Model–Driven Content Management of Web Applications Using Atlas Transformation Language

Ms. Amruta Pawar, Mr. R. B. Kulkarni

Solapur University, Solapur,

Maharashtra, India

Abstract— Development and maintenance of Web application is still a complex and error-prone process. We need integrated techniques and tool support for automated generation of Web systems and a ready prescription for easy maintenance. The MDA (Model Driven Architecture) approach proposes an architecture taking into account the development and maintenance of large and complex software. In this paper, we apply MDA approach for generating PSM (Platform Specific Model) MVC 2 Web Implementation from UML design. For this we have developed two meta-models handling UML class diagrams and MVC 2 Web separately and then we have set up transformation rules using ATL. To specify the transformation rules we used UML profiles. The result of this transformation is an XMI file which is further used to generate code ingredients of web applications

Keywords— MDA, Meta-models, Transformation rules, ATL, MVC 2 Web, UML Profiles.

I.   INTRODUCTION

Model Driven methodology is the technique used to assure modularized existence of a web application. A Web Application is a software program that is stored on a remote server and delivered to the end user over the Internet through a web browser. It is an application that uses web browser as a client and is made up of logical units called tiers namely; presentation, application and storage. Web browser is the first tier, dynamic web content determines middle tier and database or storage is the third tier.

With increased complexity of web applications it becomes necessary to provide quick and easy access to the desired content. This content of the web applications changes frequently and according to the user specification. It becomes tedious task to make the content accessible dynamically. Thus if the entire application is modularized and modeled according to the functionalities the dynamical feature of it can be preserved. Various technical areas like software engineering, hypertext engineering, data design, graphic design and presentation modeling merge in to achieve the need of complex web applications.

Model-View-controller (MVC) is a software architecture that helps separate information and its representation from its resulting interaction with the user. Thus the primary task of the project to which this design document belongs is converting conventional web application into MVC architecture.Web applications today are using this concept for reusability and modularity. Model, view and controller are the three aspects of the MVC architecture that deal with data definition, navigation, presentation and so on. Together they are responsible to generate the final code for the user access.

Atlas Transformation Language, an Atlas INRIA & LINA research group product is a model transformation language specified both as metamodel and as a textual concrete syntax. It is a declarative style of transformation generally concerning simple mappings whereas complex mappings are handled by imperative constructs. ATL programs contain rules that define how target model elements are obtained from source model elements. This paper aims at the automatic management of user accessible content of web applications wherein large amount of data is involved. Model-driven architecture is to be used to manage the content code generation and help reuse the specifications across numerous applications as well as reduce repetitive development.

II.   IDEA OF PROJECT

A. Problem Statement

The paper proposes to implement a model-driven methodology that transforms a class diagram to a target model which takes an XMI format. This XMI file can be subsequently used to generate the source code of an MVC based web application. Associations between classes are given a great emphasis. The idea is to reuse common aspects across different applications as rules and reduce repetitive tasks.

B. Objective and Scope

Model driven methodology is to be used to transform a class diagram to a target model. This is going to ease the task of repetitive design along with the task of testing the application. The project intends to ease the testing and maintenance

of an application. Making it model driven, this project aims at reliving the task of repetitive programming and tedious testing procedure. The idea is to begin with UML representation of classes and then using the concept of *Atlas Transformation Language*, generate a target model and use it to develop source code of MVC web application.

III. LITERATURE REVIEW

The evolution of web applications is marked by Web Engineering methodologies which help develop them systematically merging various areas of technology like, software engineering, human computer interaction, graphic design etc., [1]. Also an application that appeals to the user's needs aesthetically proves quite beneficial for their stay with the site. The impact of purpose on user preferences is very large. The content orientation as well attracts users and needs a lot of work behind [2].

Model-driven approach imports a number of models into real web application. The method separates various aspects of web design i.e., process view, content view, navigation view and presentation view [3]. This approach has been saving a lot of repetitive work. An example approach is MVC architecture. Using MVC architecture allows the creation of large and complex, but flexible and easy to maintain web applications. These applications are easy to develop and maintain because the user interface code is separate from the database access code. This makes the code much easier to write, test, and debug. Model driven architecture is higher version of MVC itself. With regards to web application development, study so far has been accompanied with MVC architecture. Model driven method may prove to be a better approach towards the same because of all the abstract models used. These help in separating varied tasks. [7].

