Global Research **Journal of Global Research Computer Science & Technology**
Researcher Forum **(JGRCST)**
Vol-I, Issue-III, August 2014
ISSN: 2349 - 5170

Java Technology in the Design and Implementation of Web Applications

Kavindra Kumar Singh
School of Computer and Systems Sciences
Jaipur National University
Jaipur

Abstract: This paper reviews the development of Web applications in the Java programming language, namely the creation of a Web application through three-layer architecture using Java Servlet technology and Java Server Pages (JSP). This paper shows the development of Web applications in the Java programming language. Web applications are used in order to respond to the user's requests, i.e. generate Web pages on the client's request. The applications are stored on the server and operate on the principle of client-server connection, i.e. client sends a server request, when the server fulfils the request, it sends a response to the client. The advantages that we specified in this paper are the development of web applications applying Java technology. Motivated by the current popularity of this technology, our aim was to introduce a way of developing web applications that will become a trend in the future. Primary goal of this paper is to present the most appropriate methods of developing web applications. Conclusions from our study are that the Java is programming language with exceptional opportunities when it comes to web development in terms of simplicity of implementation and speed of execution of the final product.

Keywords: Internet, Java, Java Servlet, Java Server Pages.

1 Introduction

A web based application is a software package that can be accessed through the web browser. The software and database reside on a central server rather than being installed on the desktop system and is accessed over a network.

A common example is web-based email. ie, Hotmail, Yahoo! mail.

Along with the great demand of web-applications a number of issues related to quality of web-applications have also gained a great attention in web-application development process. Great competitions in the web-application business compel web-application developers more quality conscious. It will be of great value if a web-application could survive against demanding and changing customer requirements, and changing business requirements. Building high quality web-application is really a difficult and challenging task.

As the development platform influences associated development process, methods, tools, and people, it really plays a major role for making development process simple, efficient, and robust and for achieving high quality of web-application. Java EE platform is open, standard based hardware and OS independent platform on which distributed enterprise applications can be developed and run. The Java EE platform based web-applications use Model/View/Controller (MVC) design pattern for three architectural components: presentation logic, controller logic, and entity/business logic.

We discuss MVC design pattern used in traditional GUI based applications and also discuss how it is adapted in the architecture of Java EE platform based web-applications. We analyze the Java EE platform as per (i) quality attributes

requirements of web-applications (ii) related support in development process (iii) related support to involved people during development, and present our results in quick digestible form. Our results indicate that web-application development on Java EE platform has a great value in providing high quality to web-application, satisfying changing customer requirements and satisfying changing business requirements. The presented work, in this paper, concentrates on achieving high quality for web-application developed on Java EE platform from different perspective of overall development process.

2 The Web-Application Architecture on Java EE Platform Using MVC Design Pattern

An MVC design pattern has remained fundamental architectural design pattern even after great evolution in the architectures of user interactive applications. So, we would like to provide short introduction on it to the readers before we discuss its usage in web-application architecture on Java EE platform.

The MVC design pattern is widely used by programmer, software designer, and GUI component developer to architect they are developing. The MVC design pattern, as shown in Figure 1, consists of three kinds of objects: Model, View, and Controller, which handle three basic responsibilities of any: entity (data), boundary (presentation), and control (behavior) respectively. The model encapsulates application data and business logic; the view handles rendering of application data and visual interface to user; and the controller handles user's interaction with the application.

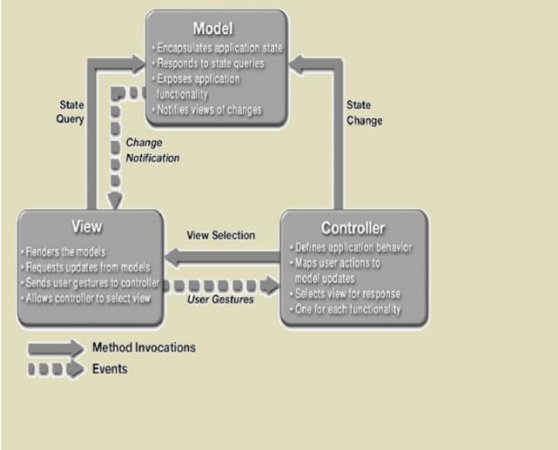

Fig.1- MVC Architecture

The MVC design pattern separates views and models by establishing subscribe/notify protocol between them. A view object must ensure that its appearance reflects the state of the model. The model object is independent of both view and controller objects, so it is possible to have multiple views (presentations) of same model (data). All associated views can subscribe with model and model notifies them about its state change. When user interacts with the MVC design pattern based

GUI form or page, all fired events are captured by controller object. The controller then decides whether the fired event is related to change in state of model or change in state of view. As an example, when the user fires event related to changing value in text field, the controller calls the method of model to change its content. But if the user performs horizontal or vertical scroll-up or scroll-down, the model content does not change and only the view should be notified to reflect the changes in its appearance.

3 Web-Application Development Process on Java EE Platform

In this section, we discuss about web-application development process in Java EE platform and the support available for web-application development from the Java EE platform.

