International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Modeling Web Applications : A Different Perspective

Mr. R. B. Kulkarni, Mrs. S. S. Ambarkar, Dr. S. K. Dixit

***Abstract* - Traditional web applications do not support reverse engineering capabilities and it become difficult to recover navigation, content from web applications. This paper describes web application modeling with different perspective .Conventional modeling either use UML or MVC This papers proposes web application modeling to recover these aspects with an hybrid approach using UML and MVC. Many models have been defined for capturing different aspects of a user interface. There has been several attempts to model a web application but none address the complete model recovery of web applications Although UML has been widely accepted as a standard language to specify software models, it was not designed with user interfaces in mind. This is precisely the reason why no particular model in UML is very successful in capturing details of user interactions and interface design We state out some issues in this paper related to UML modeling of a web applications and propose a hybrid method of modeling a structure of web page and simultaneously navigational aspect of a web page. A combination of UML and MVC is proposed in this paper which bring out the best of both. New stereotypes have been used to support the hybrid method of modeling. Our modeling method tries to capture structural, navigational and functional aspects of web application.**

Keywords - Web Applications, UML, MVC, DB4O.

I. INTRODUCTION

Web applications are becoming most important aspect of software application in the industry with many modeling languages proposed to handle the complexity of these applications and can handle the development, documentation and deployment. However, the complexity arises due to several factors, such as larger number of hyperlinks, more complex interaction. The consequence is that many web applications are poorly structured, which leads to difficult maintenance. The lack of abstraction also makes it difficult to maintain and evolve Web-based applications [2]. There has also been much concern on the quality of hypermedia systems being developed and the apparent absence of disciplined development practices also, "software crisis" is afflicting hypermedia systems development, shoddy project management, inadequate requirements analysis and planning, and ad hoc "quick and dirty" development approaches.[4]. One way for dealing with web application design complexity is to tehm model on a higher level of abstraction.

Model-driven techniques use in the area of web development, often transform a high-level input model to lower level models or executable code. Many of the model-driven methods of web engineering are tuned towards specific platforms or develop only certain parts of web applications. However, if a method aims at supporting multiple web platforms, it would need meta-model that describes all features of web applications and an abstract level. The suitability of such abstract models may be validated by transforming them to different platforms. Our model is to be mapped to specific editor developed by us. Modeling can helps to understand the same.

Web development tends to focus solely on the low-level implementation [6] using a diverse range of technologies [7], but at the same time must be accessible to a diverse range of platforms and devices [7]. It is this conceptual gap between application requirements and the implementation that needs to be addressed.

Existing solutions are woefully inadequate for modeling web applications [7]. In particular, solutions lack support for interactivity; cannot support dynamic content generation; ignore existing standards and frequently have no implementation support [5]. It is clear that since existing solutions cannot model web applications, industry will hesitate to use them. There is a significant gap between any existing methodology and the implementation of web applications.

The first step towards analysis and restructuring in the definition of a model representing the various entities involved in Web applications and their mutual relationships is part of the study. The models proposed in literature usually aim at describing the Web application from a logical point of view at a high level of abstractions, as required when the application is being designed. We therefore define our own model of UML and MVC to fill the gap which is put forth in the next part of our thesis On the contrary; we focus our model on the implementation of the site, which is the starting point for analysis. We have added one more feature of storing the contents in an object oriented database OODB for reusability.

II. OBJECTIVES

- Model should be complete i.e. the most important entities should be explicitly represented in the model.
- It should be possible to provide automatic support for its extractions.
- It should have Modularity.
- It should be possible to have proper navigational features of the web applications. Emphasis is on relation between page and link.
- It should be consistent, simple and elegant.
- Model should provide reusability and portability.

III. RELATED WORK

The increasing amount of web development process needs to be coordinated in a proper and systematic manner and there should be methodology established [3]. There are numerous different approaches to the modeling of Web applications. Some of them focus on the modeling notations, while others focus on the development process. The very first work is by Conallen [14] who defines a UML profile for Web application design The Web

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Application Extension (WAE) to UML enables to represent Web pages and other architecturally significant elements in the model alongside the "normal" classes of the model. Koch et. al. [18] described a hypermedia extension to the UML to model Web applications, focusing on the navigation and presentation aspects.