A study of model driven architecture with MVC presents a work of model-driven approach to developing web applications based on Ubiquitous Web Application conceptual design methodology and MVC architecture. The UWA design framework along with its methodology and models is particularly suited for designing web applications which are intended to be accessible by different user types in different usage contexts and with different goals [11]. The combination of characteristics of UWA and the MVC architecture, the resulting approach is particularly suited for developing ubiquitous web applications as well as for supporting their maintenance and evolution.

The MVC design pattern divides a web application into three parts; handling the user-interactions, computations and output display. This separation favours architectural reusability to great extent. The designers need to decide which of the components of the architecture are to be partitioned between the server and client. For a designer is very difficult to incorporate future changes in policies related to partitioning in application. A solution to this problem is to duplicate the Model and controller components at both server and client respectively. This too may add problems like delay in fetching data, security and scalability issues, etc. Hence to overcome this drawback, a new architecture SPIM has been proposed that deals with the partitioning problem in an alternative way. Third party mash up technique is applied to client views hence reducing the task of the server or storage. This approach shows tremendous improvements in performance when compared with a similar architecture. Hence it can also be considered to be an improvisation with MVC. Certainly model driven methodology can reduce the task further by simply modularizing the application itself In literature the problem of automatic display layout is applied for fields like computation of graph based diagrams, layout of graphs, document presentation and so on [5] and [8]. The problem is realized by means of constraint specification or a set of rules, fixed or customizable.

Research presents a solution to web based system evolution which consists of components like understanding, representation and evolvable framework development. Everything here thence can be modelled. Displace of the content of applications is made easy because of model-driven approach. It also helps compose and reuse components. But development and maintenance of Web application is still a complex and error-prone process for which we need integrated techniques and tool support for automated generation of Web systems and methods for easy maintenance. The MDA approach proposes an architecture taking into account the development and maintenance of large and complex software. Model transformation then becomes a necessity. Hence a model transformation language like ATL is widely used [14] and [15]. ATL i.e. Atlas Transformation Language helps overcome the gap between two distinct models. In model engineering all handled items are to be considered as models. The model transformation itself therefore has to be defined as a model and this transformation model has to conform to a transformation metamodel that defines the model transformation semantics. As other metamodels, the transformation metamodel has, in turn, to conform to the considered metamodel. The entire procedure gives you a target model on which further automation depends.

The quality of any web application depends on the GUI, the content, different entity structure wealth and challenges associated with them are navigability, visual presentation etc. This takes time to achieve. Idea has thus emerged to invest in reducing time required for doing repetitive tasks like create, remove, update and display of the different objects. The idea is to display the objects of a given class, basing on information of another object of another class provided that the two classes are connected via associations on the class diagram [17].

This work is very useful for users called to manipulate some information linked to each other in an arborescent structure where the display of information depends on another (display, as part of a web application, employees of a service, of a division, of a direction, etc.).Therefore, we apply MDA approach for generating PSM form UML class diagram. This can take form of an MVC web application. We can start with UML class diagram itself which has information about its attributes and operations. The UML class diagram is the source model. Target model is MVC. The transformation is to be carried out by using ATL rules. Both source and target metamodels are created. Transformation by ATL results in an XMI file which can be further used to generate MVC implementation code. Any application requires some repetitive, non-changing code for implementation. This idea helps for the provision of the same. It can greatly reduce development time.

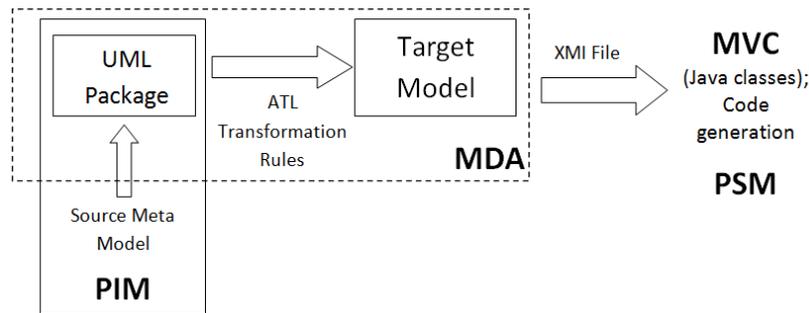# IV.   PROJECT DETAILS

A. Basic Architecture

Fig. 1 Basic Architecture

The figure above is the basic architecture of the proposed system. The idea is to begin with a platform independent model, use ATL and convert it later into a platform specific model thus generating implementation code.

