

Developing Web Applications from Conceptual Models^{*}

Vicente Pelechano, Joan Fons, Manoli Albert, and Oscar Pastor

Department of Information Systems and Computation
Technical University of Valencia
Cami de Vera s/n, E-46022, Spain
{pele,jjfons,malbert,opastor}@dsic.upv.es

Abstract. This paper presents a strategy for the systematic development of dynamic web applications. The proposal extends an existing OO software production method introducing navigation and presentation expressiveness for applications of this kind. This extensions allow the method going from the problem space to the solution space in a systematic way.

1 Introduction

In the last decade, the Web has become the favourite platform for the development of information systems. The reach is constantly expanding and so are the number and size of the applications, along with the underlying complexity, range of purposes and the time needed to develop and maintain them [1].

The development of a Web application is a multifaceted activity, involving not only technical questions, but also organizational, managerial, and even social and artistic issues [2]. The term *Web Engineering* is used to refer to the methods, techniques and tools that have appeared to provide methodological support to undertake the development of applications of this kind.

Navigational and presentational modelling is a key point to develop web applications [3]. Many approaches have appeared following these main ideas to enrich classic conceptual modelling methods to introduce the *new* requirements. These approaches also define a systematic method to specify and to develop a full web solution. Some representative examples are OOHDM [4], WebML [5], and UWE [6]. These methods combine very well known models (OO and Entity-Relationship models) introducing new models and abstraction primitives to capture the essential expressivity of web applications.

The OOWS method introduces two new models to the OO-Method [7], an OO software production method with systematic code generation. These models (navigational and presentational models) allow to specify the inherent characteristics of web applications, and they are the input for our web applications generators/compiler.

^{*} This work has been partially supported by the CYTED Program, Project VII.18, WEST and the MCYT Project with ref. TIC2001-3530-C02-01.

2 OOWS: A Method for Web Conceptual Modelling

OOWS (Object Oriented Web Solution) is the extension of the OO-Method [7] that introduces the required expressivity to capture navigational and presentational requirements to develop web applications.

It defines a full software development process for the web that comprises two major steps: “*system specification*” and “*solution development*”. At first, a full specification of the system requirements is built. A strategy oriented towards generating the software components that constitute the solution (the final software product) is defined in the second step.

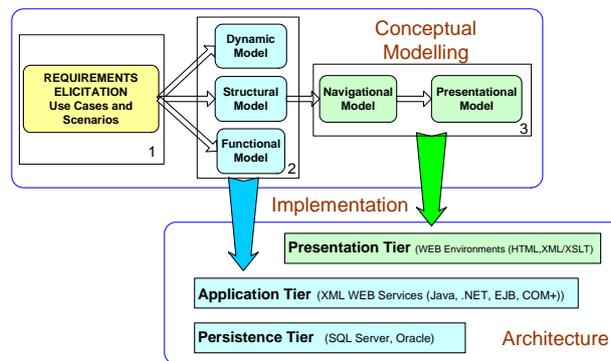


Fig. 1. Methodological Approach

The steps and their associated techniques are the following:

- 1. Conceptual Modelling:** conceptual models are built to appropriately capture the requirements of web applications. The modelling tools allow us to specify the functional, navigational and presentation requirements of dynamic web applications. The modelling process is divided into two steps:
 - (a) *functional requirements elicitation.* Techniques based on use cases and scenarios are applied to build a conceptual schema (class, sequence and state diagrams are built). An extensive work have been developed in our research group in the OO-Method context [8] See box number 1 in Figure 1.
 - (b) *classic conceptual modelling.* Using Structural, Functional and Dynamic models, the system structure and behavior is captured (box number 2).
 - (c) *navigational and presentational modelling.* A **navigational model** is built in order to capture the navigational requirements. Once the navigational model is built, presentation requirements are specified using a **presentational model** using a set of patterns. These are the extensions that the method introduces to the OO-Method (box number 3).
- 2. Architectural Design:** A multi-tier architectural style is used taking into account the nature of the **web services**.

3. **Implementation:** A set of correspondences (transformation rules) between the conceptual abstractions and the software elements that implement each tier of the architecture are defined, making intensive use of **design patterns**.

3 Navigational Modelling

This model allows to capture the navigational semantics of web applications by describing how users would access the system information and functionality. This model uses a **User Diagram** to represent the users of the system and their inter-relationships. For each kind of user, a **Navigational Map** is built to structure and organize the view of the system. **Navigational Contexts** (nodes of the map that are stereotyped with the `<<context>>` reserved word) represent specific views over a set of class attributes, operations and relationships from the Structural Model. Figure 2 shows the User Diagram and a Navigational Map from a Conference Review System in the Web.

