

A Structured Approach to Personalize Websites using the OO-H Personalization Framework*

Irene Garrigós¹, Sven Casteleyn² and Jaime Gómez¹

¹ Universidad de Alicante, IWAD, Campus de San Vicente del Raspeig, Apartado 99
03080 Alicante, Spain

{igarrigos, jgomez}@dlsi.ua.es

² Vrije Universiteit Brussel, Department of Computer Science, WISE, Pleinlaan 2, 1050
Brussel, Belgium

Sven.Casteleyn@vub.ac.be

Abstract. Most current web engineering approaches don't have clear criteria for personalization, nor a clear strategy that designers can use to classify users and formulate a personalization strategy for the web site at design time. In this paper, we present a structured approach to personalize websites based on three criteria: characteristics, user requirements and context. The users are classified into groups of users (profile groups) based on these criteria. For each group, a personalization strategy is defined by attaching personalization rules to its profile group. The work is presented in the context of the OO-H method which' personalization framework can be instantiated by the web designer, and connected to any OO-H based site to empower it with personalization support. Finally, we introduce a tool that allows designers to specify and generate the personalizable client-side of a web application based on this approach.

1 Introduction

Nowadays, web sites are complex applications, offering both static and rapidly changing information and functionality. When implemented at hoc, this leads to enormous usability and maintenance problems [9] for web designers.

Introduction of web design methods and methodologies [10] [1] [7] have provided some solutions for designers (design support, help in determining consistent structuring, easier maintenance) and for visitors (better tailored content, easier navigation). In order to better tailor the site to one particular user, or a group of users, some methods provide personalization support (see next section for an overview). However, approaches vary widely both on how and what they personalize, and most of the approaches do not provide an underlying CAWE1 tool to support the personalization. The lack of such tools causes the personalization to be implemented in an ad hoc manner.

* This paper has been supported by Ministerio de Educacion y Ciencia with the METASIGN project ref. Code TIN2004-00779.

¹ Computed Aided Web Engineering

In this paper, we tackle the problems described above by providing support for defining personalization during web site design cycle and by offering the designer a method and a tool to specify the adaptation of the client-side of a web application (i.e. personalize). We present a structured approach to personalize websites based on three criteria of a user profile: user characteristics, user requirements and user context. A *user profile* can be defined as a set of data representing the significant features of the user. User profiling is becoming more and more important in adaptive web applications, due for example to the heterogeneous devices used to access the World Wide Web. In our approach, users are classified into user groups², called profile groups, based on the values of the defined criteria in their user profiles. The profile groups are defined by means of profile rules. For each group, a personalization strategy is specified by attaching personalization rules to its profile group. In this way, the OO-H (Object Oriented Hypermedia) method allows personalization of the content and the navigation of the website, both for single users, and groups of users.

The paper is structured as follows. In the next section, related work is studied. The paper continues describing in section 3 the general criteria for personalization showing, by means of a running example, how to categorize each criterion. The criteria are embedded into a personalization framework in the context of the Web Design Method OO-H [5] [6] that is presented in section 4. We continue in section 5, explaining how personalization relates to navigation in OO-H. Section 6 shows how the personalization strategy for our running example is described using the Personalization Rule Modeling Language, and how it is connected to navigation. Finally, section 7 sketches some conclusions and further work.

2 Related Work

There are other methods (besides OO-H) that also allow personalization on the basis of the user profile information. Some of them permit adaptation for users with some common user profile data (e.g. groups of users). But, existing methods supporting profiling don't consider the attachment of users to profile groups, although groups are implicitly defined (i.e. users with the same user profile information). In OO-H, the total user base can be partitioned (on the basis of user profile information) into groups. The advantage of explicitly defining these user groups is that for each user only the personalization rules attached to her/his profile group(s) are considered (i.e. the system has not to check the rest of the personalization rules). Next we explain how other methods use the profile information to personalize.

