Evaluation and Refinement
# of Web Application Architecture Framework

1. Introduction:

Internet and its applications have evolved rapidly in the last fifteen years. Internet is capable of providing different kinds of services through dynamic web applications. Web applications have become more interactive and flexible to address different areas related to the needs of the society and business. Enterprise solutions have become too complex in terms of fulfillment of different kinds of customers requirements. Banking, e-commerce, communication (mails, chat), stock trading, social networking are the areas where web applications are providing services to large number of users. In order to provide solution for such complex problems, web applications have become too lengthy, complex, distributed and divided into different layers and components. In such a scenario where different customers are accessing services of web applications from different geographic locations concurrently, performance issues of web applications become very important as they are associated with customer satisfaction. If a web application performs poorly, the customer will switch to another web application. There are many factors which determine performance of a web application, such as system parameters like network protocol, maximum number of connections supported by the web server and maximum number of threads supported by the database management system at server side, resource parameters, like disk seek time, latency and transfer rate, network bandwidth and CPU speed etc. and workload parameters like number of hits per day on the web proxy server, number of requests/sec, number of transactions submitted to database/sec. etc. influence the performance of web applications. (Subraya & Subrahmanya, 2000)

Web application design is based on an architectural framework design so as to improve functioning of the web application and perform better partitioning of different components or parts of application according to the architecture of that framework. Effective architecture is necessary for structured development and increased maintainability. There are different views of architecture which can be relevant to the understanding of web application structure (Ziemer, 2002).

- Physical view :It provides physical abstraction of client – server architecture; 2 tier , 3 tier , N tier architecture
- Logical view: It provides high level of abstraction of different parts of application.

The Framework design pattern represents logical view of application. Figure 1 illustrates 2 tier and 3 tier logical view of application.

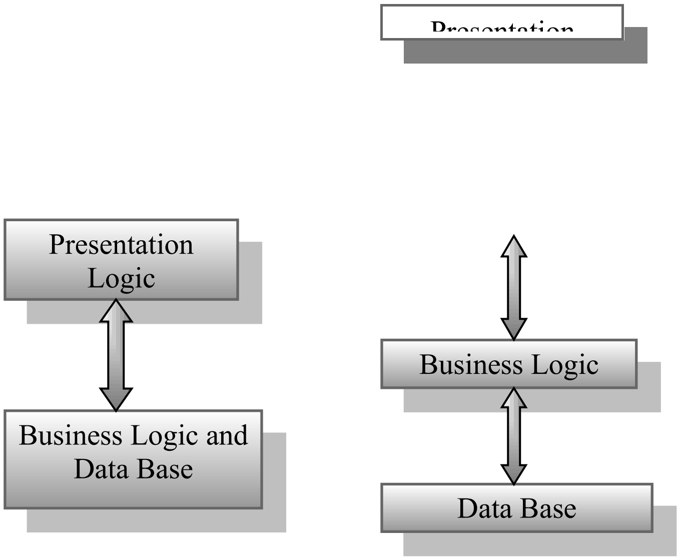

Figure1: 2 tier and 3 tier logical view of application (Ziemer, 2002).

Logical view can be generated in design and implementation phase of application development. Design frameworks improve designing process and productivity of designers and developers. Few architectural design frameworks are used to design web applications, such as: MVC, PAC etc. PAC (Presentation-Abstraction-Control) is more suitable for interactive software's. It has three components Presentation, Abstraction and Controller which have parent-child relationship and MVC (Model-View-Controller) is popularly used for web application designing. It makes application more maintainable, scalable and efficient. The MVC architecture provides a way for decomposing an application in to three parts: Model, View and Controller. This architecture is extensively used for separating the user interface from its application data and functionality (Schwabe & Rossi, 2010). The following figure illustrates the component architecture of MVC:

