Chapter VIII

OO-H Method:
Extending UML to Model Web Interfaces

Jaime Gómez
The Web Engineering Research Group, University of Alicante, Spain

Cristina Cachero
The Web Engineering Research Group, University of Alicante, Spain

ABSTRACT

The mostly “creative” authoring process used to develop many Web applications during the last years has already proven unsuccessful to tackle, with its increasing complexity, both in terms of user and technical requirements. This fact has nurtured a mushrooming of proposals, most based on conceptual models, that aim at facilitating the development, maintenance and assessment of Web applications, thus improving the reliability of the Web development process.

In this chapter, we will show how traditional software engineering approaches can be extended to deal with the Web idiosyncrasy, taking advantage of proven successful notation and techniques for common tasks, while adding models and constructs needed to capture the nuances of the Web environment. In this context, our proposal, the Object-Oriented Hypermedia (OO-H) Method, developed at University of Alicante, provides a set of new views that extend UML to provide a Web interface model. A code generation process is able to, departing from such diagrams...
and their associated tagged values, generate a Web interface capable of connecting to underlying business modules.

INTRODUCTION

In the last few years, we have witnessed how existing and to come Web technologies have induced much more flexible distributed environments where new business opportunities have appeared but also new risks related to software development (Murugesan et al., 1999). Although the scientific community agrees that in order to keep the possibility of failure to a minimum, the development process for enterprise applications should evolve in a Web engineering manner, there is no agreement at how the core activities behind a sound Web application development process should be addressed. Some approaches, most coming from the hypermedia community, consider Web applications as information delivery systems, where only trivial functionality is offered (Mecca, Merialdo, & Atzeni, 1999; Schwabe & Almeida, 1999; Ceri et al., 2000). Others, mainly coming from the traditional Software Engineering field, regard Web applications as traditional distributed applications, and propose modeling approaches that make exclusive use of standard models and notation to capture the idiosyncrasy of this new platform (Conallen, 1999). Still other approaches consider the Web as a dynamic business Web (Gartner Group, Inc., 2001), where the application development consists of a process of communication and integration of Web services disseminated over the net and offered via (often) collaborating technologies such as UDDI (Universal Description, Discovery and Integration, 2001), DSML (Directory Services Markup Language, 2001), SOAP (Simple Object Access Protocol, 2001) or WSDL (Web Services Description Language, 2001).

We agree with Manola (1999), in that each of these trends partially addresses the nuances Web applications involve, and so a fusion of their respective points of view is needed in order to provide a cohesive solution to the Web application modeling process. On one hand, hypermedia modeling methods contribute a deep reflection on Web navigation and interaction issues, which are basic for a Web application to succeed. However, these sound navigation features are not enough, as Web applications must also provide the user with the (often complex) functionality they need, far beyond navigation. The modeling of functional requirements has already been partially addressed in a number of Advanced Software Production Environments, many based on the UML (UML Specification. V1.3., 1999) notation, that use Model Based Code Generation techniques (Bell, 1998) and have the support of “traditional” software engineering methods. The problem is that those traditional models,
Creative Information Systems
www.igi-global.com/chapter/creative-information-systems/16844?camid=4v1a