

# Modelling Dynamic Personalization in Web Applications<sup>\*</sup>

Irene Garrigós, Jaime Gómez, and Cristina Cachero

IWAD Group

Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Alicante. SPAIN

{igarrigos,jgomez,ccachero}@dlsi.ua.es

<http://www.dlsi.ua.es/iwad/>

**Abstract.** This article presents an extension of the OO-H conceptual modelling approach to address the particulars associated with the design and specification of dynamic personalization. We describe how conventional navigation and presentation diagrams are influenced by personalization properties. The main benefit is that personalization specification can be modified without recompiling the rest of the application modules.

## 1 Introduction

Current Web Engineering approaches [5] help designers to make easier the understanding, development, evolution and maintenance of web applications. These methods are based on new constructors and hypermedial views [8,9,2,10] that broach the problem of navigation/presentation of the user across the information space. We argue that new techniques to extend metamodels with personalization aspects are needed. This article presents how the *Object Oriented Hypermedia Method (OO-H)* [7,6,1] is extended to support dynamic personalization. We describe how conventional navigation and presentation diagrams are influenced by personalization properties. These properties are captured in the form of external files written in XML, that represent the rules. These rules, that form the variable part of the interface logic, will be treated at execution time by an engine included in the execution architecture. The support for this architecture is achieved by two models: A reference model, that also registers the user activity in the system. A user model that collects the needed information to personalize.

The remainder of the article is structured as follows: Section 2 shows the elements that support the personalization in OO-H. Section 3 presents the concept of association rules to support the specification of dynamic personalization. Finally, section 4 presents the conclusions and further work.

---

<sup>\*</sup> This paper has been partially supported by the Spain Ministry of Science and Technology, project number TIC2001-3530-C02-02.

## 2 Personalization Support in OO-H

The architecture of OO-H has been extended to support dynamic personalization. More specifically, personalization properties are captured at navigation/presentation level and are reflected in their corresponding conceptual models (NAD, APD) by means of a set of association rules. In this way the design and the generation of the navigation logic is specified in two parts: a stable part, that is independent from the properties of personalization, and a variable part, that supports the treatment of these rules. Finally, a rule engine provides the context to interpret the generated rules at execution time. The support for this architecture is achieved by a *Reference Model* that allows to capture the relevant properties of personalization. Moreover, a *User Model* must be defined to support the personalization requirements.

In OO-H the *User Model* captures information about the features that the system believes the user has. The information that should contain this model depends on the personalization requirements that we want to support. In OO-H the user model is centered on the concepts of user and role, the same that in another hypermedial approaches [2]. For the provision of personalized views to the user, OO-H provides a basic user model which is constructed around a class named *User*, that provides the information and behaviour that must be inherited by every <<actor>> of the system. A user may not have a role associated, in this case s/he would be treated as an *anonymous user*. Moreover, a user can only have associated a role at the same time. This model can be enriched to support the desired personalization policy adding attributes, methods or links from the class *User* to the rest of the classes of the domain or the *OO-H Reference Model*, which is presented next.

In OO-H, a repository contains the initial set of basic elements of information on which the desired personalization policy can be established. This is what is called the *Reference Model*. A first version of this repository was presented in [4]. The current OO-H Reference model (see Fig 1) structures the modelling of personalization in OO-H in three parts: user profiles, context information and association rules. For the purpose of this paper we must pay more attention to the part B of Fig. 1, it is, to the association rules that are presented in depth in the following section. This type of rules allow to capture the personalization properties embedded directly in the models of OO-H.

## 3 Specification of the Personalization in OO-H

OO-H incorporates an external mechanism to model the personalization of the adaptive and proactive applications in form of association rules. An advantage of using these rules is that they are defined in external files and can be modified in an independent way from the rest of the application. The rules have been divided in three types: *Acquisition Rules*: in which the needed information for personalization is collected. *Personalization Rules*: support the action of personalization.

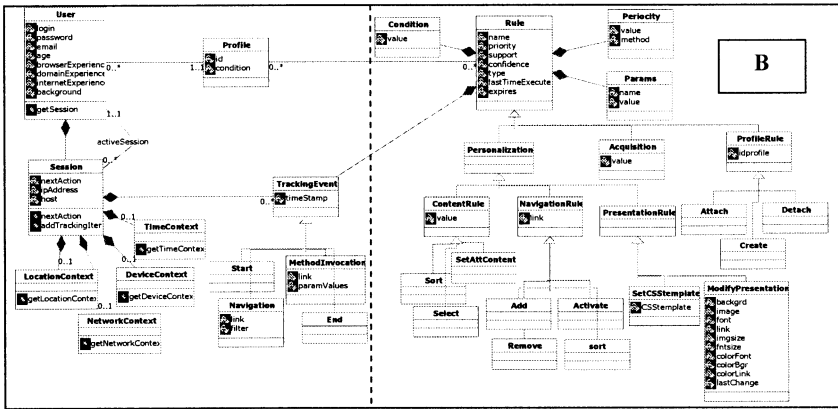


Fig. 1. Personalization Framework

As we have seen in the Fig. 1, this type of rules are also categorized, based on what we want to personalize, in:

