A New Approach for Web Engineering Based on Model Driven Architecture

Yen-Chieh Huang[1,2*], Chia-Chan Wu[1], Chih-Ping Chu[1]
[1]Dept. of Computer Science and Information Engineering,
National Cheng-Kung University, Taiwan
[2]Dept. of Information Management, Meiho University, Taiwan
*E-mail: x00003199@meiho.edu.tw

Abstract

This paper presents an approach for Web Engineering based on Model-Driven architecture. This approach proposes an UML-based web modeling method and a systematic development process for developing Rich Internet Applications (RIAs). The process begins with a requirements analysis to final implementation and covers the Computation-Independent Model (CIM), Platform-Independent Models (PIM), Platform-Specific Models (PSM) and the transformation rules between each model. This approach defines a UML Profile for modeling this specific domain. This profile extends from Web Application Extensions to modeling the concept of RIAs. In addition, we develop a tool named "WAE2CODE" for generating the code template from PSM. It can parse the class diagrams and sequence diagrams drawn by *MagicDraw*, representing the static and dynamic concept of the system separately. We then translate these diagrams to code templates automatically. As a result of transforming these models within differing levels of abstract concern about a system which the Model-Driven Development desires, the stakeholder can build a complex web application in an efficient way.

Keywords— Model-Driven Architecture, Unified Modeling Language, Web Application Extension, Richer Internet Applications

1. Introduction

Model-Driven Software Development (MDSD) is becoming a widely accepted approach for developing complex applications, especially Object-Oriented Programming (OO-P) which needs to model objects and the collaborations between items. The feature of MDSD is these models address different concerns as the key artifacts in all phases of development. The goal of MDSD is to (semi)automate the process of software development from requirements analysis to code implementation by standard rules, particularly focusing on model transformation.

Web Engineering is also a specific domain to which MDSD could be successfully applied, and is termed Model-Driven Web Engineering (MDWE) [1]. Because the model in MDWE plays an important role in development, it should effectively express the system concept and be easy for developers to use. There are several approaches for modeling web applications. These approaches can generally be divided into two categories, the UML-based model and the specialized languages with tools. The UML-based uses the UML extension mechanism to extend the model for specific domain modeling. However, specialized languages means the Domain Specific Language (DSL) which tend to support higher-level abstractions than general-purpose modeling languages, and are closer to the problem domain than to the implementation domain [2]. These specialized languages are often proposed with its tools, such as WebML/WebRatio [3], OO-H/VisulWade [4], UsiXML/IdealXML [5] and so on.

Despite the DSL-based models allowing one to express in a precise and natural way closer to the problem domain, it cannot fully interoperate with other modeling tools. On the contrary, the benefit of using UML-based model is we can design and draw the models by standard UML-compliant CASE tools. Developers can use their favorite tools by importing or plugging-in the UML profile. In addition, recent versions of some UML CASE tools have already included the Web model, such as *RationalRose* and *MagicDraw*. Due to these web diagrams not having any standard diagram in UML, there are still some discrepancies with others. However, most of these web diagrams use the concept of the UML-based Web Application Extension (WAE) model proposed by Jim Conallen [6].

In this paper, we propose a web model extended from WAE to model the new concept of RIAs. The extended web model focuses on the scripting-based RIAs implemented by web pages with the Ajax framework. The Ajax framework is a Java Script library including the asynchronous request/response functions and desktop-like widgets for enriching the user interface. The extended web model can describe the asynchronous actions and dynamic areas in a web page. Further, it also describes some user interface widgets supported by JQuery UI. Besides, we introduce a MDWE method for developing these kinds of RIAs with our web model. The process starts from requirements analysis to implementation following the Model-Driven process. First, describe the requirements by a case diagram as artifacts of CIM. Second, analyze the domain concept and construct the UML class diagram and sequence diagrams as artifacts of PIM. Third, both the class diagram and sequence diagrams should be transformed to the extended web model as artifacts of PSM. Fourth, the code templates could be generated from the web model using the tool- WAE2CODE we proposed. Finally, developers then finish the system based on these templates. Thus, the efficiency of web system development will be improved by this MDA-compliant approach.

