International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 5, Issue 3, May-June 2016 **ISSN 2278-6856**

Development of E-Institute Management System Based on Integrated SSH Framework

Lakshmi R

Assistant Professor, Department of Information Science and Engineering,
Global Academy of Technology, Bangalore, India

ABSTRACT

The J2EE platform is a multi-tiered framework that provides system level services to facilitate application development. The main drawback of classical J2EE architecture is its poor performance on business application due to its heavy weight EJBs. Combining the three popular open sources – Struts, Spring, and Hibernate (SSH), provides an alternative lightweight approaches for building J2EE enterprise based web application. This paper analyses the merits of integrated SSH framework and describes its specific use in E- Institute Management (E-IM) as an effective web application framework. By using SSH framework, not only the efficiency of system development increases greatly and high usage of code generation tools and application layout template reduce the workload of coding, but also it provides a convenient platform for higher software reusability, reliability and maintainability.

Keywords - Framework; Struts; Spring; Hibernate; E-IM

1. INTRODUCTION

In recent years, the web based applications have received greater attention and have become more pervasive. Consequently, the complexity of developing a web application has grown rapidly and J2EE is regarded the most popular platform for building such application systems. To ensure the reuse and efficiency of development process, adopting J2EE technologies to build the web application must select a system framework which has a good performance and being examined. SSH is expected to be a good framework for solving such problems. It uses layered structure and provides a good integrated framework for web applications at all levels in minimizing the interlayer coupling and increasing the efficiency of development. Programmers in the course of practice achieved some successful development pattern which proved practical, such as MVC and ORM, etc. These separate the presentation layer, business layer and data persistent layer thus ensuring reusability and maintainability. The E- Institute Management supplies more advanced techniques in daily transaction management, administrative management and appreciation service. In this paper, web application framework based on SSH is discussed and applied to develop E-IM system. The basic theory of SSH framework is discussed in next section.

2. STRUTS SPRING HIBERNATE FRAMEWORK

2.1 Struts framework

The most popular design pattern usually applied is Model-View-Controller (MVC) which divides the system into three components each responsible for specific task. The model contains the application data and manages the core functionality. The visual display of the model and feedback to the users are managed by the view. Controller interprets input from the user and controls the model and view to change appropriately. Struts [1], which is a very useful technology to construct web application by Java servlet and JSP, provides an open source code framework based on MVC architecture. Components of Struts are as shown in Fig 1.

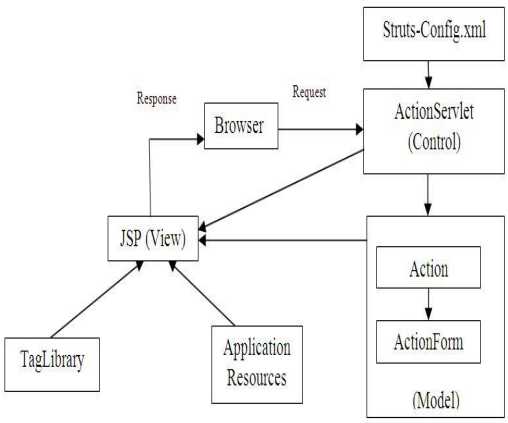

Fig 1 Struts Framework

The business logic of the system is done in Model layer and it contains Action and Action Form objects. Action object encapsulates the processing logic and business calling logic modules and submits the request to the appropriate view component to generate response. ActionForm component describes the data of client form by defining properties. The model level is composed of JavaBean and EJB components. View layer is responsible for the user interaction and consists of JSP files. Struts provide a custom tag library which will interact with the model level and achieve the mapping of ActionForm. The controller controls the action execution that receives

request and redirects this request to the appropriate module controller. It consists of Action servlet and Action Mapping object. The module controller processes the request and returns results to the central controller using a JavaBean object (Action). The central controller redirects the returned JavaBean object to the main view that displays its information.

Struts focuses on reducing the coupling between performance and logic, while it takes less attention to the relevance between business layer and persistent layer. Contrast to the strong interface function, Struts is inferior in business logic function which can be made up by Spring.