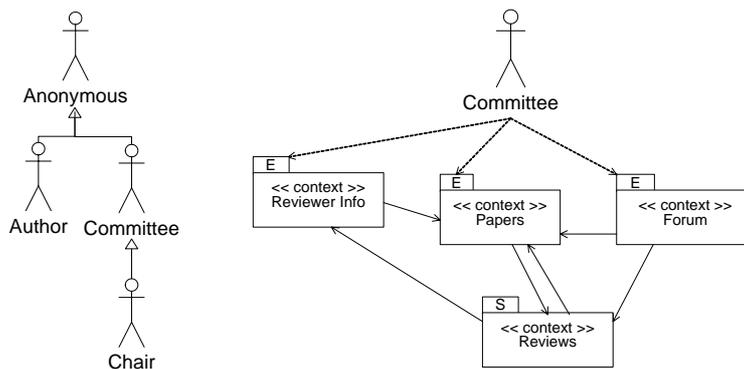


Fig. 2. User Diagram and Navigational Map

4 Presentational Modelling

Once the navigational model is built, we specify some presentation requirements of web applications. Using a set of predefined patterns (layout, information paging, ordering, ...), the method allows to capture a high-level abstract presentation description to obtain the final web interface.

Figure 3 shows the Reviews navigational and a presentational context for the Conference Review System. The first context “specifies” which information of the structural model would be visible for the Committee Member (class attributes, operations and relationships), and the second (with presentation information) gives additional information related to how this user would “see” the information.

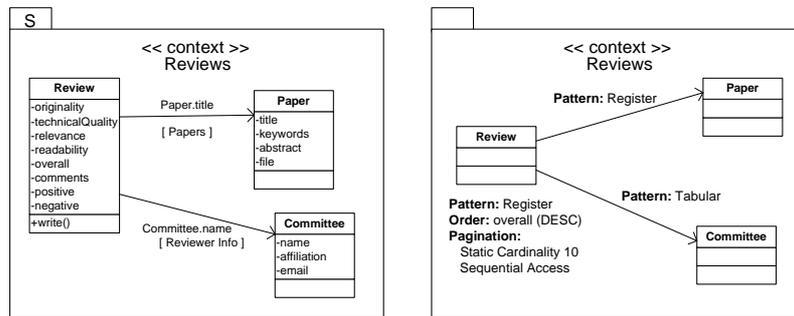


Fig. 3. Reviews Navigational and Presentational Context

5 Conclusions and Further Work

The ideas presented in this paper extend a current OO method to develop web applications. Two models have been introduced to properly capture web application requirements (navigational and presentation models). They have been integrated into an existing model-driven approach (OO-Method) that is capable to generate code from conceptual schemas. Currently we are developing a CASE tool to support the full development process of web applications using our integrated model-driven approach.

References

1. Muruguesan, S., Desphande, Y.: Web Engineering. Software Engineering and Web Application Development. Springer LNCS - Hot Topics (2001)
2. Fraternali, P.: Tools and approaches for developing data-intensive Web applications: a survey. *ACM Computing Surveys*, ACM Press **31** (1999) 227–263 ISSN:0360-0300.
3. Rossi, G., Schwabe, D., Lyardet, F.: Web Application Models are More than Conceptual Models. In: 19th International Conference on Conceptual Modeling (ER'00), Salt Lake City, USA, Springer-Verlag (2000)
4. Schwabe, D., Rossi, G., Barbosa, S.: Systematic Hypermedia Design with OOHD. In: ACM Conference on Hypertext, Washington, USA (1996)
5. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a Modeling Language for Designing Web Sites. In: Proc. of the 9th International World Wide Web Conference, WWW9, Elsevier (2000) 137–157
6. Koch, N.: Software Engineering for Adaptive Hypermedia Applications. PhD thesis, Ludwig-Maximilians-University, Munich, Germany (2000)
7. Pastor, O., Gómez, J., Insfrán, E., Pelechano, V.: The OO-Method Approach for Information Systems Modelling: From Object-Oriented Conceptual Modeling to Automated Programming. *Information Systems* **26** (2001) 507–534
8. Insfrán, E., Pastor, O., Wieringa, R.: Requirements Engineering-Based Conceptual Modelling. *Requirements Engineering* **7** (2002) 61–72