2. Related Work

Model-Driven Architecture (MDA) is an effort approached by the Object Management Group (OMG) using models in system development. It is model-driven because it provides a means of using models to direct the course of understanding, design, construction, deployment, operation, maintenance and modification [7].

2.1 Web Modeling Method Approach

There are several approaches, such as WebML [8], UWE [9], OOHDM [10], WAE, etc, introducing new models and mechanisms to capture the web applications concept and proposing the complete development of a Web solution. WebML describes Web applications at three levels: the content objects, the hypertext model and the look and feel of a user interface. The content objects are specified using a simplified UML class diagram including the relationships and data derivation expressions. The hypertext model has front-end hierarchical organizations in terms of site views, areas, pages, content units and so on. The presentation model is concerned with how WebML represents pages on the screen. UML-based Web Engineering (UWE) is a model-driven development approach for Web applications. UWE follows the principle of ‘separation of concerns’ by modeling the content, the navigation structure, the business processes, and the presentation of a Web application separately [11]. It is an UML-based Language providing a UWE UML profile for extension. In addition, they provide a plugging-in for the *MagicDraw* and *ArgoUML* CASE tool. Another UML-based approach is WAE [6], comprised of the typical Web artifacts (server pages, client pages, forms, framesets, etc.) proposed by Jim Conallen. The main advantage of WAE is it allows an efficient reflection of the design artifacts of a Web structure, as well as implementation from these design artifacts. As mentioned earlier, WAE’s concept was already used in some UML CASE tools for web solutions, such as *RationalRose, MagicDraw, Enterprise Architect*, etc. Another related work is presented at [12]. This work describes a method to generate a user interface

code using the Model View Controller (MVC) pattern. The method uses robustness analysis [13] to reduce the gap between the analysis and design phase of a web system. Then the boundary objects resulting from robustness analysis are transformed into JSP pages. In contrast to our work, our code generation not only focuses on the interface but also includes the control object and entity object code generation. In addition, we provide a tool for automatic code generation.

2.2 Rich Internet Applications

RIAs are complex web applications based on thick client architecture, asynchronous communication, and a great variety of user interface widgets [6]. The definition of RIAs does not refer to any specific technology for implementing Rich Web applications. In [6], classifies some categories of RIAs implementations. We focus on the scripting-based RIAs implemented with client-server page architecture using asynchronous actions composed by the richer user interface widgets provided by the Ajax framework. The research in [14] presented a pattern approach for the model-based engineering of RIAs. This method extended the presentation model of UWE with some RIAs patterns, such as *drag & drop, live validation, auto completion, periodic refresh* and so on.

Due to the increasing complexity of Web applications, current Web Engineering methods cannot provide web solutions any more. Thus, there are several approaches extending from the existing method that have appeared recently. A WebML extension for RIAs was proposed in [1] with new units for the client side operation of a web application. An OOH extension, the so called OOH4RIA [15], introduced new model elements and applied new transformations to them. This approach is implemented by the Google Web Toolkit (GWT) framework. In contrast to our work, we use JQuery Java Script framework to implement RIAs and extend the WAE for web modeling because of the simplicity of expression of Web architecture and the widely used JQuery Library.

3. Model Driven Development Process

The development process of RIAs is based on the MDA concept. That is, it decouples the system concept by creating distinct models of a system at different levels of abstraction. Then, model transformation is performed during the development life cycle with regular transformation rules (or patterns). The models not only help describe the system concept at different phases, but also contribute to automatic code generation. To comply with MDA, the models we used are divided into three categories, which were computation independent, platform independent, and platform specific. There are tools for constructing each model. The MDA-compliant process is shown in Figure 1, and the system development step is as follow:

