

Model-Driven Web Development with VisualWADE¹

Jaime Gómez

Web Engineering Group. DLSI. Universidad de Alicante 03690 Alicante SPAIN
jgomez@dlsi.ua.es

Abstract. VisualWADE² is a model-driven web development environment for web applications that includes model-based code generation techniques. This paper describes the capabilities of that environment and how Web applications can be generated in an automated way.

1 Description

The focus of this work is the presentation of the tool VisualWADE, a CASE (Computer-Aided Software Engineering) tool to develop Web applications. VisualWADE provides an operational environment that supports the methodological aspects of the OO-H Web design method [1]. The goal of this tool is to simplify the design and implementation of Web-based information [4][5]. This is addressed from an object-oriented perspective by means of enhanced conceptual models. With VisualWADE, a Web application is modeled by three major design dimensions; Structure describes the organization of the information managed by the application in terms of the pieces of content that constitute its information domain and their semantic relationships. Navigation is concerned with the facilities for accessing information and for moving across the application content. Finally, Presentation affects the way in which the application content and navigation commands are presented to the user.

At conceptual level, applications are described by means of high level primitives which specify the structural, navigational and presentational views making an abstraction from any implementation issue. Those primitives are expressed with the UML [3] notation. Structural modeling primitives describe the types of objects that constitute the information domain and their semantic relationships. An standard UML class diagram is used to model that structural view. Derivations can also be expressed in that structural modeling to enrich the system specification. VisualWADE provides an extended OCL-like language [2] to specify formulas for derived attributes. This language can be viewed as an action language [3] for Web applications and provides a powerful mechanism to support code-generation capabilities in the environment.

Navigation modeling primitives express the access paths to objects of the information domain and the available inter- and intra-object navigation facilities, again without committing to any specific technique for implementing access and navigation. The designer specifies a navigation view of the system by means of a UML profile. This profile has been defined on the top of the main navigation primitives provided

¹ This paper has been partially supported by the Spain Ministry of Science and Technology, project number TIC2001-3530-C02-02.

² VisualWADE can be reached at <http://www.visualwade.com>

by the OO-H method and following the UML 2.0 specification (these primitives are collections, navigational classes, navigational targets and a taxonomy of links). The OCL-like language is also used in navigation models to specify preconditions (navigation constraints in origin) and filters (navigation constraints in destination). A set of well-known visualization patterns can be applied over navigational models to refine the way the user navigates through the system. Some of them are indexes, guided tour, indexed guided tour and show all. The designer can also specify the activation of services in the navigation modeling. These services are a set of predefined methods provided by the environment. They correspond to the classical CRUD operations used in the development of data-intensive applications. Other works in this area like WebRatio [4] have shown how a wide variety of Web applications can be modeled with this set of predefined operations.

Presentation modeling aims at representing the visual elements of the application interfaces in a way that abstracts from the particular language and device used in the delivery. In VisualWADE, starting from a navigation model, a default presentation that fullfills that specified navigation can be obtained applying model transformation rules. As a result a set of abstract pages (expressed in XML) are generated. The independent specification of presentation, separate from structure and navigation, is particularly relevant in Web context, because the final rendering of the interface depends on the browser and display device and thus there may be the necessity to map the same abstract presentation scheme to different designs and implementations. This is the reason why the VisualWADE presentation model includes web-enabled hypermedia authoring capabilities. In fact, this model can be viewed as a kind of authoring tool (i.e. Frontpage or Dreamweaver) were the designer renders the final look and feel of the application using WYSIWYG techniques. Also, animation (prototyping) capabilities are provided in that model. In this way, the designer can navigate through the links and abstract pages generated to test the navigation model. When a presentation model has been refined a model compiler generates the final target in the desired implementation environment. Three deliverables are generated. These are the dynamic pages that constitute the Web user interface, the middleware navigation pages that constitute the navigation engine and the database script. Currently, we provide well-tested model compilers for php/mysql, php/Oracle, php/SQLserver technologies.

References

1. J. Gómez, C. Cachero. OO-H method: Extending UML to Model Web Interfaces. Information Modeling for Internet Applications. Pages. 50-72. Idea Group. 2002.
2. J. Warmer, A. Kleppe. The Object Constraint Language: Getting Your Models Ready for MDA. Addison-Wesley, 2003.
3. S. Mellor, M. Balcer. Executable UML: a Foundation for Model-Driven Architecture. Addison-Wesley, 2003.
4. WebRatio. <http://www.webratio.com/>
5. A. Knapp, N. Koch, F. Moser and G. Zhang. ArgoUWE: A CASE Tool for Web Applications. In EMSISE03 held in conjunction with OOIS03, 14 pages, September 2003