WUML [7], in the context of ubiquitous computing, has personalization based on context. The authors propose an object oriented framework (which consists out of four models) that can be extended by the designer. The context and profile models provide detailed information about the environment of an application and trigger the actual customization as soon as the environment changes. Context

² A group can be composed (only) by one user, and a user can belong to one or more profile groups.

represents current and historical information about the environment of the application which is automatically monitored. Profiles cover more stable information which is explicitly given by a designer or a user (e.g. user preferences) or is transparently acquired by the system itself (e.g. usage statistics).

In the OOHDM approach [10] user roles are considered, those users roles are (only) static profile groups (defined at design time and cannot be dynamically changed). Different web applications can be built for different user profiles reusing a conceptual schema. The navigation model is built as a view over a conceptual model, thus allowing the construction of different models according to different user profiles. OOHDM also supports dynamic customization. In the navigation model, node contents and structure can be customized and links and indexes can be personalized. Only one of the six different kinds of navigational contexts supported allows to capture dynamic user preferences. In the interface model different layouts can be defined according to user preferences or selected devices.

In the Hera methodology [4], two kinds of adaptation are considered: adaptation with respect to devices capabilities and user preferences stored in a profile (adaptability, i.e. personalization defined at design time), and adaptation based on user navigation history stored in a User Model during browsing (adaptivity, i.e. personalization performed while the user is browsing the application). To model the adaptability in the Hera methodology, the Composite Capability / Preference Profile (CC/PP) [9] offers a framework to model profiles that characterize device capabilities and user preferences. In Hera static adaptation (i.e. adaptability) is applied to users belonging to a profile, (e.g. users with a wap phone). To add adaptivity functionality to Hera the authors use the AHA (Adaptive Hypermedia Architecture) system [3].

None of these methods provide the designer with a structured user model based on (all) the features of the user (characteristics, context, requirements), nor do they offer the designer a high level language to personalize based upon these criteria³. And most importantly, none of these methods provides a rigid, industrially tested CAWE tool supporting the proposed (personalization) design.

3 Criteria for personalization

In this section, we will explain the different criteria upon which a personalization strategy can be built. We will concentrate here on the user-specific features (user profile information) rather than on domain-specific features. The former ones are features independent from the domain and dependent only on the user, the latter ones consist of features specific for the domain (e.g. history of buys in e-commerce sites). We have classified these user-specific features distinguishing between user characteristics, user requirements, and context.

We first elaborate on these features and the role they may play in personalization. For this purpose we have modelled a simplified version of the Fnac web site (<http://www.fnac.com>, online shop of spare time cultural products)

³ Implicitly, our approach also includes support for behaviour by means of acting upon events, see section 3.2

and we have added personalization based on the criteria defined in this paper. Now we will explain the different criteria upon which can be personalized by means of this running example. The next sections will cover personalization itself for the case study defined here.

3.1 User characteristics

User characteristics (such as age or language) may be relevant for the structure, content or presentation of the web site. Designers might try to exploit the user's characteristics to adapt the site to either arrive at a better usability and/or have a greater benefit for the company. Examples include: adapting the font size for users with low vision; avoiding the use of specific colors in case of color blindness; etc.

In our case study we will show or not show some products depending on the user's age. (E.g. we will "censure" some books, dvds... for non-adults users). The user's age is considered as a relevant characteristic of the user and we will define a profile group based on this characteristic. The designer then can specify a personalization strategy over the defined group of users (e.g. users whose age is less than 18).

3.2 User requirements

Every user comes to a web site for specific reasons, and with specific goals. He is looking for some specific information or functionality and he expects to find it on the web site, i.e. the user has *user requirements*. A good web site design method starts out with specifying the user requirements, elaborating these requirements further and specifying the information or functionality needed to fulfil the requirements at a conceptual level (e.g. using modelling techniques such as UML, ER, ORM, ...). Next, these conceptual representations are translated into actual web pages, using some presentation and layout mapping.

