IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Volume-4, Special Issue-3, May 2016 E-ISSN: 2347-2693

Development of Web Applications Using Google Technology

Vaibhavi Nayak [1], Vinuta V Naik [2],Vijaykumar A S[3]

[1] *Bachelor Of Computer Application, KLE'S BCA PC Jabin Science College, Hubli, INDIA*

[2] *Bachelor of Computer Application, KLE'S BCA PC Jabin Science College, Hubli, INDIA*

[3] *Bachelor of Computer Application, KLE'S BCA PC Jabin Science College, Hubli, INDIA*

Available online at: www.ijcseonline.org

Abstract- This paper describes some metrics to consider for web application development with frameworks like Google Web Toolkit and Google App Engine, technologies with a particular mode of operation and with some restrictions that affect the design and the functionality of these applications, but also offer great benefits such as improved user interface, better usability, improved performance, greater scalability and as the ability to use certain services, which allow application interoperability with different systems.

Keywords— Web application, Java, GWT, App Engine, JDO, Design patterns, Model View Presenter.

[1] INTRODUCTION

A WEB 2.0 has been influential in the rise of large technologies aimed at facilitating the development of Web applications and both offer features that until recently, only they found in desktop applications. One such technology is Google Web Toolkit A oriented framework to build RIAs (Rich Internet Applications), which proposes a new paradigm in regarding the development of these applications. Furthermore, when required large capacity Web processing and low-cost housing, appear companies like Google or Amazon that allow the use of their infrastructure for accommodation and Running Applications Web through the use of frameworks such as, App Engine for Google case. Such services while offering great benefits, also they involve some restrictions that limit or condition design application.

In this article are disclosed broad aspects the use of technology Google Web Toolkit and Google App Engine for developing Web applications. First Instead the reader is contextualized about who they are and how these technologies work, then the architecture is defined suggested for these applications by using design patterns that improve and facilitate control this behavior. Also they mentioned some technical to be considered during the design and details application deployment and finally present a summary of the advantages and disadvantages incurred by making using these technologies.

[2] CONCEPTUALIZATION

The following describes the concepts necessary for achieve contextualization of what they are and how they work technologies under study.

[2.1] Google Web Toolkit (GWT)

It is a framework for developing Web applications GUI rich user (RIA) based on Ajax (Asynchronous JavaScript and XML, JavaScript asynchronous XML), through the programming language Java. Since Ajax bases its operation on languages type script, GWT compiles the Java application code to its equivalent JavaScript to use this technique. Is important to note that for thecompilation process, a set of libraries that emulate JavaScript is required the classes included in the Java virtual machine known as JRE (Java Runtime Environment, Runtime Environment Java) emulator, however, since no JavaScript It supports features offered by the Java technology as managing threads, sockets, files, etc., not all Java code can be compiled to JavaScript. In is evaluated as GWT provides APIs easy to create graphical user interfaces so similar to Java AWT or Swing libraries, this provides diversity of graphic components ranging from space managers (layouts) to lists and tables for data sample. Regarding the appearance of each Chart component, you can use style sheets cascade (CSS) in order to separate the behavior of an object and how it is presented, as well as facilitating changes in the appearance of the application without that recompilation.

A feature of GWT is that it is based on Ajax, execution is done on the client via the browser Web, which allows control of the graphical interface application is handled there and not on the server as they do traditional Web applications; this brings the advantage of fact of having to transport the user interface show only the information necessary for its operation, which in turn releases the load on the server and provides greater interactivity and usability in these applications.

An approach to such behavior can be observed in the Fig 1:

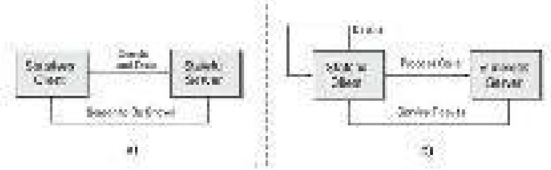

Figure1. (a) Traditional Web Application Vs (b) Ajax-based Web application

Regarding communication between the client and the server, This framework provides mechanisms for the exchange of data regardless of the technology used in the server, using XML, JSON (JavaScript Object Notation) or HTTP; however, if used technologies Java Servlets and JSP like, can be used in addition to earlier, GWT-RPC (Remote Procedure Call) a based on Java Servlets for direct transport mechanism of objects through the network.