The Web Modeling Language (WebML) [11] is a notation for specifying complex Web sites at a conceptual level. It distinguishes between a structural model, a composition model, and the topology of links between pages. WebUml, an automatic tool to reverse engineer existing Web applications. [8] WebUml does extraction/reconstruction of a UML model for an existing Web application. The extracted information is particularly meaningful for the kind of legacy Web applications where the business logic is embedded into the Web pages, instead of more recent and layered Web applications where the business logic is implemented through server-side components

The UML-based Web Engineering (UWE) [9] approach presented by Koch [2001] and extended in subsequent papers [Hennicker & Koch, 2001; Kraus & Koch, 2002] supports Web application development with special focus on systematization, personalization and semi-automatic generation. It is an object-oriented, iterative and incremental approach based on the Unified Modeling Language [UML, 2001]. Ekaterina Gorshkova et. al [19] defines an UML extension capable to refine client part of the applications.

IV. UML Extension to MVC

Current modeling languages for web system development fall short of the requirements for modeling of web system development. The modeling concepts of UML are grouped into *language units* that provide users with the power to represent aspects of the system under study according to a particular paradigm or formalism. From the user's perspective, this partitioning of UML means that they need only be concerned with those parts of the language that they consider necessary for their models. If those needs change over time, further language units can be added to the user's repertoire as required [13]. The modeling languages with hypermedia basis are closely focused on information architecture whereas software system modeling languages are more focused on functional architecture [16][17]. We have tried to integrate best of both that is UML modeling for UI and MVC model to extract the other parts of web systems such as forms, database content which is more user friendly in MVC.

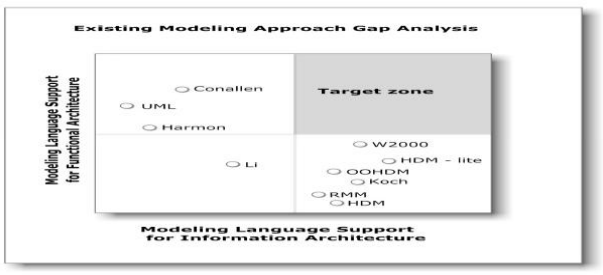

Fig.1. Existing modeling gap analysis [17]

V. Extension Model Structure

Based on analysis of the existing modeling approaches, we propose a UML extension with information modeling concepts taken from other modeling approaches, WebML in particular. [11] This option is chosen because UML notation is commonly used and accepted and it appears to provide support to system functional architecture model. It also approaches from hypermedia background demonstrate reasonably good support.

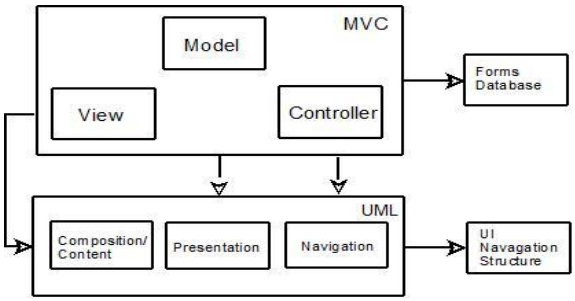

Fig.2. Extension structural diagram combining conceptual separation of MVC and UML

VI. UML Extension Mechanism

We concentrate on client aspects of the web application and provide the UML based framework for modeling the content of pages and navigation between them. The opportunity to separate the functional model of the user interface from presentation features is important. The diagrams generated from this case be given to UI designer as requirement. The very aspect of other content of particular page displayed to the user and data sources from other layers contribute to the content and data placed on the page and the structure of inter page links in the application two stereo types to model the web application are introduced.

