An Effective and User-Friendly IDE Tool to Facilitate Conceptual Design and Maintenance of Web Applications

Vincent Tam
School of Computing
The National University of Singapore
Lower Kent Ridge Rd, Singapore 119260
+65-8744463
vtam@comp.nus.edu.sg

C.S. Yeo
School of Computing
The National University of Singapore
Lower Kent Ridge Rd, Singapore 119260
+65-8746838
yeochees@comp.nus.edu.sg

ABSTRACT

Web applications are growing in demand, complexity and size, thus making it difficult to systematically design and maintain general web applications. To aid in fulfilling these difficult tasks, we present an effective IDE tool to allow the design of web applications at a conceptual level based on a web modeling approach – the WebML with its modeling specifications written in the platform-independent XML. In addition, we proposed a library of templates for different components in our IDE tool to facilitate reuses and maintenance. Besides, we studied how structural design of web applications could possibly be improved with easy-to-use features in our IDE prototype. Clearly, this ongoing work opens up many exciting directions for future investigation.

Keywords

Web Design Methodologies, IDE Tool, Component-Based Approach, Markup Language.

1. INTRODUCTION

Nowadays, there are tremendous demands for Web applications to deliver highly complicated, dynamic and interactive services to users, thus making the Web design and maintenance processes extremely difficult to manage. On the other hand, most popular Web technologies such as the Microsoft Active Server Pages (ASP), Java Server Pages (JSP) and Allaire's Coldfusion all aimed at providing frequently needed functionalities (e.g. database queries, displaying dynamic information in a certain layout specified by user) to cut down the implementation time. However, as in many other large-scale software projects, the main challenge often lies in capturing all the requirements to form a concise and precise design for the subsequent implementation, testing and maintenance phrases.

There were different Web modeling approaches proposed. Due to the close linkage of Web applications with hypermedia, earlier Web modeling approaches were based on hypermedia concepts. Later, it was found that modern software engineering approaches and models could readily be deployed to model most complex and dynamic Web applications. For instance, Conallen [1] proposed an extension of UML for Web application design. A Web application is simply considered to be a special type of a client/server application for which Conallen proposed a new set of UML classes and association stereotypes to model Web applications. Accordingly, a Web page is modeled to contain logical components that are executed either on the client (Java applets, ActiveX controls, etc) or the server side (database objects, server pages). However, Conallen's proposal tends to focus on the lower-level implementation detail and does not capture the high-level conceptual design. Besides, the WebComposition model [2] is built based on the object-oriented (OO) conceptual model to represent Web entities of any granularity such as a CGI script, a web page, and so on as *component objects*. Unfortunately, the WebComposition model does not describe the run-time operation of any Web application. Lastly, the Araneus project [4] models a web application by adopting a database-centric process. It consists of two conceptual models, a data model called Araneus Data Model (ADM) and a web navigation model called Navigation Conceptual Model (NCM). However, Araneus does not address the issue of presentation design.

In this paper, we focus on developing an IDE tool to support and possibly improve on one of the most complete Web modeling approaches so as to alleviate the complex tasks of Web design and maintenance. Based on the WebML [3] modeling approach, we implemented a prototype of the IDE tool to support the design of most Web components described in the framework. Furthermore, we highlighted two possible areas for further improvement to encourage reuse of *standard* components and systematic process for structural design. Clearly, this ongoing research project opens up many interesting directions for further investigation.

2. WEBML – A COMPONENT-BASED DESIGN APPROACH

The WebML [3] is basically a component-based, high-level and platform-independent design approach with its specifications associated with a graphical representation and the extended Markup Language (XML) syntax. As a markup specification language to design ways of describing information, XML is chosen since it has become the W3C recommendation and will very likely dominate most Web applications in the future. The specifications of WebML cover four different views: the involved data content (structural model), the Web pages that compose the Web application (composition model), the links between the pages (navigation model), the layout for presentation of pages (presentation model) and the customization for one-to-one delivery (personalization model). A tool suite called ToriiSoft to implement WebML has been

planned yet delayed for delivery. In this research project, our main interest is to learn from our experience to improve the WebML approach rather than purely implementing WebML. Nevertheless, for our subsequent discussion, the 4 models supported in WebML are given as follows.