B. Technology
- A conventional web application is studied to determine its functional differences. These different functional models actually depict functional independency of the entire web application.
- Business logic, static data and navigational information individually form MVC modules.
- The goal is to achieve easy code generation of the business logic; entities that are not much modified.
- ATL is used to convert source model to a target model using rules.The XMI file generated further helps develop code of the application.

C. Model-View-Controller Architecture
MVC (Model-View-Controller) is a software design pattern for developing web applications. This pattern consists of three parts namely, Model, View and Controller. Model is at lowest level dealing with business logic or data. View concerns user accessible part. Controller acts as the interface between the above two. This is accomplished as follows:
- Various static views and UI elements shown to the user be identified as 'views'
- Identify the navigational pattern present in views which let the views' transition from one to another as 'controller'
- The final server side code, handling Business Logic will be identified as 'model'

In this way a conventional web application is converted into an MVC patterned web application.

D. Model Driven Approach
Model-driven engineering approach defines web development as set of models and their inter transformations. Three models related to web development are data model, hypertext model and presentation model.
- Data Model: The data model consists of the data objects, their attributes, methods and inters relationships as described by ER diagrams. Data model relates to the 'model' part of MVC i.e., business logic and server side code from conventional web applications becomes data model in MDA.
- Hypertext Model: The hypertext model consists of the pages that comprise the web application, along with user navigation depicted in terms of links. Hypertext model may be considered as a site view which a particular set of users can access. It takes care of various data units and their interrelations that form the actual content. Hypertext model relates to 'controller' part of 'MVC' i.e., navigation pattern and view transition logic from conventional web application becomes Hypertext Model in MDA.
- Presentation Model: Presentation model renders the logical view of the hypertext for a particular platform. All components are thoroughly rendered on the layout by now. Presentation model relates to 'view' part of the MVC i.e., static views and UI elements that are shown to the user becomes Presentation Model in MDA.

E. Atlas Transformation Language
Atlas transformation language is a model transformation language released under the terms of the Eclipse Public License and Eclipse component used inside of the Eclipse Modelling Project (EMP). It is developed by OBEO and INRIA. ECORE is a meta-modelling language that is part of EMF (Eclipse Modeling Framework) and is the result of ETP project efforts (Eclipse Tools Project). EMF is a modeling framework and code generation to support the creation of tools and model driven applications.

V.   PROJECT INPUT AND OUTPUT

A. Input:
- UML meta model
- Test classes for the application
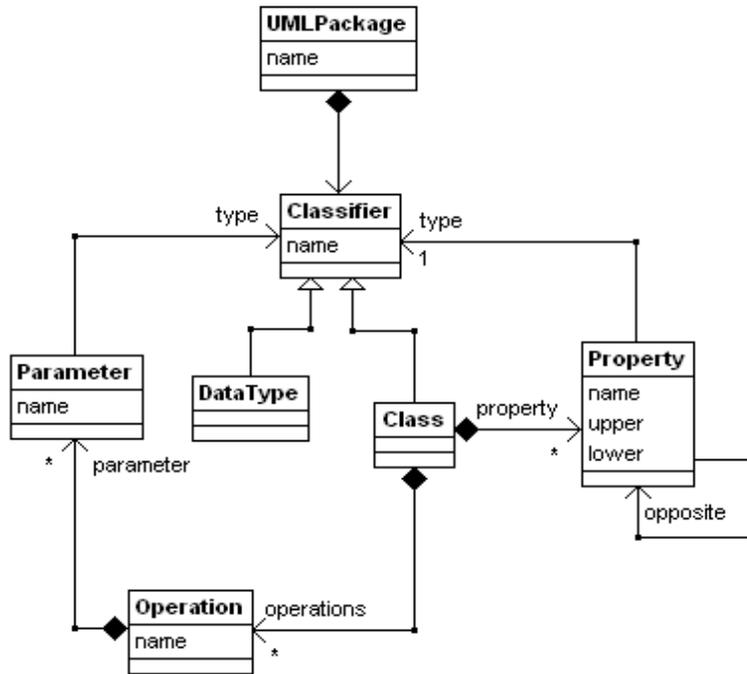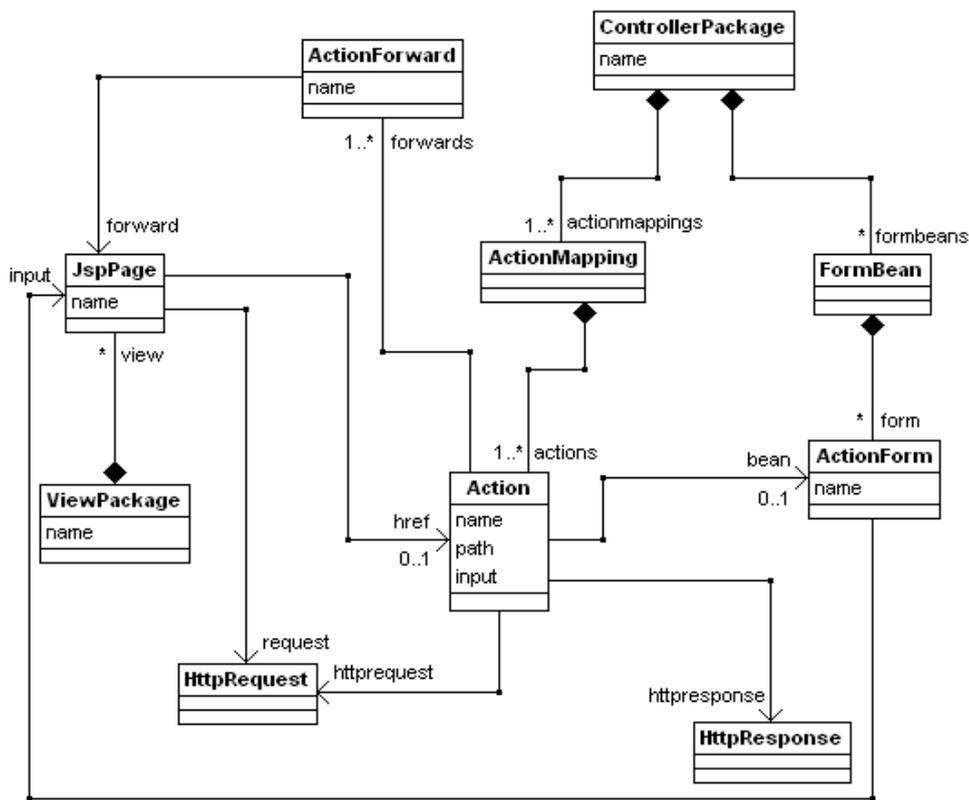- Struts meta model