A. Navigation diagram

B. Structural Diagram

The details of two types are as follows,

A. Navigation Diagram

When working with web sites often it is difficult to visualize the overall structure of the site. A basic node-link map is a great start, but to show every direct navigation pathway between every page, the node-link map quickly becomes confusing, even if the site is moderately complex. The answer to this problem is the Navigation diagram, which illustrates the navigation on a web page. You can also illustrate contact to a database. It demonstrates the possible user behavior in the web application. In fact, the navigational diagram is implemented as a special case of the UML state chart diagram.

Building a navigation model is helpful not only for the documentation of the application structure, it also allows for a more structured increase of navigability,[Modeling the User Interface of Web Applications with UML [18].

The navigation model tells which objects can be visited in a Web application. The data model of navigational

design contents web pages single and bi-directional hyperlinks.

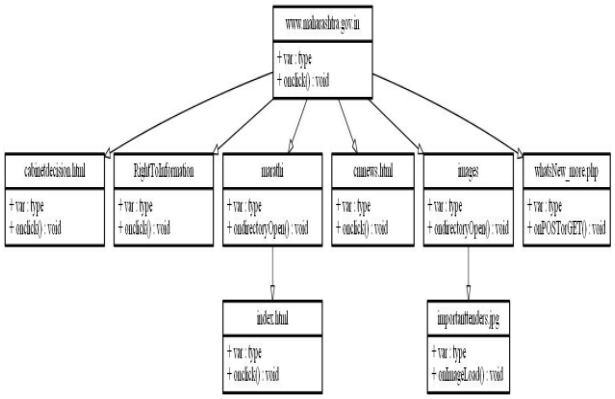

Fig.3. Example of the navigation diagram of maharashtra.gov.in prior 18th August, 2011

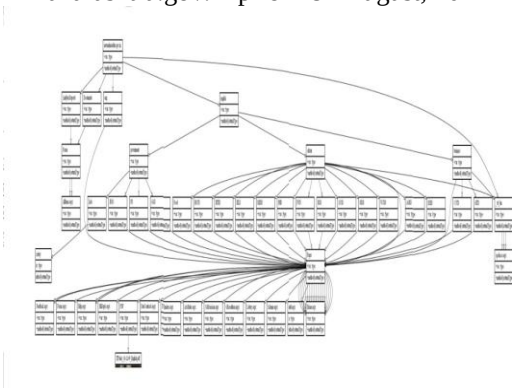

Fig.4. Example of the navigation diagram of maharashtra.gov.in current web application

We have generated two cases of navigation diagram of website of government of Maharashtra which shows how the navigation path is existing in the web application First diagram is the site taken prior to 18th August, 2011 and Second diagram is the new website which is now hosted by government.

B. Structural navigation

The structural or composition diagram is special case of class diagram. The diagrams used to express the structure of the web application. The structure of UI of the web pages can be represented and these diagrams identify the content of each page. Page corresponds to class. Structural diagrams portray the connections between the pages of web application and demonstrate the flow of data from one page to another. This diagram does not exist in the UML. It comes from WebML and covers one aspect of information architecture of Web application.

In our case, we are using it to express the structure of the web pages and identify their content: how they are connected together and what data is carried from one page to other. As the diagrams we show case two diagrams of the same web application which have changed in a period of the time the following diagram is the web application of government of Maharashtra prior to 18th august 2011 and second web application after 18th August 2011.

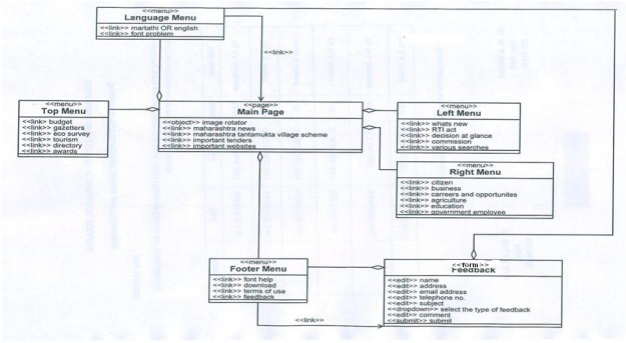

Fig.5. Example of the structural diagram of maharashtra.gov.in prior 18th August, 2011

