Rapid Development of Modular Dynamic Web Sites Using UML

Tim Schattkowsky¹, Marc Lohmann²

¹ Paderborn University, C-LAB, D-33102 Paderborn, Germany
tim@c-lab.de
² Paderborn University, Department of Computer Science, D-33095 Paderborn, Germany
macke@uni-paderborn.de

Abstract. Development of dynamic Web sites is often performed by teams consisting of graphic designers and software developers. Communication between these different team members has to be supported with a simple modeling approach that considers their different academical backgrounds. Dynamic Web sites can contain multiple modules that may reappear on different pages. Reuse of both business logic and visual design modules would be desirable. Furthermore, a considerable amount of time is usually consumed by the implementation of data flows that are already defined in the model. Rapid development is enabled by providing roundtrip engineering capabilities with support for automatic code generation. We propose a simple subset of the UML adapted to the problem domain by means of stereotypes as well as a strategy for generating code templates from such models. These templates are tailored to the tasks of each team member. This enables parallel work and automated reintegration of results.

Keywords: Unified Modeling Language, Hypermedia, WWW, Object-Oriented Design, Code Generation

1 Introduction

There is a growing demand for lightweight methods for the development of small- and medium scale Web applications. Short production time and heavy cost pressure from customers and competitors are major economic factors in the development process of these systems. Heavyweight development processes would introduce expensive overhead into such projects, which cannot be economically compensated by the process’s benefits.

The development teams for such Web applications consist of people with different skills and academic backgrounds [10]. These people use different languages to communicate their ideas which leads to misunderstandings. This may cause expensive redesigns. In order to support successful communication, a common language is required, which is understandable to people of different background. The Unified Modeling Language (UML) [3,11] fulfills this requirement by providing a family of
intuitive diagrammatic notations by which a software system can be described at a high level of abstraction.

The different views and backgrounds of the team members have to be supported by the models. These models have to be focused on the information relevant to the different roles in the development team. In many cases, we can distinguish between the domain expert, who has knowledge about the business processes behind the Web application, the graphic designer, who is in charge of the creation of the user interface, usually consisting of HTML pages, and the developer, who has to build a working software system based on the work of his partners. Furthermore, we have the customer, who has little knowledge about the technical realization of the project. As a result, another requirement for a Web application engineering methodology is the support for the views and backgrounds of different roles in development teams.

On the more technical side, many Web applications are based on dynamic Web pages generated by scripts written in a language like PHP [1], Microsoft’s Active Server Pages (ASP) [9] or Java Server Pages (JSP) [12]. Thus, these Web applications are in fact dynamic Web sites. This introduces a problem into the development process. The responsibilities for the script and the design of the generated pages are split between developer and graphic designer. Furthermore, these scripting languages usually lead to a mixture of business logic and design elements within one software module. This has to be clearly separated into separate modules to allow the reuse of both logic and design modules. The method we propose enforces modularization of Web applications. This enables the team members to work concurrently on different modules of a Web page and to consistently integrate their results.

Many small and medium scale Web applications only have small business logic. For such systems, the realization of data input, transport and presentation consumes a considerable amount of production time. A modeling approach for dynamic Web sites should support automatic code generation and thus safe production time. The models should also yield a complete structural description of the Web site. This enables the generation of a working incomplete prototype without design elements and business logic and enables rapid prototyping. Such a prototype allows early demonstrations, which are very important for customer communication.

We propose a lightweight UML-based approach to modeling small and medium sized Web applications aiming at the above requirements. The approach employs specific use case diagrams, activity diagrams and class diagrams in order to capture the fundamental structural and behavioral aspects of a Web application while providing different views on these aspects according to the needs of different team members. Using the running example of an online reservation system, we demonstrate the derivation of code templates in HTML and PHP from the models.

2 Related Work

In this section, we briefly discuss different approaches to the modeling of Web applications. Some focus on the modeling notation, like Conallen who proposes an extension to the UML. Other focuses on the development process, for example [2, 7], WebML or WSDM.