Fig. 2 UML Metamodel

Fig. 3 Struts Metamodel

B. Output:
- JSP pages
- Form classes
- Action forms
- A sorted web directory is created with above files giving framework for web application.

VI.   IMPLEMENTATION DETAILS

A. Modules
I. ATL module
  ATL uses rules to transform the UML files into XMI files.

Sub-modules are:

- Create Package
    - This module creates package where the web application is to be generated.
- JSP Classes
    - In specified web application means in created package JSP classes are need to be created in order to generate JSP pages.
- Action
    - It consists of files to be used while generating Action classes.
- Mapping
    - It consists of actionMapping xmi files for the mapping of JSP classes and Action forms.

II. Code Generation Module

- By transforming xmi files code is generated.
- Sub-modules are:
- JSP Generation
    - The JSP xmi files get processed and those are converted into JSP classes into specified Web directory or package.
- Action Class Generation
    - The Action xmi files are converted into Action Forms into specified Web directory or package.
- ActionMaping Class Generation
    - The ActionMaping xmi files are used to map the JSP classes and ActionForms by which generated JSP files can get edited as well as get compiled in order to execute using Browser and Apache Tomcat Server.

B. Results

System transforms a class diagram which contains CRUD operations, to a model target, Struts. The result of ATL transformation is an XMI file which is subsequently used as a model to generate the source code of a MVC2 Web application. This tool allows to retrieve, delete, update and create the different objects of the information system.

The emphasis is placed on the associations between classes. The process of transformation which consists of producing the XMI file is realized using ATL. The primary task is to create ECORE models corresponding to our two source and target meta-models as shown in figures 2 and 3 above. In second step, we have implemented the rules of ATL transformation language.

The ATL rules converts XMI files to action forms, java and jsp classes which are then arranged hierarchically into WEB-INF directory.

These classes can then be edited, complied and run in Browser using Apache Tomcat Server. Here a framework of a web application is obtained. Application specific changes in the code can then be made by the developer in the classes generated.

C. Snapshots
JSP Code Generation:

Fig. 4 JSP code generation form used to generate JSP pages where input is an XMI file

Fig. 5 JSP classes.