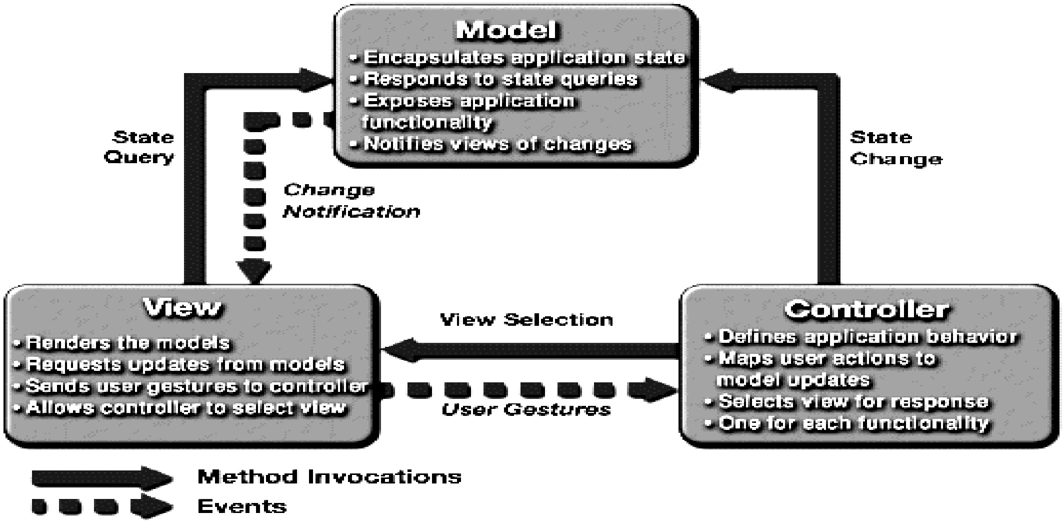
Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
State Change
Change Notification
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• One for each functionality
Method Invocations
Events

Figure 2: MVC Architecture framework (Thung & Ng Chu, 2010)

MVC has three components Model, View and Controller for different kinds of tasks (Schwabe & Rossi 2010).

Model: A model represents an application's data and contains the logic for accessing and manipulating that data. Any data that is part of the persistent state of the application should reside in the model objects. Model services are accessed by the controller for either querying or effecting a change in the model state. The model notifies the view when change of state occurs in the model.

View: The view is responsible for rendering the state of the model. The presentation semantics are encapsulated within the view, therefore model data can be adapted for different kinds of clients. The view modifies itself when a change in the model is communicated to the view. A view forwards user input to the controller.

Controller: The controller is responsible for intercepting and translating user input into actions to be performed by the model. The controller is responsible for selecting the next view based on user input and the outcome of model operations.

MVC has been adopted by all the major technology players of market, for providing programming environmental support for application development. It provides better separation of application logic. However when different kinds of web applications are developed using MVC framework, and then becomes necessary to find out various disadvantages of MVC, which may arise due to size and nature of the application. Various issues like performance, modularization of components, usability, maintainability etc, relating to MVC could be improved by analyzing web application architecture on the basis of different aspects associated with the size and nature of the application. The areas with scope for further improvement could be identified through such an analysis and then necessary improvements may be done for refinement of architecture to resolve the identified limitations, so as to improve performance, usability and maintainability of application.

2. Review of Literature:

Kunhua et al. (2010) have reported that the response time of web application depends on number of concurrent users; as the number of users increases up to a certain limit, response time increases rapidly. There are several other parameters also on which performance of web application is dependent, viz., throughput, system resource utilization, number of concurrent users, HTTP transaction /sec, number of sessions /sec etc. According to Kunhua et al. (2010), response time can be expressed as the sum of network latency and application latency:

Response time = Network latency + Application latency

They proposed a solution to reduce network latency by reducing distance between client and web server and by enhancing switching equipment performance. The solution to reduce application latency proposed by them was to reduce round trips and optimize data base queries. However the authors did not provide detailed reasons underlying the application latency and possible solutions to reduce the same. Performance of a web application can be improved by reducing the application latency, for which one of the possible solutions is to refine design framework of the web application.

Popularly MVC framework is used for web application designing. Leff & Rayfeild, (2001) worked on reduction of round trips (application latency) and have given a flexible web application programming model(FWAP), which could be used on thick client and thin client deployment model. They worked on the following two types of MVC architecture:

1. **Single MVC (SMVC):** It is a classical MVC framework in which all the three components reside at server side and execute in a single address space and generate view for clients. Hence it uses thin-client deployment architecture.
2. **Dual MVC (DMVC):** In DMVC architecture controller and model components resided at both the client and the server side. Either the client or server generated the needed view but it required model synchronization and transparent controller delegation, which was provided by FWAP. This framework required thick client deployment architecture.