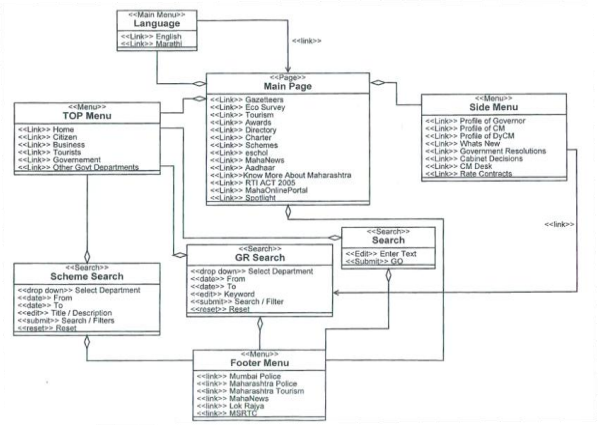

Fig.6. Example of the structural diagram of maharashtra.gov.in current web application

VII. Model View Controller

The MVC framework is a design pattern that is widely used for applications with a user interface component (1). MVC has three components whose operations are quite isolated from each other. In MVC, we divide our system into three parts in order to reduce complexity. MVC itself is a pattern (pattern is the proven solution for any problem in a particular context)implemented in the presentation layer where it handles user's interaction (controller) with a particular model through view. Almost all MVC based frameworks have the feature of URL routing that gives us more control over the URL we desire.

e.g. http://www.foo.com/content_id/edit

where 'content_id' may be unique ID and 'edit' may be controller action. The Modular design of MVC supports the design goal of reusable software. By following MVC, one can easily tests each and every part of the system.

As in the earlier case of UML we have just taken UI part in which we have taken navigation and structural view of web application. In MVC model we have extracted the different components of web applications like forms, database and other entities to complete recovery of web applications.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Since we tried making our MVC model language neutral it was not feasible to do so we have adopted a MVC controller which can applied to the three popular languages which are JSP, Asp.net and PHP. The following diagrams show the controller error code generation.

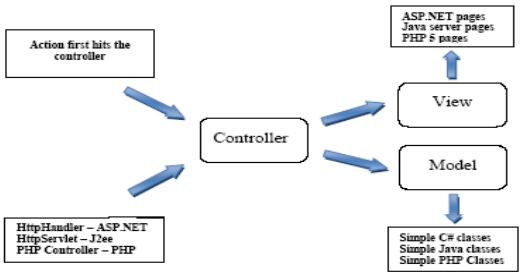

Fig.7.1 Controller view model of MVC which shows view model and controller part.

In the later part of diagram, we show error pages, object class entities, action logic and HTTP handler. We have implemented this package which shows the recovery of form and database content.

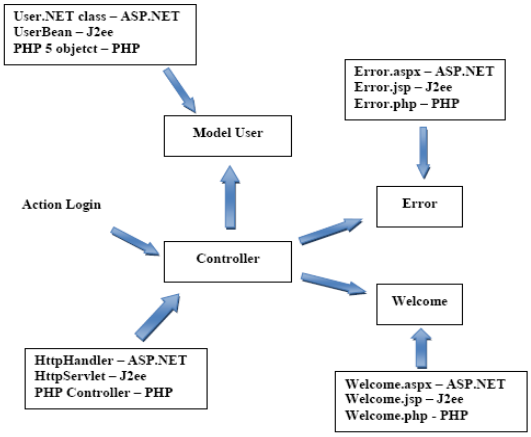

Fig.7.2 Controller view model of MVC which shows error pages, object class entities, action logic and HTTP handler

A. Extracting forms (views)

By Analyzing the DOM (Document Object Model) of a Web Page for <form>

We extract the selective markup of <form> and classify it as view of Web application.

B. Extracting Database

Database schemas and their field types are identified by analyzing the Database initialization scripts present in server side code of web application.