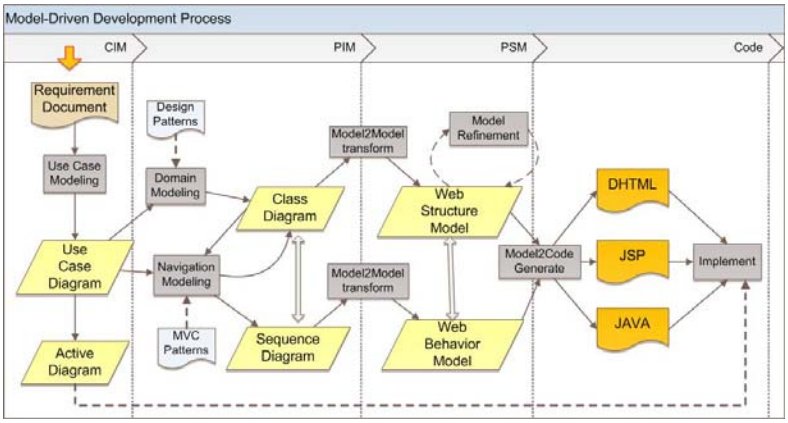

Figure 1. Model-Driven development process overview

1) ***Step-1 Use Case Modeling***

Use cases are descriptions of the functionality of the system from the users' perspective. Use case diagrams are used to show the functionality the system will provide and to show which users will communicate with the system in some way to use that functionality. Use case descriptions provide a description of the interaction between the users of the system, actors, and the high-level functions within the system [16].

2) ***Step-2 Domain Modeling***

Domain analysis is the identification, analysis, and specification of common requirements from a specific application domain [17]. The domain model of a system is used to indicate the main entity of the problem domain has and to indicate the relations between them. Class diagrams are presented for the domain modeling. These concepts are represented by the classes, attributes and relationships between classes. The attributes and operations can be found by drawing and use case descriptions. In addition, the permanent classes in the class diagram can be mapped into data tables in a database [18]. By the way, for some common problem domains, the design patterns will be useful for modeling.

3) ***Step-3 Behavior Modeling***

The behavior model of a system is used to indicate the object workflow corresponding to the human computer interaction described in the use case. Besides, each use case should correspond to one behavior model for describing the object behavior.

At this step, analysis can be performed with Robustness analysis using the MVC pattern that divides the elements into three categories: boundary, control, and entity.

The boundary element is responsible for user interface rendering and human computer interaction handling. There is at least one boundary element for a use case. The entity element is the static object used in the use case constructed at the domain modeling step for representing the data or logic functionality. The control element is responsible for handling the interaction between the boundary element and entity element, for example, sending messages or boundary element redirecting control.

There is exactly one control element for a use case to handle the scenario described in it. After analyzing the use case by robustness, the boundary and control object will be generated. The sequence diagram is then constructed to simultaneously describe the object's workflow. In addition, the generated objects should be added into the class diagram constructed from domain modeling.

4) ***Step-4 Model Transformation (PIM to PSM)***

The model transformation from PIM into PSM takes place after the PIM is constructed. In PIM, the models include the class diagrams and sequence diagrams for describing different concepts about the system. Both models should be transformed into a PSM skeleton by transforming the mapping rules, and then completed with platform features by the PSM designer. That is, the system designer should consider the platform features and technology to implement.

4.1) Step - 4.1 Web Structure Modeling

The web structure model is the summarized model merging all the concepts and including the structure of the entire system (boundary, control, and entity object), navigation, and presentation of the client pages. The summarized model uses the WAE extension model, mentioned earlier in this study. Nevertheless, this study enhanced the layout between the interface elements in the web architecture model. Therefore, UI designers can focus on the layout style in the implementation phase. This concept is the same as the Abstract User Interface (AUI) described in [8].

4.2) Step-4.2 Behavior Modeling

The web behavior model plays an important role to assist in presenting the information of system navigation and objects workflow in every use case. Although the web structure model can express partial information about navigation such as linkage between pages and as the same submit target of each form, the page redirections decided by the branch condition cannot be addressed in it. Therefore, this study used the transformed web behavior model from the PIM sequence diagram to represent dynamic behavior. Further, the operation usages from the server page to Java beans are also expressed in the web behavior model which is extended from the sequence diagram.