2.2 Spring Framework

Spring [3] is a lightweight J2EE application development framework which can organize middle level layer objects efficiently and integrates other View frames seamlessly. It allows choosing its certain module in need and provides unifies interface for different data access technology. Spring is based on model of Inversion of Control (IoC) to separate the actual application from the Configuration and dependent regulations on the application. Adopting IoC also makes it easy for assembling Beans. It also provides compact Aspect Oriented Programming (AOP) and realizes business management based on it to test easily. Spring is not attempting to replace the entire existing framework. Developers are free to choose Spring framework for some or all, since Spring modules are not totally dependant. However Spring also has many disadvantages such as many codes in JSP, excessive flexibility of controller and lack of common controller which can be made up by Struts. As a major business level detail, Spring employs the idea of delay injection to assemble code for the sake of improving the scalability and flexibility of built systems. Thus, the systems achieve a centralized business processing and reduction of code reuse thought the Spring AOP module. Spring integration framework with Struts and Hibernate is as shown in Fig 2.

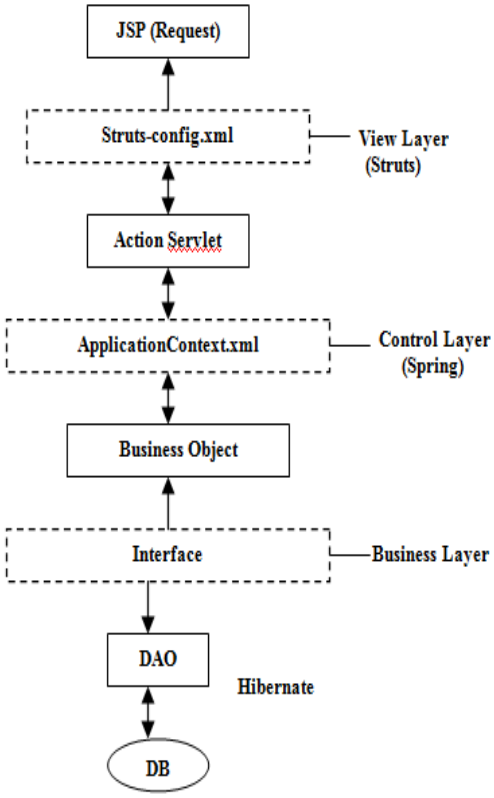

Fig 2 Spring Integration Framework

2.3 Hibernate Framework

In the Web system development, the traditional approach directly interacts with the database by JDBC. This method however is not only heavy workload but also complex SQL codes of JDBC resulted in inconvenient development and maintenance. Considering the large difference between the object-oriented relation of Java and the structure of relational database, it is necessary to introduce a direct mapping mechanism between the object and databases and mapping files will need modifying rather than Java source codes when the business logic changes in the future. Hibernate [4] is a data persistent framework, and the core technology is the Object/Relational database Mapping (ORM). It is generally considered as a bridge between Java applications and the relational database which provides durable data services for applications and allows developers to use an object-oriented query language to the management and manipulation of relational database. It also furnishes object-oriented Hibernate Query Language (HQL), so that the operations are conveniently performed in the manner of a class of objects, examples and attributes.

Hibernate is essentially a middleware providing database services being responsible for the mapping between major categories of Java and relational database. It supplies durable data services and several profiles, such as hibernate properties and XML mapping etc. Its structure is shown in Fig 3.