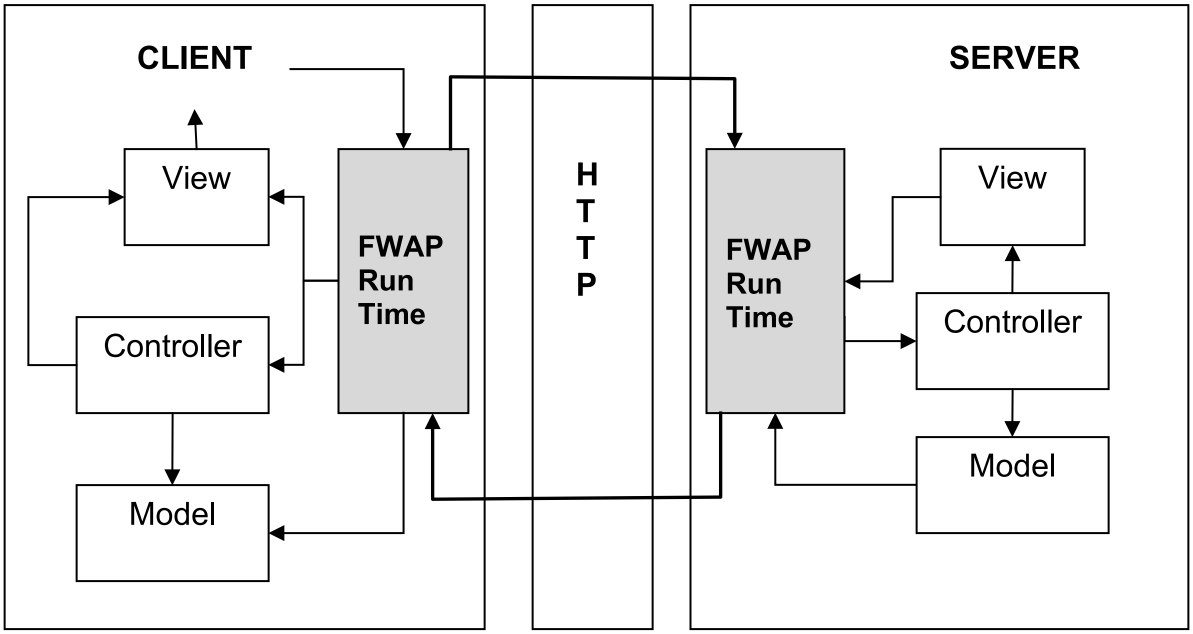

Figure 3: Simple Application deployment to Dual MVC architecture (Leff & Rayfield, 2001)

It has been proposed that when a small amount of data returns from data base as a result of request processing, DMVC should be used to reduce the round trips and when amount of data is high, SMVC should be used. However one of the limitations of the work of Leff & Rayfield, (2001) is that it was not known in the design phase that how much data would be generated by the database as an output of any transaction. As a practical consideration, if a web application have different types of transactions which generate variable amount of data, the question arises which approach should be followed? DMVC or SMVC. This question has not been addressed by them.

Another very important limitation associated with MVC framework is that it does not provide context related information to the client. In order to solve this problem Schwabe

& Rossi, (2010) proposed their own architecture framework OOHDM-JAVA2, which is based on object oriented hyper media model(OOHDM). To overcome the above limitation, they proposed another Navigational layer which may encapsulate all navigational logic in their new architecture. Schwabe & Rossi, (2010) worked on java based application so they had named it OOHDM-JAVA2.

OOHDM-JAVA2: OOHDM–JAVA2 extends the idea of MVC by using nodes clearly separated from their interfaces. Schwabe & Rossi, (2010), introduced the concept of navigational objects, which meant navigation may be context dependent.

This architecture also provided facility to define objects that implement the navigational view interface in generic manner and then refined these objects by adding application specific behavior. The OOHDM-JAVA2 architecture proposed by Schwabe & Rossi (2010) is shown in figure 4:

