China, 521041
2 The Implement of Persistence Layer

The persistence layer is an architectural layer whose job is to provide an abstract interface to information storage mechanism [2]. An API to such an interface should be abstract, and independent of storage technologies and vendors. It would typically have features such as the following:

- **store / retrieve of whole object networks by key**;
- **logical database cursor abstraction for accessing some / all instances of a given type**;
- **transaction support - open, commit, abort**;
- **session management**;
- **some basic querying support, e.g. how many instance of a given type exist etc**;

Usually such layers are built from at least two internal layers of software: the first being the abstract interface, the second being a set of bindings, one for each target database. In practice, there may be three layers since there may be an internal division between the logic for object and relational (and other) storage mechanisms.

Persistence layer is realized through the framework of Hibernate. The step is list as follows:

- **Firstly, design the database according to the system**;
- **Second, using Hibernate development kits generate configuration file by the plug-ins of eclipse plug-ins**;
- **Finally, complete factory class of Hibernate, Javabean and the corresponding map files**.

In persistence layer, using the class of SessionFactory can create a object of session. Session is the fundament of persistence layer, it is equivalent to the connection of JDBC.

3 The Implement of Business Logic Layer

A business logic layer (BLL), also known as the domain layer, is a software engineering practice of compartmentalizing. The business logic layer is usually one of the tiers in a multitier architecture [3]. It separates the business logic from other modules, such as the data access layer and user interface. By doing this, the business logic of an application can often withstand modifications or replacements of other tiers. For example, in an application with a properly separated business logic layer and data access layer, the data access layer could be rewritten to retrieve data from a different database, without affecting any of the business logic [4]. This practice allows software application development to be more effectively split into teams, with each team working on a different tier simultaneously.

Within a BLL objects can further be partitioned into business processes (business activities) and business entities [5]. Business process objects typically