As an example of personalization based on (fulfilling) user requirements, we can consider adding a link to information relevant for a certain user, even though it was not originally assessed as a requirement for this particular user.

Continuing the running example, we will consider the (fine-grained) requirements *browseBooks*, *browseMusic*, *browseDvd*... (rather than the course-grained requirement "browse products"). In section 4 (and 6) we will define as an example a profile for the users interested in books by specifying the requirement of this type of users (*browseBooks*). A personalization strategy will be applied over this group of users: if the user is interested in books, we offer him new books that have arrived on the front page.

3.3 User Context

The variety of devices that can access the web gets larger every day, and the particularities of the different devices may influence the way we want to structure

or present information for/to the visitor, or even which information we offer. In other words, information concerning the context of the current session of the user will be relevant to consider for personalization.

In the OO-H framework four kinds of context are considered: LocationContext (ubiquity of the user), NetworkContext (e.g. latency, speed, bandwidth ...), DeviceContext (e.g. PC, PDA, WAP ...), and TimeContext (date and local time of the connection). In the Fnac example we consider a type of context personalization based on the device context of the user. We will define a profile for users who are browsing the website using a small display device (in the example we consider PDA, mobile phone and MP3 player). The personalization strategy for this group of users consists of not showing any pictures associated to products, since resolution of the PDA, MP3 player and mobile phone screen does not allow images to be displayed conveniently, and big images would also hinder convenient navigation.

Now that we have defined the different (user related) criteria considered for personalization⁴, we will explain the (general) OO-H framework, and see how the different criteria are represented within the OO-H personalization framework.

4 The OO-H personalization framework: general overview

The OO-H (Object Oriented Hypermedia) Method 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. Figure 1 shows the package view of the whole approach. Web design modelling is achieved by means of the two complementary views, namely (1) the Navigational Access Diagram (navigation package), that enriches a standard UML class diagram (structure package) with navigation and interaction properties, and (2) the Abstract Presentation Diagram (presentation package) that gathers the concepts related both to structure of the site and specific presentation details respectively.

The OO-H also supports dynamic personalization (a preliminary version was described in [5]), 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 goal of the personalization framework is twofold: (1) provide the designer with the means of gathering and storing all information needed to personalize the site, and (2) provide the designer with the means of specifying the personalization policy for the different users (adapting structure, content, layout and/or presentation). According these goals, the framework can be divided in two parts: (1) *The user model* and (2) *The personalization model*. We now discuss both parts in more detail and indicate how they are related.

⁴ Although here, all examples of profile groups are based on one type of criterion, we can also define profiles combining different types of criteria (e.g. a profile defined by a characteristic and a context)

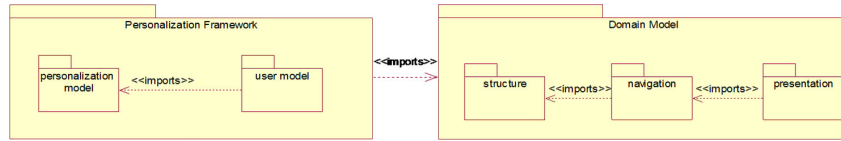


Fig 1: OO-H package view

4.1 The user model

In the user model, all the knowledge about the user is stored. It will contain information on the three criteria explained in the previous section as well as information related to the browsing behaviour of the user.