5) ***Step-5 Model Refinement***

After constructing the web structure model and web behavior model in the PSM phase, model refinement could make these diagrams more accurate for best practice and more object reuse by integrating objects having similar functions. On the client side, we can summarize different pages and the widgets on the page by frameset. On the server side, we can summarize the same functions of operations to become one class for reuse.

6) ***Step-6 Code Generation***

In this study, the application platforms are the JAVA, JSP, and Tomcat server. The front-side uses dynamic web pages and AJAX to depict the RIAs. The backend uses Java Bean to process the logical functions and communicate with the database. Code generation is performed according to the summarized Web Structure Model and several Web Behavior Models using the tool named *WAE2CODE*. The code template will generate HTML, JSP, Java, and *web.xml* files to reflect the classes of Web Structure Model.

7) ***Step-7 Implementation***

In the final phase of the development process, programmers must add/modify code from code templates. Programmers must complete the functionality of every operation in each class. For instance, while editing the JSP code template, they must add the XML data constructing code segment, which builds an asynchronous XML data logical process from the server side to the client side. Further, the client side code template must add the Java script code to receive the XML data and represent the data by dynamic HTML. As mentioned earlier, the generation code templates don't include the user interface layout style. Thus, the user interface designer must adjust the widget position, size, color, and style to become the final user interface. When finishing all the functions of the operations, XML data processes and widgets of the user interfaces in code templates, the RIA system is finally complete.

4. Case Study

The practicality of the proposed methodology is demonstrated through a case study. The running example throughout our work is an online bookstore system. As shown in Figure 2 is the use case diagram, a member can query the product, add/delete items to/from the shopping cart, or pay for an order. Unlike a traditional online shopping website, the demo one used a lot of Ajax centered technique and richer interface widgets to achieve the features of RIAs. We used the *MagicDraw UML* CASE tool to construct all the models used in our development process by importing our extended UML profile during PSM modeling step.

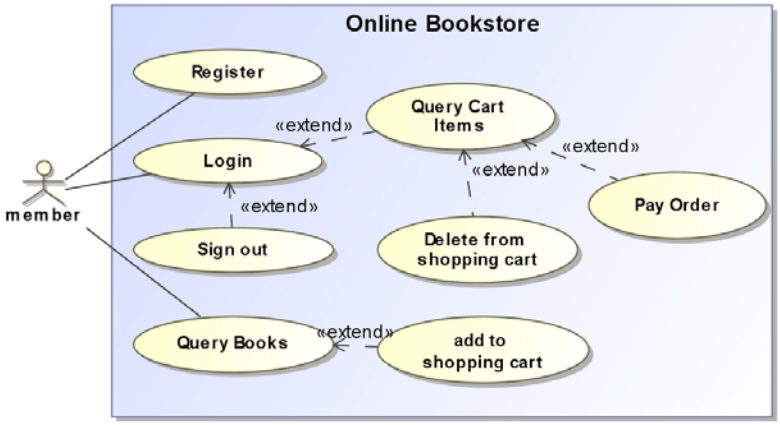

Figure 2. Use case diagram of the online bookstore

Code Generation- *WAE2CODE tool*

The interface of our transformation tool is shown in Figure 3. The *File Location* field indicates the location of the PSM project file constructed by *MagicDraw*. The *Model* field indicates the PSM package name in the *MagicDraw* project. The *Diagram* field indicates the name of the summarized Web Structure Model in the PSM package. The WAE2CODE can parse and transform the information of the Web Structure Model (class diagram) and these Web Behavior Models (sequence diagrams) under the PSM package. It can then generate the source code template at the location indicated in the *Output Location* field.