[3] **GOOGLE APP ENGINE**

It is a platform and SDK (Software Development Kit) development and Web hosting infrastructure using of Google, designed with the purpose of having great scalability and performance, being able to distribute the load in many servers as necessary in order to provide a quick response. Its main component, the App Engine Data store is based on technology Big Table Google a No SQL database (Not only SQL) with Data is stored in a structure similar to a type Hash table, that is, by the key- value even more advanced. App Engine supports languages Python and Java, for the latter allows store access data through interfaces to access to JPA (Java Persistent API) and JDO (Java Data Objects) with Data Nucleus as implementation in both cases. However, it does not support all the qualities defined in these standards because it has restrictions regarding joins between entities, polymorphic queries, advanced text search, among others.

[3.1] Java Data Objects (JDO)

It is a standard for access to persistent data bases data from Java applications, through the use of objects flat Java (POJOs). Unlike frameworks Similar, JDO is characterized by being oriented to different data sources and can be XML files, databases relational databases, No SQL databases, among others. Additionally, being a standard to make use of this is It requires some of its implementations as may be JDO (Java Data Objects TriActive) or post Data Nucleus that alone is not functional.

[4] ARCHITECTURE DESIGN

The GWT application development involves a separation of the functionality of these between the client and server, which is necessary to know that functional part the application will be executed on the client side and on the server. From the foregoing, it is desirable to structure GWT application such that all handling of the GUI (graphics creation and control components event handling), be performed on the client, while the handling and storage of information a work done on the server. This structure improves application operation since, not transport GUI code for each request, optimizes response times and improves user interaction. Regarding the management of the graphical interface, there are several design patterns that address issues such as interaction between views and independence between the ways the data of the generated shares are shown as response to an event. These design patterns include Model View Controller (MVC) and Model View Presenter (MVP), patterns during the years They have proven effective in controlling GUI behavior, why it worth noting the way they operate and their internal structure. The MVC pattern is formed by the interaction of Observer design patterns, Composite and Strategy. He first one is implemented between the view and model in order to achieve synchronization between them, so that any change in the pattern is notified to the view and perform the appropriate action. Furthermore, the Composite pattern is implemented in the view to the creation complex graphical interfaces from graphics simple. Finally, the Strategy pattern is implemented in the driver as a regulatory mechanism between actions generated and sent to the model view, this in order to separating the way the data is behavior.

As for the MVP design pattern it is known to be a variation of the MVC pattern where eliminating the implementation of the Observer design pattern between sight and model, which means that all control of the interaction among these is handled by the presenter (the equivalent of MVC controller but with more features). Regarding the other patterns that are part of MVC (Composite and Strategy) maintain their purpose and operation MVP. In either case, the patterns mentioned allow appropriate interaction between the graphical interfaces, data model and control the behavior of these. Without but for unit testing to the interface graphical applications in GWT is convenient to use pattern MVP rather than MVC pattern because, thanks to its structure, it is possible to simulate the behavior of objects GUI by use of Mocks, no technique It works great if the MVC pattern used. Returning to the architecture, the server side, is manipulates and stores the information necessary for the application operation. There, through App Engine, you can use Google's infrastructure to perform such tasks. Since App Engine can be used only by programming languages Java and Python (Although support for interpreted languages such as JavaScript and Ruby), this paper referred to his Java version. According explained, the architecture suggested for application development GWT and App Engine-Java.

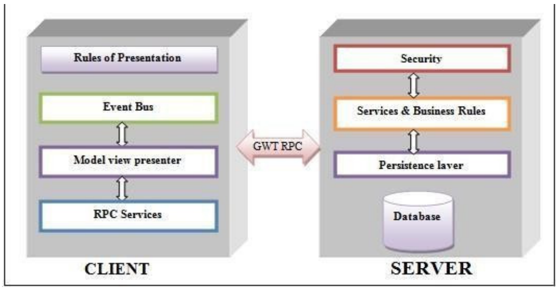

Fig. 2 : Architecture Suggested GWT and App Engine applications

The client controls everything related to the presentation layer, i.e. bus events, presenters, views or graphical interfaces and service definitions for communication with the server. On the server side are the rules business, data access control, security and implementation of services for the exchange of customer information; Note also suggested GWT-RPC used for communication between the client and the server, so this allows the direct transport of objects, which in turn facilitates handling.

[5] IMPLEMENTATION

Based on the architecture proposed in Section III, to some technical details are presented for development of Web applications that make use of GWT and App Engine technologies.