C. Methodology

The diagram which is generated from WebModel follows the process of web application under consideration is parsed using TINTFU[] we generate the tree map of the web application then tree map is converted into a file format named dot file which is an input to the Graphviz package[] navigation diagram is obtained, the dot file is then parsed by XML to HTML parser and the output is fed to the Graphviz package to generate structure diagram. The process is automatic with very less human intervention.

After generation of the diagrams and MVC we have an option of saving the content in the object oriented database OODB db40.

We used db4o[20] an open source object database as OODBMS.

D. DB4O

Overview and Features

DB4O stands for **D**ata**b**ase for **(4) O**bjects and is an Open Source object-oriented database developed by db4Objects, Inc

- It eliminates the object-relational mismatch. And It has ACID properties
- Its faster performance than object-relational mappers and can run in the same memory process It gives one the ability to modify, optimize and integrate the database engine easily according to one's specific needs.

Db4o stores objects the way they are defined within the application (Versant 2009). Therefore, it is easier to retrieve objects from the database using simple instructions within the application.

Characteristic of DB4O

Db4o supports all the basic functions required of an object database (inserting or storing, querying, updating and deleting), used in Java or .Net programming environments. The scalability of Db4o was tested with the creation of a number of objects. Db4o provides its source code to users and developers for modification. The prototype allows automatic generation of web pages and transferred to object oriented database db4o. We have chosen two approaches to interact with web applications which are presented in case study.

Advantages of storing in OODB

OODB base generation allows user to interchange between XML representation and ODMG object model. Data in OODB can automatically translate to web pages.

It Support for multiple use and ease of updating

E. Case Study

We have designed the software code for the MVC classification which works with two different options

1. Option 1 when web application is downloaded and is classified in MVC and is then transferred to the OODB db40 for further analysis and can be ported to our editor WebReen
2. Option 2 When the web application is downloaded it is directly transferred to the OODB db40 and then the content is analyzed for MVC classification

F. Option 1

MVC Classifier – Pseudo Algorithm

1. Find out all form instances.
2. For each form instance identified find the file of URL specified in action attribute of form instance.
3. Fetch the file identified in step 2 and classify it as Model + Controller logic. Repeat for all files generated in step 2.
4. Retain the form instance as View.
5. For all the pages generating dynamic content, identify the script client / server and categoric it has Model.