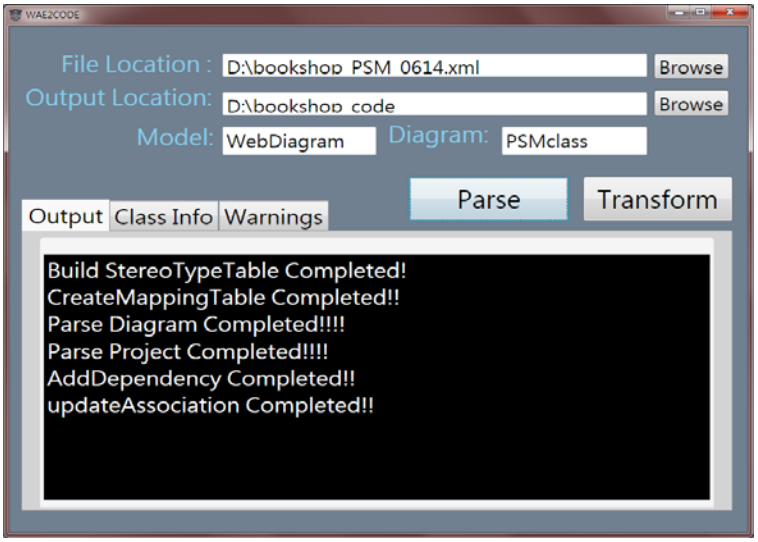

Figure 3. WAE2CODE tool

In the final step, the code template generated from the WAE2CODE should be finished by filling the remaining code segments, and the look and feel of the layout of the interface should be adjusted simultaneously. Figure 4 is the completed user interface corresponding to the client page and the presentation widget. The search bar has the auto complete feature corresponded to the *searchKeywords*

property whose stereotype is marked as an *autoComoplete* type in the *searchBar* class. Further, each book item in the *ProductContainer* was dynamically built by the Java script function named *showBookIndex* modeled by the *dynamic* dependency relationship. Besides, it could be dragged into the floating Dialog named *My Shopping Cart* as the action of adding a product to the shopping cart. The *drag & drop* feature corresponded to the *isDraggable* and *isDroppable* tagged values of the area element.

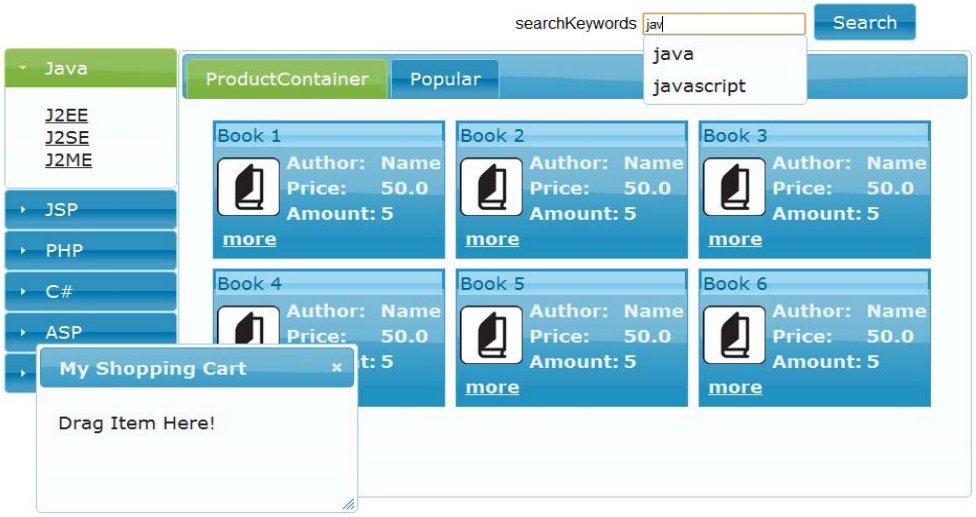

Figure 4. The completed user interface

5. Conclusion and future work

This paper presented a methodology for developing RIAs that complied with the model-driven feature. It included four major development phases and seven steps specifying the requirements, constructing the PIM with robustness analysis, and then transforming it to PSM using the web model extended from WAE. Besides, the code template of the summarized system could be generated by the tool named WAE2CODE to decrease the coding time and to increase the system development efficiency.