2.1 Structural Model

The structural model expresses the content of the application, in terms of the entities and relationships. Entities have attributes, with an associated data type. Properties with multiple occurrences are described using components. To certain extent, it is quite similar to the entity-relationship (E-R) model widely used in database design. Cardinality constraints are also given. For detail, refer to [3].

2.2 Composition Model

The composition model states what are the pages in the Web applications and how what content units form each page. There are six different types of units, namely the Data, Multidata, Index, Scroller, Filter and Direct units. The roles of these units should be obvious from their names. Due to space limit, readers should refer to [3] for a detailed description of all these units.

2.3 Navigation Model

The navigation model illustrates the navigation links between pages and content units to form the hypertext. There are two types of links. The first type is a non-contextual link that connects up independent pages. The second type is a contextual link that will display the content of the destination unit based on the content of the source unit. Contextual links are based on the relationships associated between the entities specified in the structural model.

2.4 Presentation Model

The presentation model identifies how a page (as defined in the composition model) will appear to the end-user. A style sheet specifies how the pages are layout and the content elements to be inserted into each layout. There are two types of presentation stated in the paper. The first type is page-specific presentation, which dictates the presentation for a specific page only. The second type is generic presentation, which applies to all generic pages.

2.5 Personalization Model

The personalization model allows the definition of content or presentation based on user-specific profile. Two methods of personalization are described. The first method is called declarative personalization, which defines the personalization based on user-specific data predefined in two entities called User and Group. The second method is called procedural personalization, which allows specification of business rules. The business rules state what event is to be monitored, the pre-condition to check when the event occurs and finally the action to be taken if the pre-condition is satisfied.

3. OUR PROPOSED MODIFICATIONS

We built the prototype implementation of an IDE tool to support the WebML design process using the Microsoft Active Server Pages (ASP) approach. In addition, we proposed the following modifications to possibly improve the original WebML approach at least in the following two aspects.

3.1 Structural Analysis

The structural model defined in WebML appears to be the most critical and important stage in modeling a web application. This is due to the fact that the structural model reflects all the data content of the web application. The hypertext model (consisting of composition and navigation models) could be easily created from the structural model. In WebML, there is no specification of a procedure on how to derive the structural model. It is up to the designer's competence to design the structural model as complete and accurate as possible. This can pose a major problem if the designer is does not have a good knowledge and experience. Therefore, we proposed the following 3-step structural analysis approach to assist the designer in building a more complete structural model as follows.

1) Identify all the users that will be using a certain part of the web application.
2) Identify all the elements of information that the users are interested in.
3) Identify relationships for each element. These relationships correspond to that of the relationships between entities.

3.2 Reuse

Reuse is an important issue in hypermedia development. This is because reuse can increase productivity, software reliability and maintainability. Accordingly, we suggested to provide a reuse repository that store templates and models that could be reused in WebML. It should be noted that only some generic models for certain application domains are chosen to support reuse since there are some models that are too specific to be reused.

4. ONGOING WORK

The following are the interesting directions we are investigating at the moment:

- Extending our current IDE tool with more easy-to-use features, and apply it to model different applications;
- Investigating to build more high-level design on top of the existing WebML approach to support more complicated or large-scale Web applications.

5. ACKNOWLEDGMENTS

We are grateful to Prof. Kozo Sugiyama and A/P C.H. Chi for their fruitful discussion and useful pointers.

6. REFERENCES

[1] J. Conallen, *Modeling Web Application Architectures with UML*. Communications of the ACM, Oct 1999, Vol. 42, No.10, pg. 63-70..

[2] H. Gellersen, R. Wicke, M. Gaedke, *WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle*. Proc. WWW6, 1997.

[3] S. Ceri, P. Fraternali, A. Bongio, *Web Modeling Language (WebML): a modeling language for designing Web sites*. Proc. WWW9, 2000.