6. Search the Model + Controller files for possible redirection scripts and categorize them as Controller.
7. End

First web application is MVC classified and then ported to OODB db40

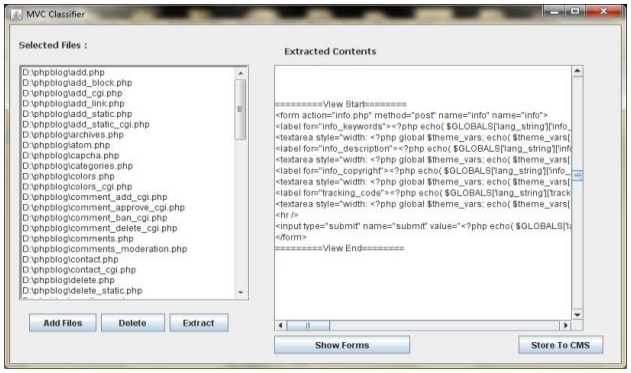

Fig.7.3 Step 1 : Selecting content to be classified as MVC

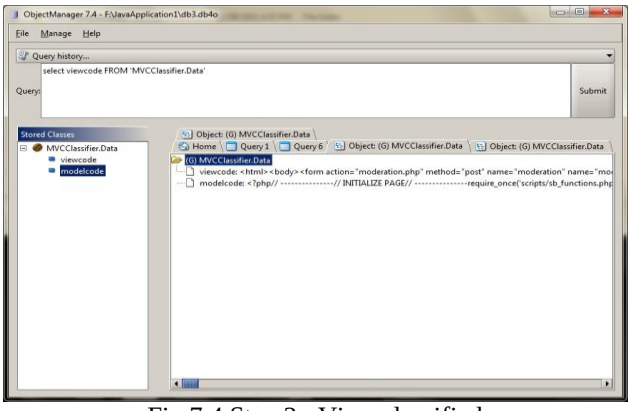

Fig.7.4 Step 2 : View classified

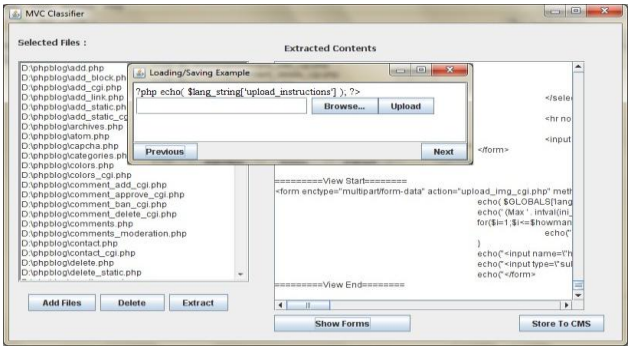

Fig.7.5 Step 3 : Browsing data from db4O

Option 2

First data consolidated to OODB db4o then converted to MVC

When content of the web application downloaded are sent directly to the object oriented database Db40.

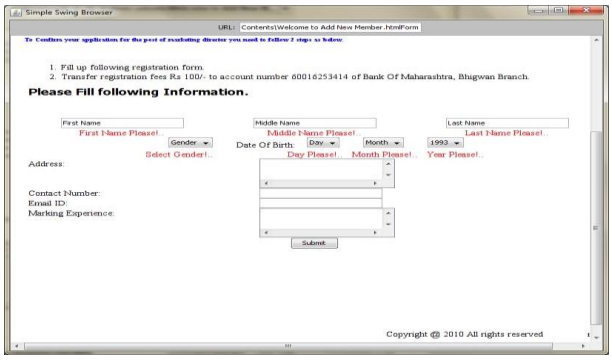

Fig.7.6 shows the extraction of the view of the OODB

In above fig 7.6.which has been used since the contents of the web application are in Db4o we extract all the contents.

Fig.7.7 shows the markup of the view extracted from the OODB db4O

In above fig.7.7 where this code can be used for further processing by our editor which is accepting this as input to reconstruct the web application.

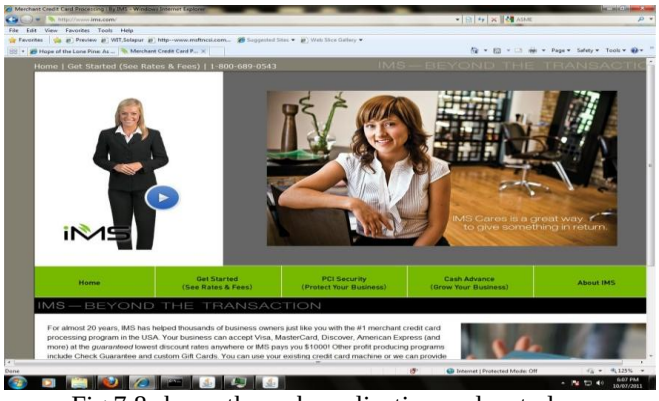

Fig.7.8 shows the web application under study

In above fig 7.8 which is used to be stored as it is in OODB before any classification code is applied to it.

Then we read the contents through our software code to be sent to further processing to our editor WebReen which shows the content inside the OODB file.