This study used sequence diagrams for modeling the behavior of these web objects where the class diagram is weakly here. For example, the redirection condition in a server page object and the relationship of the operation call from the control object to the Java Bean object. The sequence diagrams not only help simplify the collaboration expression between objects, but also contribute to the control object code generation. Further, each generated code was accompanied by a comment to describe the reference information, such as the message or alternative fragment of certain sequence diagrams. As a result, it increases the consistency between models and code, leading to a reduced gap between design and implementation.

The RIAs frameworks such as JQueryUI, which we applied to increase the richness of the design for the client, are continuously released with new versions. Thus, we should follow its steps to capture the new concepts in our extended web model for enhancing the flexibility of use.

6. References

[1] N. Moreno, J. R. Romero, and A. Vallecillo, “An Overview of Model-Driven Web Engineering and the MDA”, Web Engineering: Modeling and Implementing Web Applications, 2008.

[2] N. Moreno, P. Fraternali, and A. Vallecillo, “Web Modeling in UML”, The Institution of Engineering and Technology, 2007.
[3] S. Ceri, P. Fraternali,A. Bongio, M. Brambilla, S. Comai, and M. Matera, “Designing Data-Intensive Web Applications”, Morgan Kaufmann, 2002.
[4] I. Garrigós, J. Gómez, and C. Cachero, “Modelling Dynamic Personalization in Web Applications”, Proc. of the 3rd International Conference on Web Engineering. Oviedo, Spain, 2003.
[5] J. Vanderdonckt, “A MDA-Compliant Environment for Developing User Interfaces of Information Systems”, Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005.
[6] J. Conallen, Building Web Applications with UML. Addison Wesley, 2003.
[7] Object Management Group, **“**MDA Guide V1.0.1**”**, http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf, 2003.
[8] S. Ceri, P. Fraternali, and P. Bongio, “Web Modelling Language (WebML): A Modelling Language for Designing Web Sites,” Proc. Ninth Int’l World Wide Web Conf., Computer Networks, vol. 33, no. 1-6, pp. 137-157, 2000.
[9] N. Koch, A. Knapp, G. Zhang, and H. Baumeister, “UML Based Web Engineering: An Approach Based on Standards. In Web Engineering: Modelling and Implementing Web Applications”, HCI Series, vol. 12, chapter 7, pp 157-191, Springer-Verlag , 2007.
[10] G. Rossi, “An Object-Oriented Method for Designing Hypermedia Applications,” PhD dissertation, Univ. of PUC-Rio, 1996.
[11] J.C. Preciado, M. Linaje, R. Morales-Chaparro, F. Sanchez-Figueroa, C. Gefei Zhang Kroiss, and N. Koch, “Designing Rich Internet Applications Combining UWE and RUX-Method”, 8th International Conference on Web Engineering, 2008.
[12] J. H. Wu, S. S. Shin, J. L. Chien, W. S. Chao, and M. C. Hsieh, “An Extended MDA Method for User Interface Modeling and Transformation”, The 15th European Conference on Information Systems. pp 1632–1641, 2007.
[13] D. Rosenberg, and K. Scott, **“**Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example”, Addison Wesley, 2001.
[14] N. Koch, M. Pigerl, G. Zhang, and T. Morozova, Patterns for the Model-Based Development of RIAs, ICWE 2009.
[15] S. Meliá, J. Gómez, S. Pérez, O. Díaz, “A Model-Driven Development for GWT-based RichInternet Applications with OOH4RIA”, Proc. of ICWE, 2008.
[16] S. Bennett, S. Mcrobb, and R.Farmer, “Object-Oriented Systems Analysis and design Using UML”, 3rd edition, McGraw-Hill, 2006.
[17] R.S. Pressman, “Software Engineering – A Practitioner’s Approach”, 7th edition, McGraw-Hill, 2010.
[18] Y.C. Huang, “Transformations from Class Diagram to Relational Table and Application Template”, Master Thesis, National Sun Yat-Sen University, 2004.