Displaying XMI file contents:

Fig. 6 This form displays contents of XMI file to generate action classes.
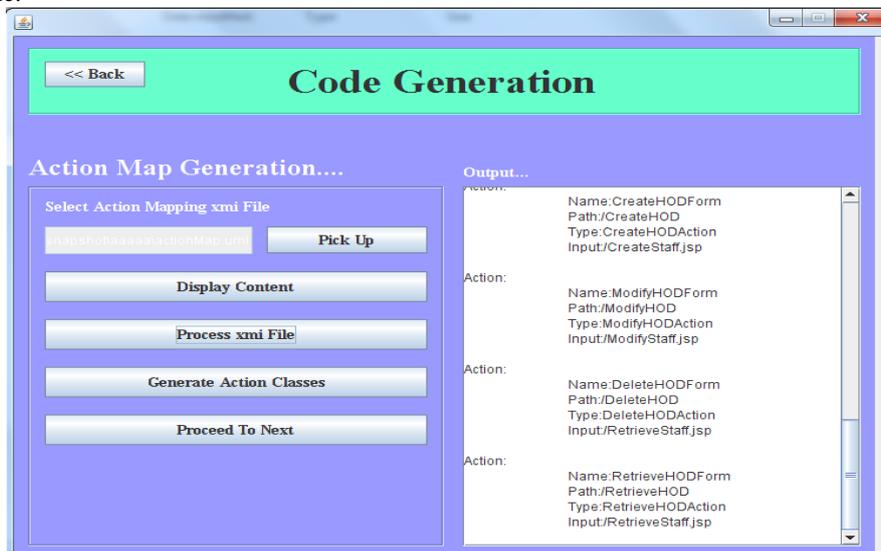
Processing XMI file.

Fig. 7 This processes ActionMapping.xmi file into action forms using ATL transformation rules.
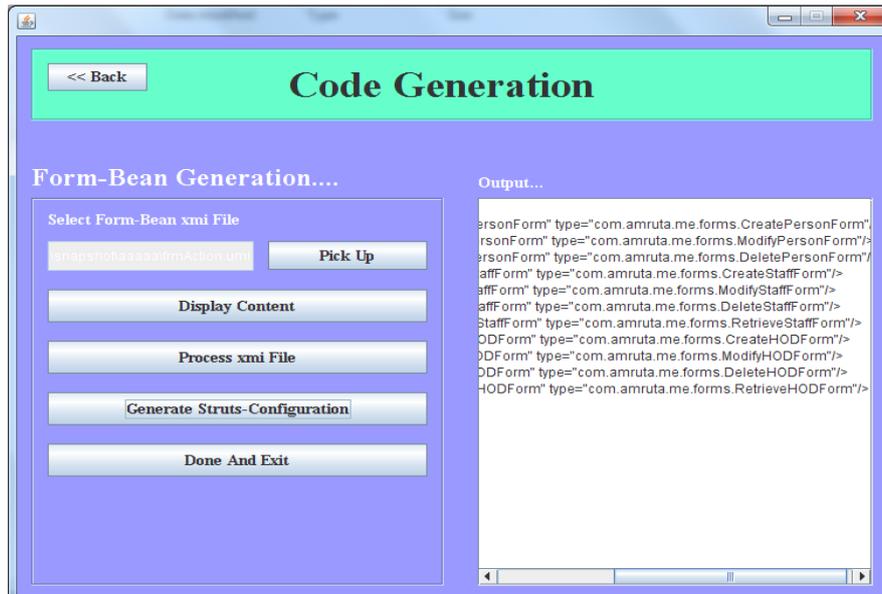
Struts Generation:

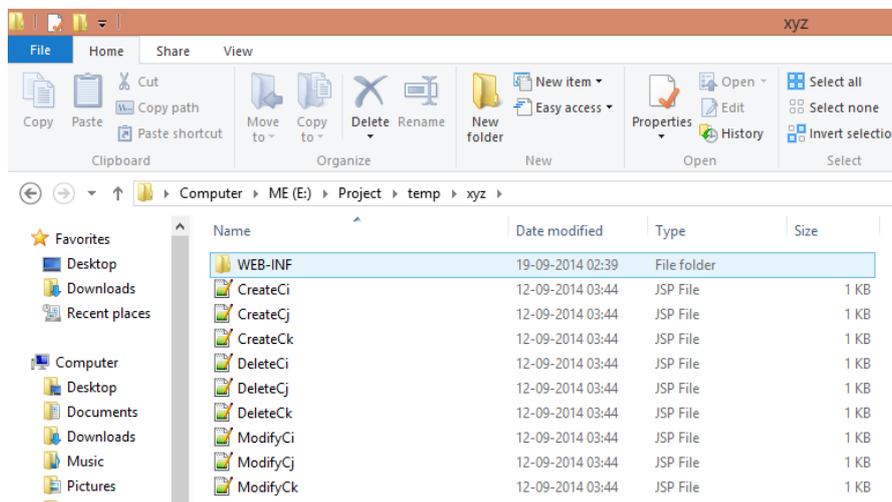Fig. 8 This form finally converts action form classes to struts data.