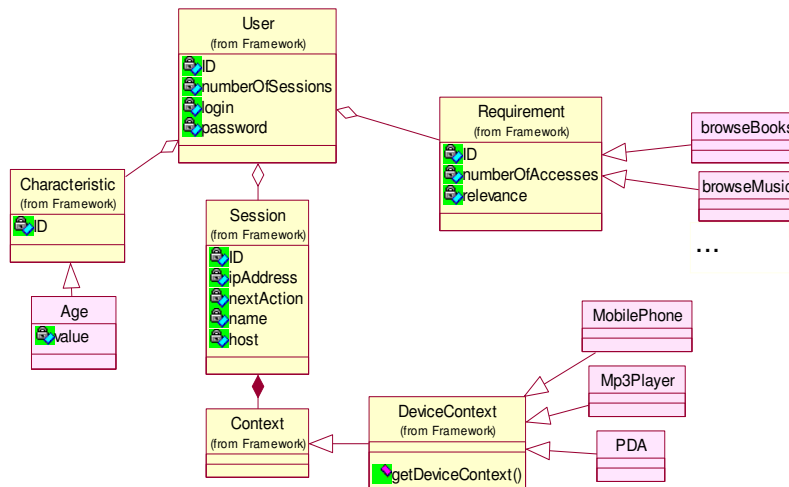


Fig 2: User Model for the FNAC system

Figure 2 shows the relevant part of the user model for the running example presented in section 3 (some attributes and methods not directly relevant for this example have been omitted). The figure shows some classes that belong to the framework package, marked with the text 'from Framework'. Those classes constitute the skeleton to adapt the generic OO-H user model to the concrete personalization requirements. In our example, the designer needs to capture information regarding the user's age. Following the criteria presented in the previous section this information can be classified as a user characteristic. That is why the class *Age* inherits from the predefined characteristic class of the user model. In the same way, the other relevant information for fulfilling the personalization requirements are captured by means of inheritance relations with the proper classes from the user model (e.g. PDA, MP3Player and MobilePhone inheriting from DeviceContext and browseBooks and browseMusic from Requirement).

4.2 The personalization model

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 [2] 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 (for example “categoryMovie=”drama””), the associated action is performed. For satisfying a personalization requirement we have to define where and what information is acquired to obtain the required knowledge to personalize (acquisition rule) and define the personalization in terms of the effects this personalization causes in the system (personalization rules). We also have profile rules to classify the users using the acquired information.

In the following table we have summarized for the different personalization requirements of our case study the different rules that should be defined.

Personalization requirement	Acquisition rule	Profile rule	Personalization rule
Adult users can see some products that are not shown to minors (<i>type: characteristic</i>)	Not defined. The user’s age is acquired in an explicit way (e.g. using a form)	If the user’s age is less than 18 we attach the user to the profile group “Minors”.	If the user belongs to the “Minors” profile, products with 18+ rated material are omitted
Users with the requirement “browse Books” will see the newest books in home page (<i>type: requirement</i>)	Not defined. The user’s requirements are acquired in an explicit way (e.g. using a form)	If the user has the requirement browseBooks, attach the user to the profile group “browseBooks”	If the user belongs to the profile “browseBooks” show the new books in the home page.
Users with small screen-devices will not be shown product images. (<i>type: context</i>)	Capture the device context when the user enters the application	If device context is PDA, Mobile Phone or MP3 Player, attach the user to the profile group “smallScreen”.	If the user belongs to the profile “smallScreen” don’t show images of products

As an example to better understand this table, consider the last row. It expresses that we’d like to support users browsing with a small screen device, by not showing them images not to overload their (small) screen. An acquisition rule is used to capture the deviceContext of a user starting a session (column 2). Based on the kind of device that is detected, the user is or is not attached to the profile group smallScreen by means of a profile rule (column 3). Finally, if the user is a smallScreen user, a personalization rule expresses that images are not shown.

The rules described in this table needed for our running example will be defined using an efficient, simple and easy to learn language defined in OO-H. The purpose of this language is to help the web designers to define all the rules required to implement a personalization strategy. This language is called PRML (*Personalization Rule Modelling Language*). The basic structure of a rule defined with this language is the following:

When event do
If condition then action endIf
endWhen

As we have said the different type of rules in OO-H are Event-Condition-Action [2] rules. In this basic structure we define the event that triggers the rule, the condition to be checked and the action to be performed. The rules defined for the running example are shown in next section and we will also show how to connect users and their personalization strategy.