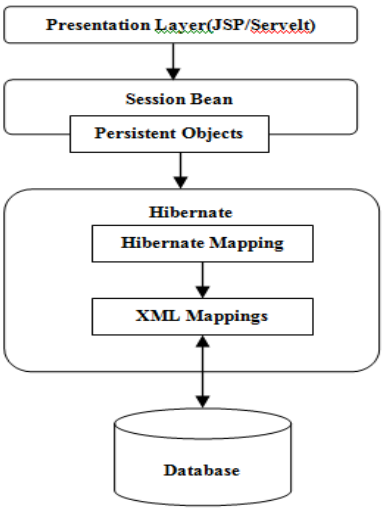

Fig 3 Hibernate Framework

2.4 Structure of Integrated SSH framework

The SSH integration architecture adopts multi-tiered model and provides a relevant strategy for each tier support to isolate the dependence of the inter-tier in order to simplify the system expansion. The developed integrated framework in this paper provides the appropriate integration strategy for web application at all levels by combining the characteristics of these three frameworks and layered idea of web applications, which reduces the coupling degree in each layer of the system and has the system easier for maintenance and extension.
For presentation layer (UI Layer), Struts is the main frame. Struts intercepts the user requests though ActionServlet, and then calls the model to process the requests appropriately, finally the results to the users are shown through JSP files. The system is conveniently maintained under such mechanism by separating the data processing and display. Struts also serves as the frame for control layer to call business logic and other upper treatments.

Spring framework is employed to realize the detail of each level in Business Layer. By using interface-oriented programming, Spring factory model provides Bean generation and management, which avoids the coupling problem between Beans in the form of hard-coded approach. Spring adopts the model of IoC to manage between layers and therefore the implementation of the object becomes transparent, the coupling between the components business is reduced. For the realization of the seamless connectivity at all levels of web application, Spring supports the management on plain old Java Objects (POJO), and it also controls the implementations of the procedures from the presentation layer to the persistent layer and provides interlayer interaction with interface.

Hibernate frame controls the persistent layer. The database objects are mapped into corresponding persistent classes though the ORM and these are useful for middle layer[4, 5]. Data access layer mainly adopts DAO module which isolates the fundamental database operations and upper layer business logic. Hibernate provides ORM which realizes the mapping from object-oriented domain module to traditional relationship database. It takes charge of mapping and provides data query and retrieval functions. This method greatly reduces the development of persistent data programming tasks by translating the Java type HQL statement into SQL statement, and then completing the data operation with JDBC driver.

3 SRUCTURE OF SSH FRAMEWORK FOR E-IM

E-Institute management offers various services for special customers like colleges, institutes etc, through network. As a large scale web application project, the E-IM system is made up of different modules namely faculty, student, parents, and staff. Due to the complex module relations, this paper concerns only specific module like Faculty and Student management to describe the web application of SSH on E-IM system.

Figure 4 shows the structure of proposed E-IM system. Here, JSP is acting as the presentation layer to collect user requests data and business data, web layer utilizes MVC framework to have control over business layer and presentation layer interactions. Business layer is mainly mastered by business logic components with Spring IoC container. Data Access layer is composed of various DAO components, which controls the interaction between database access object and persistent data object. Hibernate persistent layer consists of many PO files, and it achieves the database access under hibernate session management. Spring framework is the core of the system, and MVC control, Service-Level components and DAO components are generated by Spring IoC container, which manage the instances of components.

The modules in the presented E-IM system include faculty basic information, student basic information, marks details, subject details and others. From the perspective of web-level, presentation layer just provides the function to display the data, the middle layer is responsible for dealing with user's request and updating data in presentation layer, and the database layer stores the information. The middle layer is described as follows.

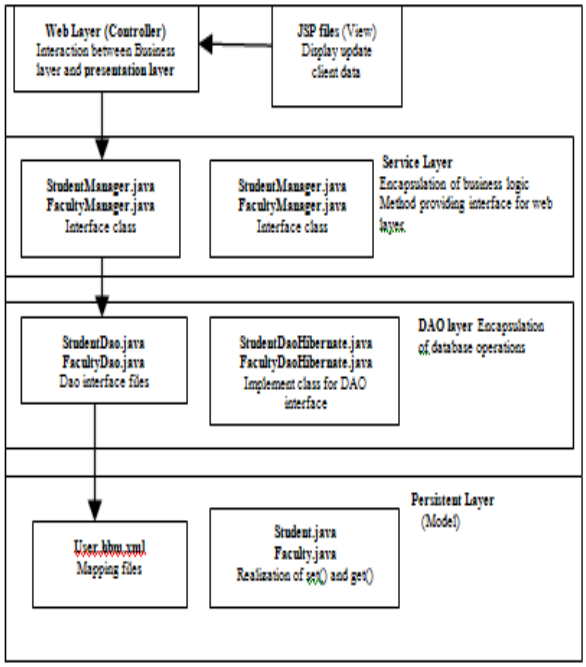

Fig 4 Structure of E-IM System

3.1 Hibernate Layer

The database is designed to contain basic tables to store the information such as Student, Faculty, and so on. Information is stored by PO objects in the Hibernate layer, while the correlative relationships between these objects are embodied in *.hbm.xml

