

J2EE and Its Application Framework for Enterprise Solutions

Mohammad Jahid Iqbal, Chandan Kumar Karmakar

Dept. of Computer Science & Engineering, SUST, Sylhet-3114, Bangladesh
E-mail: ove_cse@yahoo.com, ckkarmakar-cse@sust.edu

Abstracts

J2EE is the most generic and powerful platform for developing solutions of information systems around the ICT globe. It has opened the opportunity of bringing all the resources in one platform where the users need not to be so in reality. In this paper we have discussed about the present frameworks used by industry, which is being used along with J2EE to provide solutions. Here the underlying technology of those frameworks and the compatibility with J2EE has been shown. Moreover the criteria for usage and selection of a specific framework at any stage of development suggested.

Keywords

Servlet, EJB (Enterprise Java Bean), Sturts, Spring, Hibernate.

1. Introduction

J2EE is an on-going standard for producing secure, scalable, and platform independent enterprise applications [1]. J2EE standard defines which services should be provided by servers that support J2EE. These servers will provide J2EE containers in which J2EE components will run. The containers will provide a defined set of services to the components. The J2EE specification provides a definition from which enterprise vendors can produce J2EE application servers on which J2EE-compliant applications can be deployed.

We use J2EE for developing web application or server side application. The features of J2EE are as follows:

- a. Portability
- b. J2EE is the best answer to all the problems of enterprise development.
- c. Performance and Scalability.
- d. Eliminates low-level problems.

Though J2EE is a complete solution for enterprise software development, considering different types of development, efficiency, development time, transactions, code organization etc J2EE Framework is established.

2. Component of J2EE

Technologies underlying J2EE are J2EE component. These components are JSP, Servlet, EJB and JSF. J2EE server provides containers that holds and execute these components.

2.1 Java Servlet

We can directly use a java class as servlet if we extends it to Http servlet. Most java programmers will like to write java classes and then use it as a web component. While writing servlets programmers are isolated from designs and they feel comfort to do so. When we need to output bytes instead of text or html we use servlet. Servlets can generate image according to byte output. These servlets are accessed by way of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers.

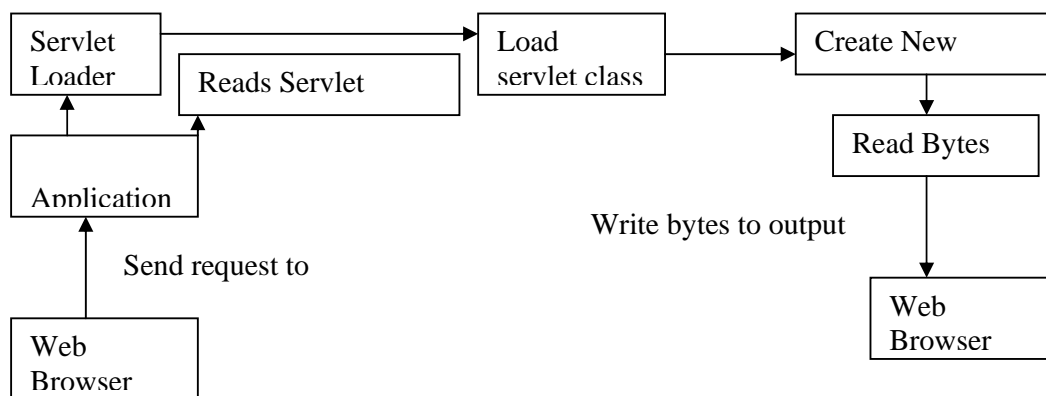


Fig 1 : Working procedure of servlet

2.2 Java Server Pages

JavaServer Pages™ (JSP™) technology lets us put snippets of servlet code directly into a text-based document. A JSP page is a text-based document that contains two types of text: static data (which can be expressed in any text-based format such as HTML, WML, and XML) and JSP elements, which

determine how the page constructs dynamic content. At JSP we can simultaneously write html and java code by using jsp indicator symbol. Designers feel comfort to work with JSP as using JSP's include tag we can directly include any page that holds designs. Another advantage of JSP is, like servlet we need not to compile JSP. JSP is compiled at runtime.