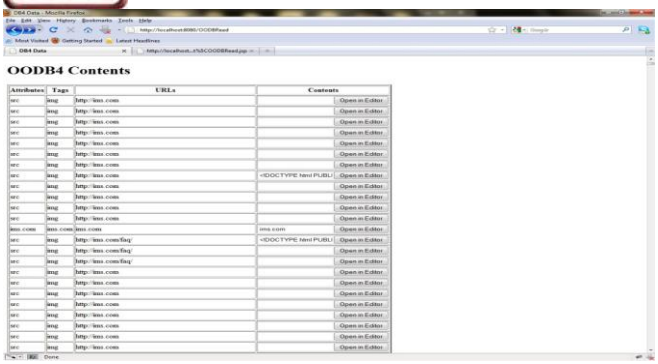

Fig.7.9 shows data content of web application in OODB

In above fig.7.9 seen the data content of web application in OODB

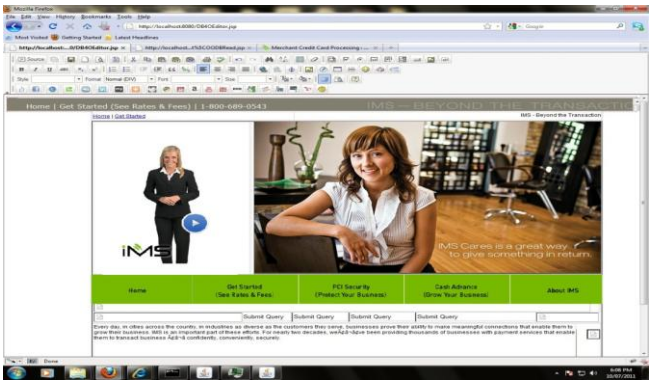

Fig.7.10 shows the reconstructed page

In above fig 7.10 seen the reconstructed page of the web application retrieved from the database in to our editor for restructuring and further changes to the same.

VIII. CONCLUSION

The model consists of set of diagrams describing navigation and structural of dynamic web application exposed to client. The resulting diagrams can be used for building HTML prototype of the application and for generating client-side and server side code. Major advantage of this approach is it provides uniform description of all components and overcomes gap between application logic and presentation features. With combination of both MVC and UML we have mapped the web application right from its UI to the other contents in the web application.

In case of OODB, we have for the first time converted and transferred web application as it is to OODB database and also have a feature which can transfer the contents after applying MVC aspects to it, which will help the developer for effective management of web elements as objects. The tool to translate the contents to OODB targets user such as web site administrator who are faced with heavy workload of website generation, restructuring and maintenance. This tool greatly facilitates the automatic website generation as the contents are stored in OODB and which can be accessed, removed and follows the principle of CRUDE.

Further we have modeled a web application using the UML diagrams for which we have used VIPs segment or tool to segment and the segmented parts are sent to XML to HTML convertor and the converted file is given to Graphviz package to get the exact UI representation and the navigation structure diagrams.

In the second phase of recovery of web application we have two options to use the content

1. MVC classified data of the web application is sent to OODB
2. Web application if first ported to OODB and then after recovery it is MVC classified.

In both the cases, we can directly port the web application to two popular content management system Joomla or drupal.

The process of recovery of the web application and transfer to OODB and our editor WebReen is automatic.

REFERENCES

[1] E. Mendes, N. Mosley, and S. Counsell, “The Need for Web Engineering: An Introduction,” in Web Engineering, E. Mendes and N. Mosley, Eds. Springer, 2006, pp. 1–27.

[2] H.W. Gellersen and M. Gaedke, “Object-Oriented Web Application Development,” IEEE Internet Computing, vol. 3, no. 1, pp. 60–68, 1999.

[3] M. J. Taylor, J. McWilliam, H. Forsyth, and S. Wade, “Methodologies and Website Development: A Survey of Practice,” Information & Software Technology, vol. 44, no. 6, pp. 381–391, 2002.

[4] M. Lang and B. Fitzgerald, “Hypermedia Systems Development Practices: A Survey,” IEEE Software, vol. 22, no. 2, pp. 68–75, 2005.

[5] S. S. Selmi, N. Kra¨ıem, and H. H. B. Gh´ezala, “Toward a Comprehension View of Web Engineering,” in ICWE, 2005, pp. 19–29.

[6] R. Gitzel, A. Korthaus, and M. Schader, “Using established Web Engineering knowledge in model-driven approaches,” Sci. Comput. Program., vol. 66, no. 2, pp. 105–124, 2007.

[7] J. Offutt, “Quality Attributes of Web Software Applications,” IEEE Software, vol. 19, no. 2, pp. 25–32, 2002.

[8] C. Bellettini, A. Marchetto, and A. Trentini, “WebUml: reverse engineering of web applications,” in SAC ’04: Proceedings of the 2004 ACM symposium on Applied computing. New York, NY, USA: ACM, 2004, pp. 1662–1669.