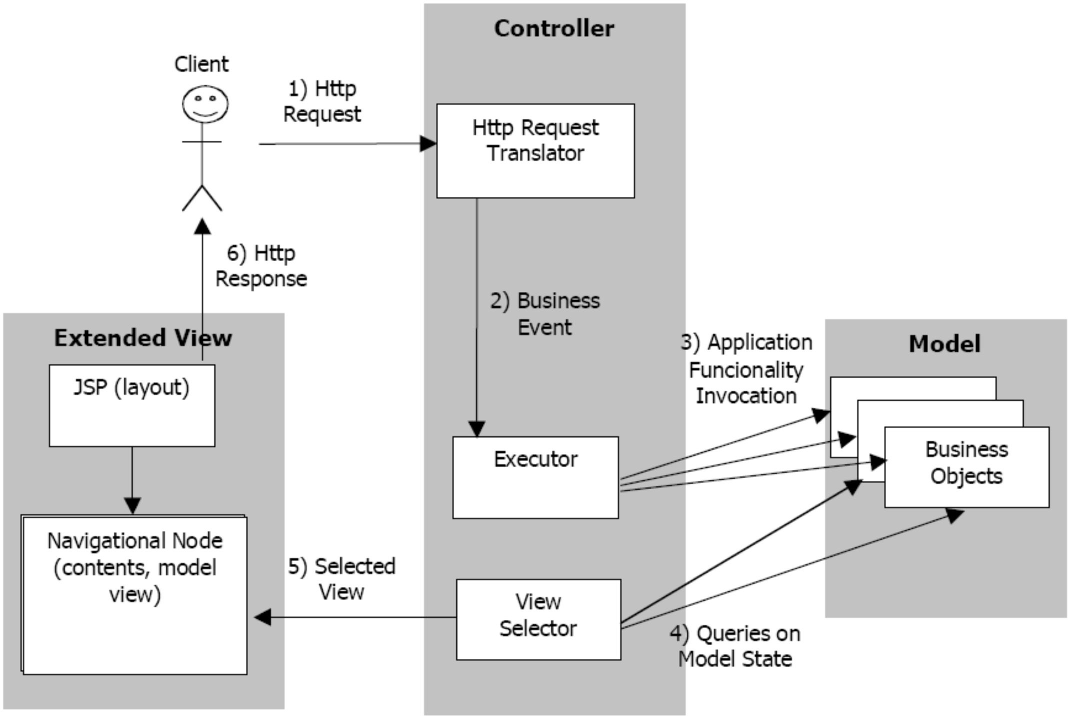

Figure 4: OOHDM-JAVA2 Architecture (Schwabe & Rossi, 2010)

However in OOHDM-JAVA2 architectures it is not specify how navigation node handle dynamic response generation by mapping different resources. Another limitation of MVC is that JSP files are cannot be edited automatically. To solve this problem, Huang &

Zhang (2008), proposed a new improved architecture XMVC based on MVC and XML technology, shown in figure 5.

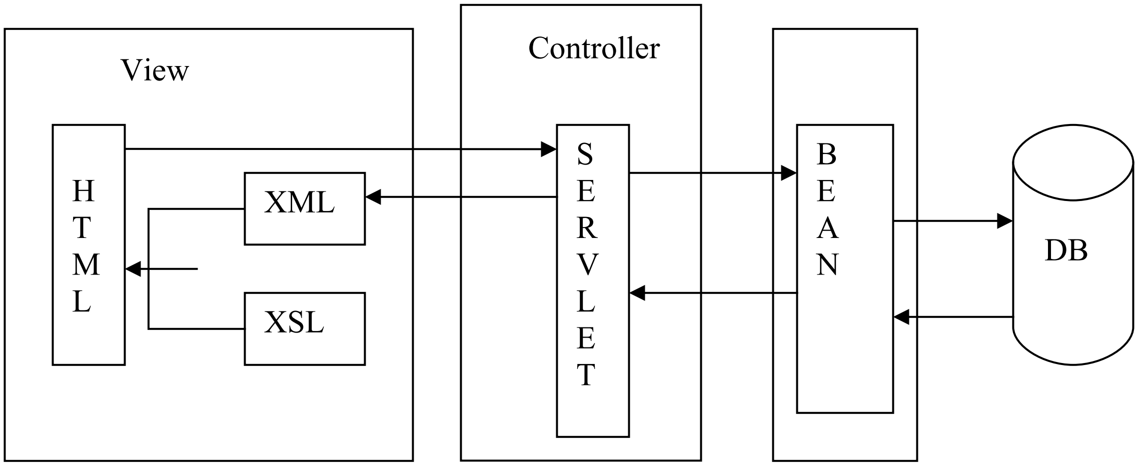

Figure 5: Improved MVC-XMVC Model (Huang & Zhang, 2008)

.

According to the proposed framework, the controller of XMVC is made up of many servlets. It receives HTTP requests from the client browser and called different beans to complete the relevant operations and finally a combination of XML string and Extensible Stylesheet Language (XSL) file is sent back to client browser from bean. In this model html pages can be generated by two different methods:

1. The combination of the XML technology and XSL is made for the input to XSL Transformation (XMLT) server/ processor and output is in the html format.
2. The path of XSL is added into XML as style sheet element and XML is sent to client browser, then the client browser downloads the XSL according to path and finally XML is converted in to html page.