Fig. 9 Example of hierarchy of classes obtained as a result

VII.   CONCLUSION

With the use of model driven approach and model transformation technology we aim to achieve a more automated process for developing Web applications. We aim to generate ingredients of a web application based on UML class diagrams. These ingredients are to form model, view and controller entities of an MVC web application. The XMI file obtained as a result of ATL transformation can be used to generate necessary partial code of a target web application. The developers of web applications can use such tools and reduce implementation time to a great extent. To achieve this target we first developed source and target metamodels along with source model. This forms the platform independent part of model driven architecture. Using model transformation language like ATL we further transform this PIM to PSM which is platform specific model. Technology then avails us with the notion of generating java classes that are ingredients of a web application. The snippets are like partially ready implementation of an entire application. Thus reducing the task of a developer.

REFERENCES

[1]   Murugesan, S., Deshpande, Y., Hansen, S., And Ginige, A. 2001. Web Engineering: A New Discipline For Development Of Web-Based Systems. In Web Engineering, 3–13.

[2]   Pavla Baierova, Mary Tate, Beverley Hope, The Impact Of Purpose For Web Use On User Preferences For Web Design Features.

[3]   Creating A Model-Driven Web Application Framework Lasseschou S991512 Kgs. Lyngby, Marts 2008 Creating A Model-Driven Web Application Framework.

[4]   Feng Chen*, Hongji Yang, Hong Zhou, Bing Qiao, Web-Based System Evolution In Model Driven Architecture.

[5]   Borchers, J., Deussen, O., Klingert, A., And Knrzer, C. 1996. Layout Rules For Graphical Web Documents. Comput. Graph. 20.

[6] Comai, S., Ttrigueros, M. L., Preciado, J. C., And Sanchez-Figueroa, F. 2007. Designing Rich Internet Applications With Web Engineering Methodologies. In Proceedings Of The WSE. IEEE, Los Alamitos, Ca, 23–30.

[7] Fraternali, P., Comai, S., Bozzon, A., And Carughi, G. T. 2010. Engineering Rich Internet Applications With A Model-Driven Approach. Trans. Web. 4, 2.

[8] Bajwa, I., Siddique, I., And Choudhary, M. 2006. Web Layout Mining (WLM): A New Paradigm For Intelligent Web Layout Design. In Proceedings of the ITI 4th International Conference On Information And Communications Technology (ICICT'06). IEEE, Los Alamitos, CA, 1–2.

[9] Lok, S., Feiner, S., And Ngai, G. 2004. Evaluation Of Visual Balance For Automated Layout. In Proceedings Of The 9th International Conference On Intelligent User Interfaces (IUI'04). 101–108.

[10] Rumbaugh, J., Jacobson, I., And Booch, G. 1999. The Unified Modeling Language Reference Manual. Addison-Wesley.

[11] Damiano Distante, Paola Pedone, Gustavo Rossi And Gerardo Canfora, Model-Driven Development Of Web Applications With UWA, MVC And JSP

[12] Model View Controller [Web Application Component Toolkit]

[13] Diana M. Selfa, Maya Carrillo, Ma. Del Rocío Boone, A Database And Web Application Based On MVC Architecture

[14] ATL: Atlas Transformation Language User Manual, Version 0.2, January 2005, By ATLAS LINA & INRIA, Nantes

[15] M'hamed Rahmouni And Samir Mbarki, MDA-Based ATL Transformation To Generate MVC 2 Web Models

[16] Redouane Esbai, Mohammed Erramdani, Samir Mbarki, Ibtissam Arrassen, Abdelouafi Meziane, And Mimoun Moussaoui, Transformation By Modeling MOF 2.0 QVT: From UML To MVC2 Web Model

[17] Samir Mbarki And, Mohammed Erramdani, Towards Automatic Generation Of MVC2 Web Applications.

[18] Rumbaugh, J., Jacobson, I., and Booch, G. 1999. The Unified Modeling Language Reference Manual. Addison-Wesley.