

# Personalizing Digital Libraries at Design Time: The Miguel de Cervantes Digital Library Case Study\*

Alejandro Bia, Irene Garrigós, and Jaime Gómez

Web Engineering Group. DLSI  
University of Alicante, Spain  
{abia, igarrigos, jgomez}@dlsi.ua.es

**Abstract.** In this article we describe our experience in the development of a personalizable dissemination model for the Miguel de Cervantes Digital Library's Web-based newsletter-service, which combines adaptive with adaptable personalization techniques, being capable of ranking news according to navigation-inferred preferences and then filter them according to a user-given profile. We explain how Web engineering design techniques have been applied to make that service evolve into a more adaptive personalization approach obtaining more effective results. The work is presented in the context of the OO-H [5] web design method.

## 1 Introduction

The Miguel de Cervantes Digital Library (MCDL) of the University of Alicante is the biggest electronic publishing project in Spain, and perhaps the biggest digital library of Spanish texts on the Internet, currently with more than 10000 entries in its catalogue. One of the goals of the MCDL is to act as a communication channel for the academic community. In this sense we have implemented a number of communication services, and we try to maintain a permanent communication with our readers. It is obvious that the personalization of these services is a key issue in the management of the DL in order to get a better use.

The idea of giving the user the impression of interacting with the application by means of a dedicated interface, specifically tailored to the user's needs and preferences is a "must" in current Web development. Content personalization systems select, adapt and generate contents according to user models that define information needs. Dissemination models help to achieve this goal. In a dissemination model environment, users subscribe to an information dissemination service by submitting a set of personalization settings that describe their interests, which are usually called profiles. Then they passively receive information filtered according to such profiles. Enhanced dissemination models allows the sender of such information to get a feedback of the users interests, which may evolve with time, or to get a feedback of the users opinions on the information received. In this context, user profile and preferences can be acquired in three ways [1]: be set by the application designer by

---

\* This paper has been partially supported by the Spanish Ministry of Science and Technology (Ministerio de Ciencia y Tecnología de España), project TIC2001-3530-C02-02.

means of *stereotyped* techniques, be given by the users themselves by means of *interview* techniques (this includes input forms) or be automatically inferred by the application based on user activity (*observation* techniques). The applications of the second type are also referred to as “*adaptable*” while the third type are “*adaptive*”, and when the origin of personalization information is a system event we talk about “*proactive*” applications [2, 3].

## 2 Personalization of the Newsletter Service: Ranking and Filtering

In the first stage, our profile was very simple and allowed only for the choice of one or more possible thematic newsletters out of five [4]. In the second stage of development, we have remodeled the personalization approach to both offer a finer granularity (more detailed personalization choices) and also obtain explicit (adaptable approach) and implicit (adaptive approach) feedback from users which allows for dynamic changes in personalization settings and will also provide, as said before, useful information on user navigation habits and preferences. The user model we have implemented for newsletters automatically and transparently incorporates information gathered from user navigation (adaptive part). In addition, the user can set-up some filtering restrictions and customization preferences when registering for this service (adaptable part). The final model is based both on implicit interests on certain digital library sections (information gathered during navigation) and on explicit preferences that compose the user profile.

News are classified by category and subject-matter. Categories are: new publications (new digital resources), future publications, new sections, chat announcements, call for papers, suggestions from our departments, letters from readers, visits of important people, contests, and the remainder are classified as general news. Subjects or matters are derived from the actual thematic structure of the DL. Each theme section or subcollection generates a subject-matter, as for instance: Latin-American literature, humanities research, history, children's literature, theatre, interactive services of the DL, computers and humanities, critical studies, tribute to Hispanists, Argentine Academy of Letters, PhD theses, movies, magazines/journals, recently printed books, law, and many more. This allows for a very fine granularity. Every time the user clicks on an entry of the newsletter's table of contents, the Web page jumps to a single piece of news, and the server increments in one the corresponding category and subject-matter counters. Only one category but multiple subjects can be assigned to a piece of news. Relative access frequencies can be computed for categories and for subjects. Then news can be given a ranking value for a given user for a given news-reading session, which is calculated as the sum of subject frequencies of the subjects corresponding to a given piece of news, multiplied by the frequency of its category. For instance, if a user has an access frequency of 0.3 for the “new-publications” category, and, 0.1 for the “history” and 0.2 for the “PhD-theses” subjects, then a piece of news announcing the publication of a PhD thesis on history will weight  $(0.1 + 0.2) \cdot 0.3 = 0.09$ , and will be ranked accordingly. On registration, the users can specify Boolean constraints for categories and subjects, saying which ones should be sorted out and which should be displayed. This profile can be modified by the user.