- Content Rules*: using these rules we can select (*Select*), sort (*Sort*) or modify (*SetAttContent*) the content information of the application.
- Navigation Rules*: With this type of rule we can *Add*, *Remove*, *Activate* or *Sort* any links in the user navigation.
- Presentation Rules*: The possible actions that can be associated to this type of rule are: *SetCSSTemplate* or *ModifyPresentation*, according to whether we want to use a predefined template of presentation or we want to modify a specific value. We have a set of CSS templates, that can be associated to specific cases (as for disabled persons) captured in the *Reference Model* and a *Presentation Class* with presentation features. Moreover, in the *User Model* we can complete this *Presentation Class* with the features designed for a specific conceptual model. The rules permit the selection of a predefined template or the modification of specific features of the *Presentation Class*.

*Profile Rules*: With these rules we can associate (*Attach*) or disassociate (*Detach*) specific behaviours to user profiles, or we can create a new profile (*Create*). OO-H supports the specification of these rules by means of an XML schema. It is important to note that the execution architecture of the generated applications from OO-H models allows to modify and reprocess this schema without recompiling the rest of modules. In the schema the different XML elements correspond to classes/attributes of the frame presented in Fig. 1. All the rules are ECA (Event-Condition-Action) rules [3].

## 4 Conclusions and Further Work

Web Engineering methods have to provide a well-defined software development process by which the community of software engineers can properly design pow-

erful web-based applications in a systematic way. Our purpose has been to address these problems in the context of the OO-H conceptual modelling approach that has been successfully proven for the design of web applications. We focus on how to properly capture the particulars associated to the design of dynamic personalization. In order to achieve this goal, OO-H adds acquisition and personalization rules, which define the suitable semantics for capturing and representing the specific functionality of dynamic personalization. In this way, navigation and presentation models can be compiled to obtain an XML specification that represents the desired dynamic personalization. The final web application is viewed as a composition of a stable and variable part, where the variable part is interpreted by the rule engine to give personalization support. The main benefit is that personalization specification can be modified without need for recompiling the rest of the application modules. Others relevant contributions of this paper are the following: A user model that describes how the knowledge that the system has about the user is captured, A reference model that provides a way to extend the OO-H metamodel by means of a specific a set of information structures.

## References

- [1] C. Cachero, J. Gómez, and O. Pastor. Object-Oriented Conceptual Modeling of Web Application Interfaces: the OO-HMethod Presentation Abstract Model. In *1<sup>st</sup> International Conference on Electronic Commerce and Web Technologies (ECWEB'00)*, Greenwich, London., volume 1875, pages 206–215. Springer-Verlag. Lecture Notes in Computer Science, 09 2000.
- [2] S. Ceri, P. Fraternali, and A. Bongio. Web Modeling Language (WebML): a modeling language for designing Web sites WWW9 Conference. In *First ICSE Workshop on Web Engineering, International Conference on Software Engineering*, 05 2000.
- [3] U. Dayal. Active Database Management Systems. In *Proc. 3rd Int. Conference on Data and Knowledge Bases*, pages 150–169, 1988.
- [4] I. Garrigós, C. Cachero, and J. Gómez. Modelado Conceptual de aplicaciones adaptivas y proactivas en OO-H. In *II Taller sobre Ingeniería del Software Orientado al Web (JISBD 2002)*, 11 2002.
- [5] A. Ginige and S. Murugesan. Web Engineering: an Introduction. *IEEE Multimedia Special Issue on Web Engineering*, pages 14–18, 04 2001.
- [6] J. Gómez, C. Cachero, and O. Pastor. Extending a Conceptual Modelling Approach to Web Application Design. In *12<sup>th</sup> International Conference on Advanced Information Systems (CAiSE'00)*, volume 1789, pages 79–93. Springer-Verlag. Lecture Notes in Computer Science, 06 2000.
- [7] J. Gómez, C. Cachero, and O. Pastor. Conceptual Modelling of Device-Independent Web Applications. *IEEE Multimedia Special Issue on Web Engineering*, pages 26–39, 04 2001.
- [8] N. Koch, A. Kraus, and R. Hennicker. The Authoring Process of the UML-based Web Engineering Approach. In *Proceedings of the 1st International Workshop on Web-Oriented Software Technology*, 05 2001.
- [9] Daniel Schwabe and Gustavo Rossi. A Conference Review System with OO-HDM. In *First International Workshop on Web-Oriented Software Technology*, 05 2001.
- [10] Olga De Troyer and Sven Casteleyn. The Conference Review System with WSDM. In *First International Workshop on Web-Oriented Software Technology*, 05 2001.