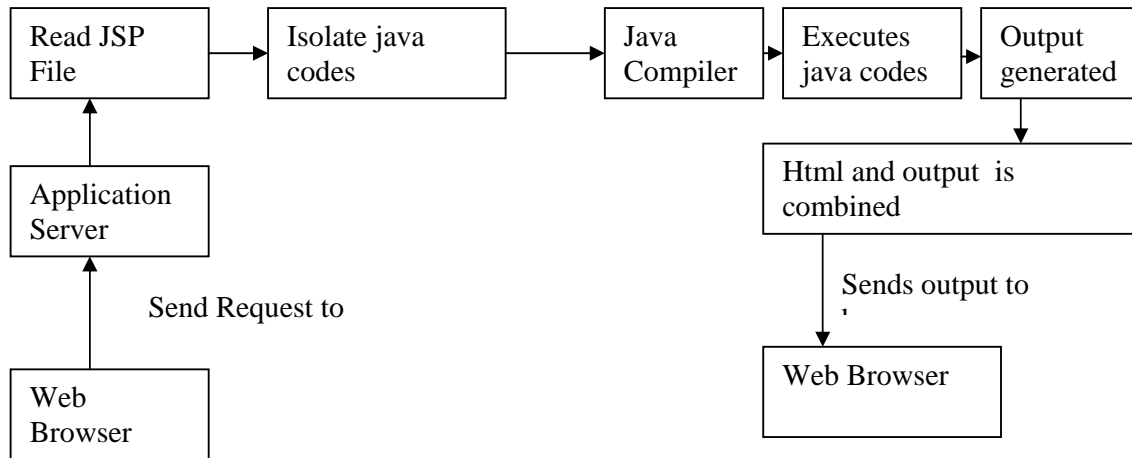


Fig 2 : Working procedure of JSP

2.3 Enterprise JavaBeans (EJB)

An enterprise bean is a server-side component that encapsulates the business logic of an application. .EJB container provides system-level services to enterprise beans [2]. Because the beans--and not the clients--contain the application's business logic, the client developer can focus on the presentation of the client. As a result clients can be thinner and can run on small devices. Application assembler can build new applications from existing beans [3]. To accommodate a growing number of users, we may need to distribute an application's components across multiple machines. In this situation we can choose EJB. We can also choose EJB when we need a Transaction that must ensure data integrity [4]. The application developed by EJB may have a variety of clients. With only a few lines of code, remote clients can easily locate enterprise beans. These clients can be thin, various, and numerous. There are two kinds of enterprise beans: session beans and message-driven beans. A *session bean* represents a transient conversation with a client. When the client finishes executing, the session

bean and its data are gone. A *message-driven bean* combines features of a session bean and a message listener, allowing a business component to receive messages asynchronously [5].

EJB contains Enterprise bean class, Business Interfaces and Helper classes. Enterprise bean class implements the methods defined in the business interface and any life cycle callback methods. The business interface defines the methods implemented by the enterprise bean class. Classes needed by the enterprise bean class, such as exception and utility class are Helper classes.

2.4 JavaServer Faces Technology

JavaServer Faces technology is a user interface framework for building web applications. The main components of JavaServer Faces technology are as follows:

- A GUI component framework.
- A flexible model for rendering components in different kinds of HTML or different markup languages and technologies.
- A standard RenderKit[6] for generating HTML/4.01 markup.

The following features support the GUI components:

- Input validation
- Event handling
- Data conversion between model objects and components
- Managed model object [7] creation
- Page navigation configuration

3. J2EE Application Framework

J2EE Application Frameworks are the supporting technologies that maintain J2EE specification and work with J2EE. These are Struts, Tiles, Spring, Hibernate etc.

3.1 Struts

Struts Framework is the implementation of Model-View-Controller (MVC) design pattern for the JSP. Struts can be used at large development as well as small development. We use JavaBeans at JSP.

JavaBeans acts as an intermediary between JSP and database. Struts is a framework that implements that concept and combines classes and Servlets with JSP[8]. Struts contains the following:

- a. The Struts ActionServlet — the main controller component. It is in charge of deciding which Action class will process the web request and which JSP will generate the web response.
- b. Action classes — the user-written controller components. They are in charge of processing the web request with the help of other Java classes (or Java Beans). Action classes must be a subclass of Struts Action class.
- c. Java classes — the model components. They hold the application data.
- d. JSP pages — the view components in charge of generating the web response using the data stored in the Java classes.
- e. Configuration files:
 - o web.xml — an XML file with the configuration for a J2EE web application; and
 - o struts-config.xml — an XML file with the configuration for Apache Struts ActionServlet.