To prove the efficiency of their model they developed a student information system on the basis of XMVC model, in which they designed their own controller and model. They used controller to establish database connection and the model returned XML string. However the limitation of Huang & Zhang's work is that they did not consider the time required for XML parsing at client side. Data format conversion and XSL file downloading time at client side decreased the performance of application. They developed a student information system application using the new framework proposed by them and claimed that it was more efficient.

Even though research work has been successfully carried out in the past for the improvement/enhancement of MVC framework, as is evident from the above mentioned studies, still there is still enough scope for further refinement of the same. The issues that could be further addressed include controller location for different types of transactions, separation of different components, round trips reduction during processing of requests of the clients, thick client performance issue etc. Improvement in these areas may further improve MVC framework leading to better web application design.

3. Motivation:

Web applications are distributed and network oriented in nature. They provide different services to various clients at their different geographic locations. Since last few years as the user requirements of different services have increased, various business organizations are willing to fulfill such requirements by employing web applications. Now-a-days web applications have become more popular as compared to traditional software.

As Web applications have become more flexible and dynamic for end users, the size and complexity of applications has also increased. Due to large number of concurrent operations handled by an application, performance of web applications is a critical issue for both the service providers and the end users. Hence, in order to improve the performance and maintainability of a web application an efficient architectural framework becomes very necessary.

MVC architecture is quite popular between different vendors and developers due to its functionality of dividing an application into different types of logical components/layers according to their operations and as it also provides efficient resource mapping mechanism using controller. However, in spite of the above mentioned advantages of MVC framework, there are a few limitations which need to be addressed for further improvement of MVC framework. A few of these are as given below:

1. It does not address navigational aspect of web applications (Schwabe & Rossi, 2010).

2. Context related information needs to be provided to the user, provision for which is not available in MVC.
3. View generation page file cannot be edited automatically (Huang & Zhang, 2008).
4. No clear separation between business logic and controller.
5. It has large controller classes [12].
6. It increases the user interface code, which in turn increases complexity in debugging (Huang & Zhang, 2008).
7. Models are just divided in to two functional parts server side operations and database operations. Whereas they could be further divided for different functional operations.

The above mentioned limitations provide a wide scope to analyze and refine the MVC architecture further to improve web applications.

4. Objectives:

The objectives of the proposed research are:

1. Detailed analysis of existing web application architectures.
2. Evaluation of related ongoing research work on web application architectures.
3. Analysis of the performance of MVC architecture framework, by performance analysis of application developed on the basis of MVC.
4. Identify the areas of improvement in framework, such as: location of controller, decomposition of controller, decomposition of model, Server side validation handling, etc.
5. Propose refined architecture framework.
6. Performance analysis of the proposed framework.
7. Comparative performance analysis of existing and refined architecture frameworks to validate the proposed improvements.

5. Methodology:

To successfully carry out the proposed research, it requires detailed analysis of existing web application architectures being used in the on going research works by different researchers in their respective frameworks. Performance of web application is the major concern, so it requires analyzing web application performance using available performance testing tools. There is wide scope of improvement in MVC.

We shall attempt to improve upon the MVC architecture in view of the identified limitations of MVC and propose a better improved and refined framework.

Different steps of the methodology are as follows:

1. Analysis of existing web application architectures: To carry forward this research we propose to analyze web application architectures in detail and simultaneously keep a track of ongoing research work on web application architectures by referring to different research papers appearing in research journals and proceeding of various conferences.
2. Performance evaluation of MVC: We propose to analyze the performance of web applications which have been developed using the existing MVC architecture, by using performance testing tools in a simulated environment.
3. Identification of the areas of improvement in framework: On the basis of previous researches on web application architecture and new challenges which appear due to diversified requirements of customers and users, we need to identify areas of improvement in existing framework.
4. Propose refined architecture framework: To overcome the limitations of existing web application architecture and to fulfill new requirements, we need to propose a refined architecture framework.
5. Testing of the refined architecture framework: To test for improvements we need to analyze performance of the proposed new architecture framework.
6. Comparative analysis of results: Logical and quantitative comparative analysis needs to be done by using suitable data analysis tools to prove the suggested refinements.
7. Conclusion: On the basis of comparative analysis of performance a suitable conclusions regarding the betterments achieve shall be drawn , limitations if any shall also be put on record.
8. Report Generation: All the above research activities and results have to be arranged in a systematic and structured format in the form of a thesis. New results shall also be reported in research journals / conferences in the form of research papers.

6. Plan of work:

We can divide the proposed research work into different sections and broadly assign certain time frame to each of them. The following table shows a broad division of time required to complete each activity.