Newsletters are accessed through a monthly index, where news are ordered first by category and then by subject, according to the dynamically computed ranking. But not

all the ranked news appear, they are filtered according to the user explicit profile. The first  $N^1$  ranked entries that pass the filter are shown openly, and the rest appear as a collapsed “more news” button.

### 3 Adapting the MCDL Using the OO-H Personalization Frame-Work

The OO-H (*Object Oriented Hypermedia*) method [5] is a generic model, based on the object oriented paradigm that provides the designer with the semantics and notation necessary for the development of web-based interfaces. Web design modelling is achieved by means of the two complementary views, namely (1) the Navigational Access Diagram, that enriches a standard UML class diagram with navigation and interaction properties, and (2) the Abstract Presentation Diagram that gathers the concepts related both to structure of the site and specific presentation details respectively. OO-H also supports dynamic personalization (described in [6]), allowing the designer to better tailor the site to the particularities and needs of the individual user. This is done by means of a personalization framework that is a part of the model. That framework can be instantiated by the web designer, and connected to any OO-H based site to empower it with personalization support for (individual) users. The framework can be divided in two parts: (1) the user model, which allows to store the beliefs and knowledge the system has about the user, and (2) the personalization model, which is used to specify the personalization strategy for the different groups of users.

In the context of the MCDL, we have used this approach to personalize the newsletters. In Figure 1 we can see the user model for the MCDL case study.

The user model represents the variable part of the conceptual model. We have two association classes in which we store by *subject-matter* and by *category* the number of times that the user consults a piece of news. We also store the user preferences in a class that the user can change when desired.

The personalization model allows the designer to define a collection of rules that can be used to define a personalization strategy for a user or group of users. The rules are Event-Condition-Action [6] rules: they are triggered by a certain event (e.g. a browsing action, the start of a session) and if a certain condition is fulfilled, the associated action is performed. Figure 2 shows the NAD corresponding to this modeling example. The NAD’s entry point is a link to a user-connection parameters form to access the system (login, password). Registered users are shown a link to access the newsletter. The personalization rule is attached to it. When the user clicks on it, a table of contents appears, showing the ranked and filtered list of news. To determine the user interest in a piece of news, we have the *Acquisition* rule that is attached to the “consult” service link. In this rule we’ll acquire the number of times a user consults a piece of news and its related *subject-matters* and *category*, and using these two attributes together with the user-set preferences profile, we rank and filter

---

<sup>1</sup> N is a user given parameter.

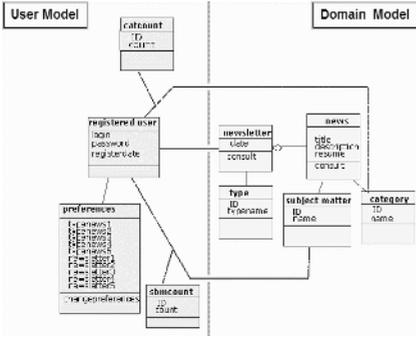


Fig. 1. User and Domain models

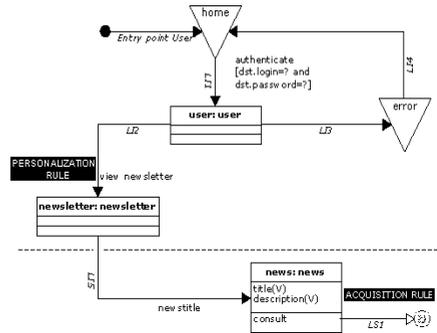


Fig. 2. NAD for the newsletters system

the news according to the user’s interests. The ranking is computed with the above described algorithm. Later, this NAD is compiled to obtain the XML specification that is interpreted by a rules engine to give personalization support to our application. That compilation process produces 2 types of XML specs acquisition rules and personalization rules that are presented next.