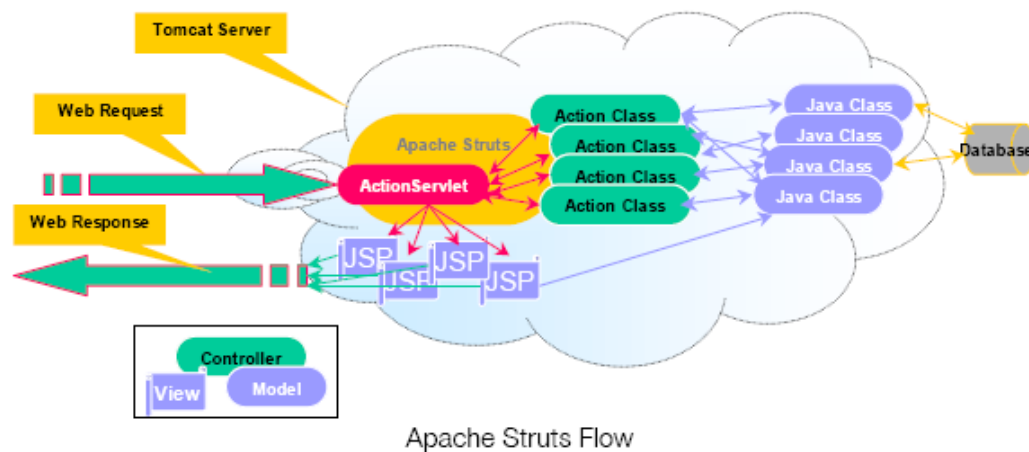


Fig 3 :The working process of Struts

Figure 3 shows the working process of struts [9]. Working process of struts can be described as:

- a. Struts ActionServlet captures a client's web request.
- b. The web request is passed to an Action class.
- c. In the Action class the request is processed with the help of other Java classes.

- d. The flow is sent back to the Struts ActionServlet. It then decides to which JSP page it will forward the processed web request.
- e. Finally, the JSP with the help of other Java classes generates the web response. The web response is sent to the client.

The Action classes implement the controller role of the MVC pattern, the Java classes implement the model role (data holder classes) and the JSPs implement the view role.

3.2 Tiles Framework

To explore templating and layout solutions, we use the Tiles framework. The Tiles framework's view components are known as tiles. Tiles efficiently organizes Html and JSP view components. The framework uses an XML configuration file to organize those tiles. This framework not only enables you to reuse tiles, but also organizes the layouts. Tiles make an excellent enhancement when it is combined with struts [10]. We make decision in using jsp include directive or tiles or struts and tiles, considering the criterias: Page Number, Code repetition, Layout Control, Coupling and Complexity.

3.3 Spring Application Framework

Spring is a light-weight framework for the development of enterprise-ready applications. Spring can be used to configure declarative transaction management, remote access to your logic using RMI or web services, mailing facilities and various options in persisting your data to a database. Spring framework can be used in modular fashion; it allows using in parts and leaving the other components which are not required by the application. The main aim of Spring is to simplify the J2EE development and testing.

Spring's goal is to be an entire Application Framework. Other popular frameworks like Struts, Tapestry, JSF etc., are very good web tier frameworks but when we use these framework, we have to provide additional framework to deal with enterprise tier that integrates well with these framework. Spring tries to alleviate this problem by providing a comprehensive framework. Spring technology has features like Transaction Management, JDBC exception Handling, Integration with Hibernate , JDO, IBATIS, AOP Framework, MVC Framework.

Spring is well-organized architecture consisting of seven modules. Fig 4 represents the Spring Framework Architecture [11].

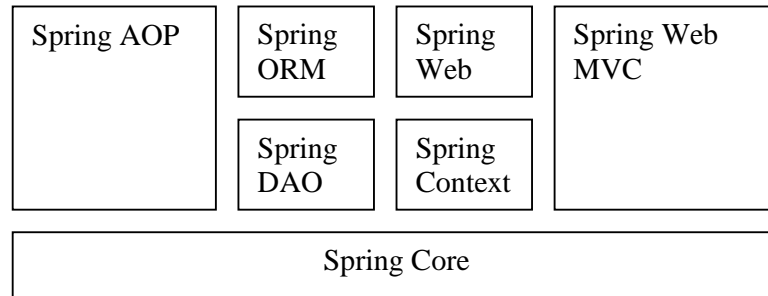


Fig 4: Spring Framework Architecture.

Spring attacks EJB as unduly complicated and not susceptible to unit-testing. Instead of EJB, Spring suggests that we make use of ordinary Java beans, with some slight modifications, to get all the supposed advantages of EJB environment. Thus, We can think Spring as a replacement of EJB .But a quote from a book on Spring, “EJB is complex”, as the author put it nicely, “not for just being complex. It is complex because it attempts to provide solutions for complex problems” [12]. Spring testing is easier. As spring is based on JavaBeans naming convention, programmer finds it easy to work with. Its make use of AOP (Aspect Oriented Programming) [13].