Fig 5 lists the main code in User.hbm.xml as Faculty and Student classes are inherited from User class and they are indicated as joined sub class. The primary job of hibernate layer is to set up PO objects and corresponding mapping files, but the database options are managed by DAO layer.

```
.....
<hibernate-mapping>
 <class
   name="com.eim.model.User"
     table="APP_USER"
 >
 <id
   name="id"
                 column="ID"
  unsaved-value="null"
 >
 <generator class="native">
 </generator> </id>
```

```
.....
<beans>
<!-- FacultyDao: Hibernate implementation -->
<bean id="facultyDao"
class="com.eim.dao.hibernate.FacultyDaoHibernate">
<property name="sessionFactory"
ref="sessionFactory"/>
  </bean>

  <!-- StudentDao: Hibernate implementation -->
  <bean id="studentDao"
class="com.eim.dao.hibernate.StudentDaoHibernate">
    <property name="sessionFactory"
ref="sessionFactory"/>
  </bean>
```

```
.....
<property
       name="cityTown"
       column="CITY_TOWN"
       length="20"
      not-null="false"></property>

<property
       name="lastLoginDate"
       column="LAST_LOGIN_DATE"></property>
.....
<set
 name="roles" table="user_role"
 lazy="false" cascade="save-update">

<key column="user_id"></key>

<many-to-many
  class="com.eim.model.Role"
  column="role_id" outer-join="auto" />

</set>
.....
<joined-subclass
    name="com.eim.model.Faculty"
      table="FACULTY">
<key column="ID" />
.....
</class>
</hibernate-mapping>
```

Fig 5 Main Code in User.hbm.xml

3.2 DAO Layer

Each DAO component is utilized to realize the encapsulation of database access in DAO layer.the implementation of the DAO class relies on the DAO support from Spring, which uses interface defined above. The StudentDao interface and FacultyDao interface is shown in Fig 6. In order to run the program, the

corresponding configuration is required after depicting DAO classes. Figure 7 shows the part of configuration code. Spring container takes charge of generating and managing components of DAO, and the injection of necessary SessionFactory to DAO components.

```
public interface StudentDao extends Dao
{
  public void saveStudent(Student student) throws Exception;

  public void saveParent(User user) throws Exception;

public Student getStudentByEmailId(String email);

  public List<Student> searchStudentFromName(String firstName, String lastName, Long
centerId) throws Exception;

  public List<Student> getStudentsByBatch(Long batchId);

  public List<Student> getStudentByParent(String parentEmail);

  public List<Student> getStudentByCenter(Long centerId);

  public List<Student> getStudentByMentor(Long mentorId);

}
```

Fig 6 StudentDao and FacultyDao interface definition

```
public interface FacultyDao extends Dao
{
   public void saveFaculty(Faculty faculty)throws Exception;

   public Faculty getFacultyByEmailId(String email);

   public Faculty getFaculty(Long id);

   public void removeFaculty(Long id);

   public List<Faculty> getFacultyByCenter(Long centerId);
}
```