A. Web-Application Development Process:

A typical development process on Java EE platform involves following tasks: designing, coding, creating deployment descriptor, packaging, assembling, and deployment. These tasks are also applicable to web-application development also. The Java EE platform specification specifies the roles and their responsibilities in Java enterprise application development.

Configuring deployment descriptor Development, Integration Deployer Deployment of the assembled application in a specific operational environment.

B. Support in Development Process:

The development becomes efficient if tools/technology support is available during coding, testing, integration, deployment and maintenance phases. Here, we discuss how Java EE platform provides support during all these phases to make development process efficient.

i) Support in Coding: The development environment should provide facilities of avoiding typos, require minimal effort for adding new functionalities, and provide support for standards based technology and tools.
ii) Support in Testing: Good debugging and unit testing support are basic requirement in testing.
iii) Support in Integration and Deployment: The integration and deployment of web-application should be as easy as possible to minimize deployment time and downtime. And the deployment process should be independent of different hosting application server provider.
iv) Support in Maintenance:

The Java EE platform specification clearly specifies the roles and responsibilities of different involved people.

4 Valuable Features for Web Application on Java EE Platform

As Java technology is object oriented and platform independent, many features such as scalability, portability, reusability, security, high performance, flexibility are inherent in Java classes or components. Both Servlet and JavaBeans components are java classes, so above stated features are inherent in them. This also applies to EJB component also, which is collection of java classes and deployment descriptor. JSP scripting language is used to create JSP page. In short, all three components: servlet, JavaBeans/EJB, and JSP that are used in implementing MVC design pattern on Java EE platform are scalable, portable, reusable, secure, high performance, flexible. The following available features on the Java EE platform add value to web-application.

A. Security

The communication security is provided through SSL support. For SSL support, the SSL connector should be configured on the Java EE container and the server certificate signed by Certifying Authority (CA) should be installed on the Java EE container. A Java EE based web-application is configured for confidentiality and integrity declaratively.

B. Transaction Processing

Java EE platform supports container managed transaction and bean managed transaction for session bean and message driven bean (MDB). In Container-managed transaction, the enterprise bean code does not explicitly mark the transaction's boundaries using begin transaction and commit transaction. Instead, the transaction is configured in deployment descriptor. The container begins a transaction immediately before a business method in enterprise starts. It commits the transaction just before the business method in enterprise bean exits. If an exception gets generated during business method execution, the container will automatically roll back the transaction. The support for only either single transaction or no transaction at all is available from container managed transaction.

C. Support for session management:

Most of the web-applications on the Internet handle the session using cookie mechanism. So, the users, who are accessing such web-applications from browsers that do not support cookie mechanism, cannot participate in session and will not be able to access any personal resource. The encode URL mechanism on Java EE platform automatically determines whether the client's browser supports the cookie or not, and then decides how the information about session identification should be stored on client machine. If the client's browser does not support the cookie or the cookie feature has been disabled by the user, the session id value is encoded in URL part of each hyper-link on the page that client is going to use.

D. Customized error-pages:

If a user is trying to access a resource, which is not available, the server shows the error message 404 (SC_NOT_FOUND) resource not found exception. Instead, the error page containing beautiful description in non-technical English language is

shown, the user will really understand the cause of problem and sometimes feels that you care for the users. A good web-application should not show the error messages generated by application server or web-server directly to users. As an example, a web-application can be configured for 404 resource not found exception as shown in. Using this configuration, the error pages for user- defined error messages can also be specified. Instead of error- code, the exception-type can also be specified. The resourceNotFound.html file would contain error message in a language understandable to user.

E. Internationalization/Multi-language Support:

If information is provided to users in a language that they understand and use, it would be easier for users to understand and use the application. The Unicode support for strings is inherent in the Java language.

5 Conclusion

We analyzed the Java EE platform as per quality attributes requirements of web-applications and found that quality attributes: usability, functionality, reliability, efficiency and maintainability can all be satisfied with it. From the discussion it is evident that Java EE platform simplifies the designing, development, deployment, integration and testing process of web-application without compromising high quality. It also empowers the development of web-application with scalability, portability, interoperability, reusability, flexibility and security. Many frameworks for web-application development are available on Java EE platform, but no one is found providing support for high quality distributed web-application. In future, we intend to work on developing distributed high quality web- application framework on Java EE platform.

References

[1] A. Leff and J. T. Rayfield "Web-application development using the model/view/controller design pattern," in Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing, 2001, pp. 118–127.

[2] E. Gamma, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.

[3] F. A. Masoud, Dana. H. Halabi and Deema. H. Halabi, "ASP.NET and JSP frameworks in model view controller implementation," in Proceedings of Information and Communication Technologies, 2006. ICTTA '06, pp. 3593–3598

[4] G. E. Krasner and S. T. Pope, "A Cookbook for Using the Model-View- Controller User-Interface Paradigm in Smalltalk-80," Journal of Object- Oriented Programming, August/September 1988.

[5] Olsina, L. et al., "Specifying quality characteristics and attributes for web sites," in Proc.1st ICSE Workshop on Web Engineering, ACM, Los Angeles, May 1999.