3.4 Hibernate

Hibernate 3.0, the latest Open Source persistence technology at the heart of J2EE EJB 3.0. Hibernate maps the Java classes to the database tables. It also provides the data query and retrieval facilities that significantly reduce the development time. Hibernate is not the best solutions for data centric applications that only uses the stored-procedures to implement the business logic in database. It is most useful with object-oriented domain modes and business logic in the Java-based middle-tier. Hibernate allows transparent persistence that enables the applications to switch any database [14].

Hibernate can be used in Java Swing applications, Java Servlet-based applications, or J2EE applications using EJB session beans.

Hibernate 3.0 provides three full-featured query facilities: Hibernate Query Language, the newly enhanced Hibernate Criteria Query API and enhanced support for queries expressed in the native SQL dialect of the database. Filters for working with temporal

(historical), regional or permitted data Runtime performance monitoring: via JMX or local Java API, including a second-level cache browser. Hibernate is Scalable: Hibernate is very efficient and due to its dual-layer architecture can be used in the clustered environments.

It reduces the development timings as it supports inheritance, polymorphism, composition and the Java Collection Framework. Hibernate XML binding enables data to be represented as XML and POJOs interchangeably.

The following Architecture describes the high level architecture of Hibernate.

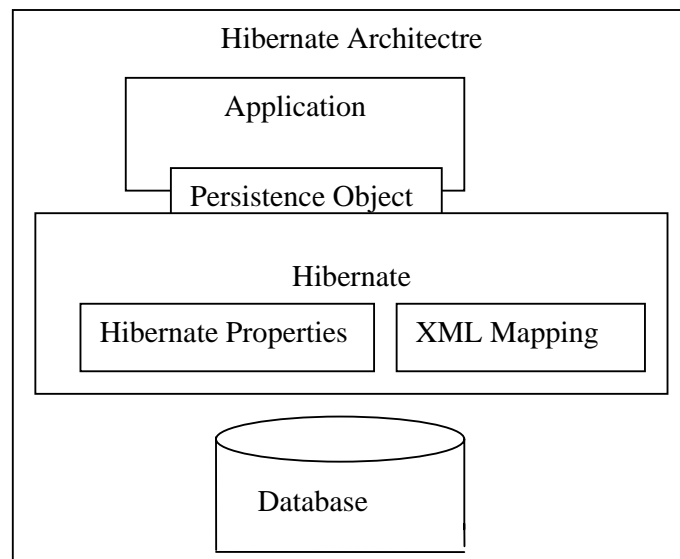


Fig 5: Architecture of Hibernate

4. Conclusion

J2EE is the mostly used platform for enterprise software development. But the technology behind this is generic and simpler like the natural world. So for working with the framework we need to conceptualize the technology behind it and the usage. Because according to OOPs principle

application is only the usage and integration of objects. In this paper we have shown the relationship among framework and the underlying technology and the usage as well as.

References

- [1] Teach Yourself J2EE in 21 Days - Sams (eBook-pdf).pdf
<http://www.sampublishing.com/title/0672325586>
- [2] MasteringEJB3rdEd http://www.theserverside.com/news/thread.tss?thread_id=31942
- [3] J2EE Tutorial <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/>
- [4] Data integrity <http://www.utexas.edu/its/windows/database/datamodeling/rm/rm5.html>
- [5] Java2EnterpriseEditionBible.pdf <http://www.amazon.com/exec/obidos/tg/detail/>
- [6] http://java.sun.com/javaee/javaserverfaces/1.1_01/docs/api/javax/faces/render/RenderKit.html
- [7] Model Object <http://www.javapractices.com/Topic187.cjp>
- [8] Mastering Jakarta Struts.pdf <http://14160.tomore.com/367/31974.html>
- [9] Manning-Struts.in.Action-2003 http://www.esnips.com/_t_/Struts
- [10] Integrating struts tiles <http://www.ibm.com/developerworks/java/library/j-integrate/>
- [11] Spring Architecture <http://www.roseindia.net/spring/springarchitecture.shtml>
- [12] Spring in Action - Craig Walls-Manning press
<http://www.amazon.com/exec/obidos/tg/detail/-/1932394354?v=glance>
- [13] AOP http://en.wikipedia.org/wiki/Aspect-oriented_programming
- [14] Hibernate <http://www.systemmobile.com/articles/IntroductionToHibernate.html>