Fig 7 Part of the DAO code

C. Service Layer

To achieve the separation between business layer and persistent layer, DAO components are encapsulated by business logic layer components in business layer, and then it is capable of DAO interface-oriented origramming. Business logic layer is realized by IoC framework. In the programming, only the dependent business logic components are needed to be declared, and getters and setters are written in terms of dependency injection pattern, while action instantiation is done in configuration files. Figure 8 displays some of the codes in ManagerImpl

```
public class StudentManagerImpl extends BaseManager
implements StudentManager
{
        private StudentDao studentDao;
        private CenterDao centerDao;
        private RoleDao roleDao;

          ..............

            public void setStudentDao(StudentDao dao)
        {
                this.studentDao = dao;
        }

        public void setCenterDao(CenterDao centerDao)
        {
                this.centerDao = centerDao;
        }

        public void setRoleDao(RoleDao roleDao)
        {
                this.roleDao = roleDao;
        }

          ..............
```

```
public class FacultyManagerImpl extends BaseManager
implements FacultyManager
{
        private FacultyDao facultyDao;
        private RoleDao roleDao;
            private SubjectDao subjectDao;
        private MentorDao mentorDao;

          ..........

            void setMentorDao(MentorDao mentorDao)
        {
                this.mentorDao = mentorDao;
        }

        public void setFacultyDao(FacultyDao dao)
        {
                this.facultyDao = dao;
        }
           public void setRoleDao(RoleDao roleDao)
        {
                this.roleDao = roleDao;
        }

        public void setSubjectDao(SubjectDao subjectDao)
        {
                this.subjectDao = subjectDao;
        }

          ..............
```

Fig 8 Some of the source code in Student and Faculty ManagerImpl.

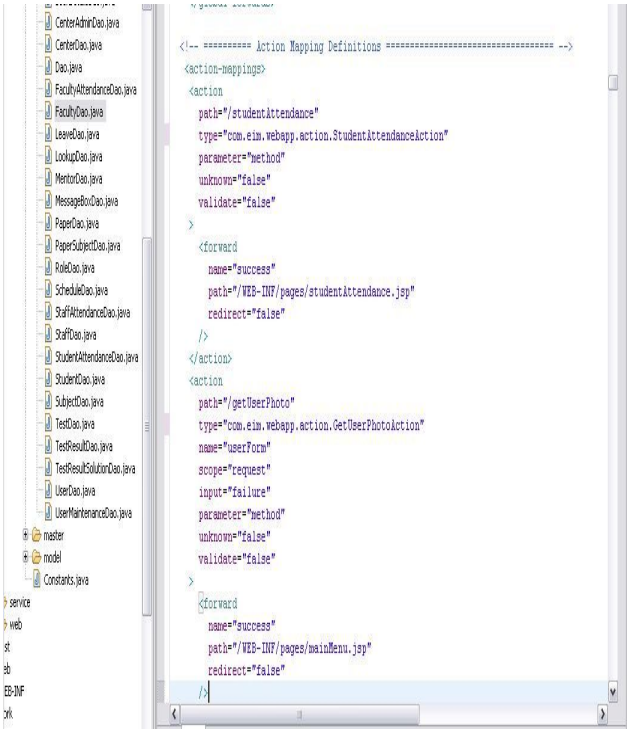

Fig 9 struts-config.xml file content

3.3 Web Layer

Struts+Spring integration method are used in presentation layer. Action in Struts is required to configure in Struts-Confug.xml, and tha actual configuration is set in Spreing container. Whnen client requests are help up by ActionServlet, they are allocated to the corresponding Action in Struts-Config.xml, and then to the bean in Spring container. The code added in configuration file Struts-Config.xml to obtain the Student attendance information by implementing functions is shown in Fig 9.

4 CONCLUSION

Struts, Spring, and Hibernate, each frame has its own advantages and disadvantages. Integrating the distinct characteristics of each frame resulted in effective development and maintenance of web application by using perfectibility of Struts in presentation layer, the flexibility of spring in business management which provides seamless connectivity between frames, the maturity of sophisticated Hibernate in persistent layer. By analysing the merits of integrated SSH frames, certain modules of E-Institute Management system is discussed in this paper.

REFERENCES

[1] C. Cavaness, "Programming Jakarta Struts," USA:O'Reilly Publishing, 2002.
[2] J. C. Maxwell, "A treatise on electricity and magnetism," 3rd ed., vol.2, Oxford: Clarendon, 1892.
[3] M. Raible, "Spring Live," Colorado: SourceBeat Pbulishing, online book, 2004.
[4] Hibernate: Hibernate Reference Manual, version 2.1.6, 2004.
[5] Bing Xue, Zuo-liang cao. Design pattern and in data lasing level frame in web system application. Tianjin Technical institute journal, vol.20, no.1. 2004.