[5.1] Model

As is known, the domain model is an abstraction where entities and relations between them are expressed, involved in the execution of a process. Applications GWT generally has a model to the server, where the business rules and entities persist, and a model for the customer used for transport and data visualization. If POJOs are used, it may be the When using the same model in both the client and the server, as long as all the classes that make up the model can be compiled to JavaScript. However, the fact of using App Engine-Java means consider restrictions on how they saved data when designing the domain model, because the Google Data store does not support some characteristics of object-oriented design. As a solution to the above, the following arise recommendations to consider in the design model domain for an application that makes use of these technologies:

- Use class composition rather than inheritance.
- Prefer the use of property relations without property relations.
- Apply the design pattern DTO (Data Transfer Object) for sending data to the client.
- For many to many relationships they can be used only through relationships without property, but it is the responsibility of the application keeps both sides of the relationship.

[5.2] Persistence with JDO

With regard to the persistence, App Engine-Java allows that through interfaces for accessing data or JPA JDO creating, editing, deleting, and obtaining is controlled Object on Google Data store. For him JDO case, the first step is to perform the mapping of the entities that make up the application domain, task It can be performed by means of entries including Java over the entity or independently based files XML. As the manipulation of objects with JDO you need to know:

- To save an object for the first time, just using the interface function make Persistent Persistence Manager and send the object as a parameter saved.
- To update an object is necessary to obtain the current transaction using the function Persistence Manager current Transaction the interface, begin open by function, to obtain the object update the data store, edit attributes or object properties and finally confirm function changes by the transaction commits.
- To delete an object function can be used delete Persistent the interface Persistence Manager sending as parameters the object to delete.
- For one or more objects of an entity is necessary to use the Query interface, sent by JDOQL query parameter (JDO Query Language. - Object Query Language for JDO) where specifies the entity and the condition of the objects get. As can be seen, the use of JDO facilitates control and manipulation of objects that reside in the data store Google, however, the complexity in use depends on the kinds of relationships used in the model domain and how the mapping is performed.

[6] ADVANTAGES AND DISADVANTAGES

As it has been seen throughout the article, GWT and App Engine frameworks provide some issues when developing Web applications outweigh the benefits they offer. In order to provide the reader the possibility of issuing their own judgment about the use each technology, shown in Table I and Table II some advantages and disadvantages encountered in these.

ADVANTAGES OF FRAMEWORK GWT

- It facilitates creation of complex graphical interfaces, web applications support features like drag and drop.
- Support for JSON and XML allows interoperability with different technologies for the server side.
- Using GWT-RPC provides data exchange with server as long as they use a java technology.
- There are graphical tools facilitating design user interface with this technology.

When compiling an application GWT is obtained as a result Java Scripts which they are executed by any web browsers java script supports without need of plug-in.

DISADVANTAGES OF FRAMEWORK GWT

- Generates too much code HTML to display things simple, this increases the time process for display.
- Building interfaces simple graphical means more effort if languages were used.
- The use of asynchronous calls for communication with the server effects how some data are handled.
- The facts that the results of compile a GWT application in JavaScript, increases the risk of XSS and indexing and difficult tracking the implementation by search engines.

ADVANTAGES AND OF FRAMEWORK APP –ENGINE

- Allows the app to make use of Google infrastructure.
- It facilitates the creation model data from POJOs by means specifications to JPA and JDO.
- It offers a great performance for application execution.
- It provides a mechanism to facilitate the use of services such as me cached, mail, XMPP authentication by accounts of Google, among others.
- It provides tools that facilitate applications rise to Google infrastructure.

DISADVANTAGES OF FRAMEWORK APP –ENGINE

- Restrictions on the type of relations model domain
- Despite using JPA and JDO does not implement all functionality defined in these.
- Restricts the use of certain classes include in the JVM
- Not allow scratch Advantages Text directly. For it is needed some adoptions to the model applying techniques for FTS (Full Text Search) by through the use of libraries as Lucerne.

CONCLUSION

Using design patterns, in this case the Building graphical interface, help in obtaining quality software, extensible and reusable, while the solve the problems for which they were designed. Without however, it increases the complexity during the development of software and sometimes requires a large number of processes to perform simple tasks, which affects performance of the application. Regarding the use of App Engine-Java in the development of Web applications, it is necessary to consider the fact it is a technology with several restrictions, implies a greater effort in introducing features not offered by the framework and at the same time, conditions the design domain model of an application for persistence. Due to the nature in the way they function GWT and App Engine technology and restrictions have, integration with other frameworks for Java, as e.g. spring, is complex and cannot be reap the full benefits offered by them.

REFERENCES

[1] G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.[4] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.[5] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.Yorozu, M. Hirano, K. Oka, and Y