[9] N. Koch and A. Kraus, “The Expressive Power of UML-basedWeb Engineering,” in IWWOST’02, 2002, pp. 105–119.

[10] G. Kappel, W. Retschitzegger, and W. Schwinger, “Modeling Ubiquitous Web Applications: The WUML approach,” in International Workshop on Data Semantics in Web Information Systems (DASWIS-2001), Yokohama, Japan, 2001.

[11] S. Ceri, P. Fraternali, and A. Bongio, “Web Modeling Language (WebML): A Modeling Language for Designing Web Sites,” in Proceedings of the 9th international World Wide Web conference on Computer networks. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 2000, pp. 137–157.

[12] G. Rossi and D. Schwabe, “Model-Based Web Application Development,” in Web Engineering, E. Mendes and N. Mosley, Eds. Springer, 2006, pp. 303– 333.

[13] Linda Heaton, “Unified Modeling Language (UML): Superstructure Specification, v2.0,” Object Management Group, Tech. Rep., 2005. [Online]. Available: http://www.omg.org/cgi-bin/doc?formal/05-07-04.

[14] Jim Conallen (2003), Building Web Applications with UML, Addison-Wesley, 2nd edition.14. 2003, OMG Unified Modeling Language Specification, version 1.5, Object Management Group, http://www.omg.org/cgi-bin/doc?formal/03-03-01.

[15] Behshid Behkamal, Mohsen Kahani, Mohammad Kazem Akbari, (2008), Customizing ISO 9126 quality model for evaluation of

B2B application, Information and Software Technology, S 1 (2009), p. 599-609.
[16] Gu. A. Henderson – sellers, B, Lowe Web modeling language gap between requirements and current exemplars. In : Aus Web (submitted, 2002).
[17] David Lowe, Brian Henderson-Sellers, Alice Gu., Web extensions to UML : Using the MVC Traid, University of Technology, Sydney, Australia.
[18] Nora Koch, Hubert Baumeister, Rolf Hennicker, Luis Mandel, (2000), Extending UML to Model Navigation and Presentation in Web Applications.
[19] Edaterina Gorshkova, Boris Novikov, (2003), Exploiting UML extensibility in the design of we information systems.
[20] http://www.db4o.com

AUTHOR'S PROFILE

Mr. R. B. Kulkarni

He is a Assistant Professor in Computer Science & Engineering Department, Walchand Institute of Technology, Solapur, Solapur University, Maharashtra - 413006, India. He has completed BE(CSE) in 1991 from Walchand Institute of Technology, Solapur. ME (CSE) in 2005 from College of Engineering, Sangli. (Ph.D.) thesis submitted in Jan 2012. His email id is raj_joy@yahoo.com His topics of interest are Web Engineering, Web of Things and Multimedia and VRML. He is Life member of ISTE, CSI and published more than 15 papers in National/International Journals.

Mrs. S. S. Ambarkar

She is a Assistant Professor in Computer Science & Engineering Department, Walchand Institute of Technology, Solapur, Solapur University, Maharashtra - 413006, India. She has completed her BE(CSE) in 2004 from Kolhapur Institute of Technology, Kolhapur, Shivaji University, Maharashtra. M.Tech. (CSE) in 2010 from PDACOE, Gulbarga, VTU, Belgaum. Her email id is supriya.ambarkar@gmail.com. Her topics of interest are Web Technology, Networking, Database Engineering. She is a life member of IE and published more than 4 papers in International Journals.

Dr. S. K. Dixit

He is a Professor and HEAD (Electronics & Telecommunication Department), Walchand Institute of Technology, EXTC Dept, Solapur University, Solapur, Maharashrta – 413006, India. He has done his BE at Walchand College of Engineering, Sangli and ME from Walchand College of Engineering Sangli, and Ph.D. from Shivaji University. His email id is skdixit@gmail.com. His topics of interests are Robotics and Embedded Systems. He is a Ph.D. Guide in Solapur University.