**Acquisition Rule:** Below the dotted line of figure 2, we can see the part of the NAD diagram that holds the rule that stores the preferences information (acquisition rule). Every time a user consults a piece of news (event that triggers the rule), the corresponding subject-matters and category counters must be incremented. To store this information (in the user model), we need to check whether the subject-matter and category of the piece of news the user has consulted are the same we are going to update in the User model. With these data we can predict the user preferences. The XML specification of the acquisition rule defined by our system follows:

```

<rule type="acquisition" name="StorePreferences" >
  <params>
    <param name="idsbm"
      value="session.registereduser.newsletter.news.subjectmatter.id" />
    <param name="idcat"
      value="session.registereduser.newsletter.news.category.id" />
  </params>
  <event type="MethodInvocation" link="Consult" />
  <condition "idsbm=newstittle.news.subjectmatter.id and
    idcat=newstittle.news.category.id" />
  <action value="session.registereduser.sbmcount.count++
    and session.registereduser.catcount.count++" />
</rule>

```

We use the subject-matter and category identifiers of the piece of news as parameters of the rule, to simplify the expressions.

**Personalization Rule:** Above the dotted line of figure 2, we can see the part of the NAD diagram that holds the rule that models this requirement (personalization rule). When the link *View Newsletter* is activated the rule is triggered. In this case we have navigation personalization, because we have to sort the links the user is going to see. We have the two attributes as parameters (stored in the user model by the acquisition

rule). The personalization event indicates that the link *View Newsletter* must be active. When this link is activated, the action will be executed: the news are sorted and shown.

```
<rule type="personalization:navigation" name="Recommendations" >
  <params>
    <param name="sbmcount" value="
session.registereduser.sbmcountcount "/>
    <param name="catcount" value="
session.registereduser.catcountcount "/>
  </params>
  <event type="Navigation" link="View newsletter"/>
  <action type="Sort" link="newstittle" byval1="sbmcount"
byval2="catcount" ORDER="DESC"/>
</rule>
```

## 4 Conclusions

Concerning the dissemination model for these newsletters, we have enhanced the granularity by offering more detailed personalization options, which allow us to rank the news based on preferences gathered from user navigation (observation model). The design of this solution was performed according to the OO-H Model. This technology can significantly increase the productivity at the time of developing Web applications. The MCDL, with this effort, struggles to fulfill its objective of spreading research knowledge to the global academic community through the Web. Our goal is not only the mere publication of research work, but to build a rich and open communication channel for the global scientific community. The newsletter service described here plays a key role in this communication effort.

## References

1. Nora Koch. *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process*. PhD thesis, Institut für Informatik, Ludwig-Maximilians-Universität München, December 2000.
2. Gerti, Kappel, Werner, Retschitzegger, Briggit, Pöll, and Wieland, Schwinger. Modeling Ubiquitous Web Applications . Thw WUML Apporach. In: Proceedings of the International Workshop on Data Semantics in Web Information Systems. Yokohama, Japan. 2001 (11).
3. Wu, H. A Reference Architecture for Adaptive Hyermedia Systems. 2001.
4. Alejandro Bia and Jaime Gómez, Developing a simple production and dissemination model for a complex DL-news service using XML, XSLT and TEI., RCDL'03 Saint Petersburg State University, Moscow, 29-31 October 2003.
5. J. Gómez, C. Cachero. OO-H method: Extending UML to Model Web Interfaces. In: Information Modeling for Internet Applications. Pages. 50-72. Idea Group. 2002
6. I. Garrigós, J. Gómez and C. Cachero. Modeling Dynamic Personalization in Web Applications. Third International Conference on Web Engineering (ICWE'03), LNCS 2722, pages 472-475. Springer-Verlag Berlin Heidelberg, 07 2003.