Table 1: Phased plan for proposed research problem.

Sr. No.	Tasks	Time (in Months)
1.	Review of related ongoing research work on web application architectures	2-3
2.	Sample Web application development(if needed)	3-4
3.	Analysis of the performance of web application before refinement.	1
4	Identification of areas of improvement and refinement of MVC framework accordingly.	6-7
5.	Modification of sample web application according to the proposed refined framework.	1-2
6.	Analysis of the performance of web application after refinement.	1
7.	Comparative result analysis of performance before and after refinement of framework.	1-2
8.	Thesis Preparation	3-4

7. Tools and Techniques:

To carry out the proposed research we need tools for performing various tasks, a brief summary of the same is given below:

1. Performance evaluation: For performance testing there are various automated tools available in market. We can choose the appropriate tool according to our

requirements. Some of these tools, both commercial and open source, are listed in the following table:

Table 2: Performance testing tools

Sr.No.	Open Source	Commercial
1.	Jmeter	Load Runner
2.	Loads	Silk Performer
3.	Rubies	Qengine

2. Statistical analysis: To perform comparative data analysis of performance evolution of existing and refined architecture frameworks we need to use suitable statistical analysis tools, such as:
    1. MS-Excel
    2. MATLAB
    3. SPSS
    4. XLSTAT
    5. Statistica 10
    6. SAS
3. Report Writing Tools: To express different activities of the proposed and their results will be effectively expressed using following tools.
    1. Microsoft Word with Visio.
    2. Coral Draw/ Photoshop.

8. References:

[1] Das Budhaditya and Mane Priyanka (2008), “Scalability Factor of JMeter In performance testing Projects”, Step-In Conference-Performance testing 2008, Pune URL: http://www.stepinforum.org/Perf_Testing/conf_/DasMane_aztec.html

[2] Huang Shu-qiang and Zhang Huan-ming (2008), “Research on Improved MVC Design Pattern Based on Struts and XSL”, International Symposium on Information Science and Engineering.Vol.1,pp.451-455.

[3] Kunhua Zhu, Junhui Fu and Yancui Li (2010), “Research the performance testing and performance improvement strategy in web application”, 2nd international Conference on Education Technology and Computer (ICETC).vol 2, pp. 328-333

[4] Leff Avraham and Rayfield James T. (2001), "Web-application development using the Model/View/Controller design pattern," Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International ,pp.118-127. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=950428&isnumber=20561.

[5] Menascé Daniel A (2002),”Load testing of websites”, IEEE Internet Computing, Issn: 1089-7801 vol 6 pp 70 - 74
URL: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1020328

[6] Sauer Stefan and Gregor, “MVC based modeling support for embedded Real time System”,http://www.uni-paderborn.de/cs/ag-engels/Papers/19

[7] Schwabe Daniel and Rossi Gustavo (2010), “The Object-Oriented Hypermedia Design Model (OOHDM)”, *proceedings.informingscience.org/.../IISITv5p729-740Karim475.pdf.*

[8] shah Abad Ali , Mustafa Ghulam , Hussain Khadim and Ali Amjadi(2007), “A Strategy for testing web based software ”, *Information Technology Journal*, Issn:18125638 Vol 6 Pages/rec.No:74-81 URL: http://www.doaj.org/doaj?func=abstract&id=600724&openurl=1&uiLanguage=en

[9] Subraya B.M. and Subrahmanya S.V (2000) “Object Driven Performance Testing in Web Applications," The First Asia-Pacific Conference on Quality Software (APAQS'00.), pp.17-26,
http://www.computer.org/portal/web/csdl/doi/10.1109/APAQ.2000.883774.

[10] Thung Phek Lan, Ng Chu Jian, Thung Swee Jing, and Sulaiman Shahida (2010), "Improving a web application using design patterns: A case study," *Information Technology (ITSim), 2010 International Symposium in* ,vol.1, no., pp.1-6,
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5561301&isnumber=5561289.

[11] Yong Liu and Li Ke Xi (2010),"Design and implementation of the new Web application framework—Jemsf, " *Apperceiving Computing and Intelligence Analysis (ICACIA), 2010 International Conference on*, vol., no., pp.190-193, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5709880&isnumber=5709837

[12] Ziemer Sven (11, 2002), “An Architecture for Web Applications**”,** DIF 8914 Distributed Information Systems, teseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.4532.

[13] http://www.rorexperts.com/disadvantages-of-mvc-architecture-t1873.html