5 Navigation and personalization in OO-H

Having explained the OO-H personalization framework (user and personalization model) and its use, we will now describe how a personalization strategy is specified and how exactly it relates to the OO-H Navigation Access Diagram (NAD).

A NAD is a conceptual model that provides the necessary constructs to represent how a web site visitor navigates through and accesses the information/services of the application. Moreover, it shows which events trigger the different rules and which type of criteria is being satisfied to fulfil the personalization requirements.

Figure 3 shows the NAD for the running example. The starting point (*entry point* element) is the home page where information about product *categories* (navigational class) and *novelties* (navigational class) is presented. That information is shown in the context of the home page because their corresponding links “view categories” and “view novelties” are shown in origin (*origin* property). When the user selects a category (*view categories* link), the set of products that belongs to that category is shown in a new page (*destination* property). The information shown about a product is name, *price*, *picture* and *description* (attributes of *product* navigation class). The service *BuyProduct* can be invoked by means of the service link Buy.

Personalization is embedded in a NAD by means of events, which are generated by the user throughout a browsing session. The types of events are: *Start event* that is associated with the entrance of the user in the system (activation of *Entry point link*). *Navigation event* that implies the activation of any navigation link (e.g. view categories.). *Method Invocation event* that implies the invocation of a method defined in our system (in fig. 3 we can see one of this type of link: “Buy”)⁵.

When a user enters the website (*Entry Point Link*) the Start event is launched. At this point, we attach the user to user profile(s) (by means of profile rules; see next section for examples). Obviously, to be able to attach a user to a user profile, some information about the user is needed. The required information is gathered by means of acquisition rules triggered also by the Start event (in our example the rule to get the device context) but with a higher priority than the profile rules (i.e. they are executed first). Information that cannot be gathered automatically is asked to the user at first visit by means of a form. Once a user is attached to a profile(s) only the personalization rules attached to that (those) profile(s) are considered (as far as

⁵ A 4th event not associated with any type of link, the init event, will be explained shortly.

the current user is concerned). The personalization and acquisition rules attached to a profile will be triggered by certain browsing event (Start, Navigation or MethodInvocation) generated by the user during his session. In our example all the personalization rules are triggered by Navigation events (activation of navigational links). In this way one NAD has different executions for different users (as they will be attached to different user profiles). The whole set of rules associated to a NAD is stored in a PRML file. In the next section, we show the PRML file for our running example.

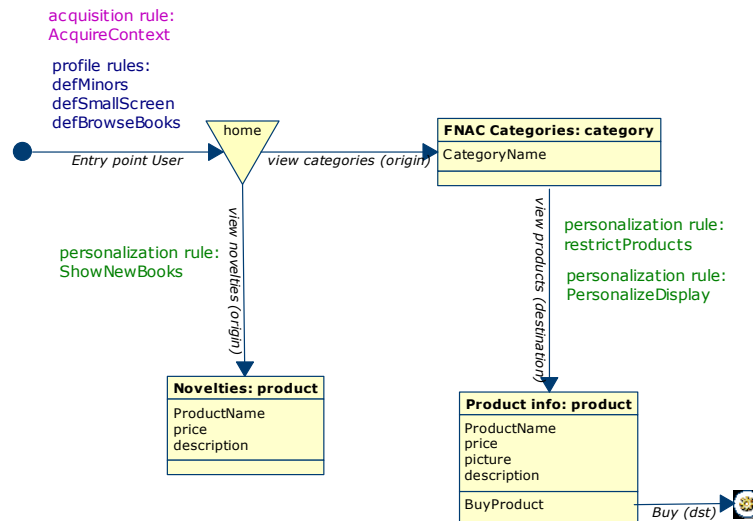


Fig. 3: (Simplified) NAD for the Fnac example

6 Personalizing the NAD using a PRML file

All that remains now is to show the content of the PRML configuration file for our running example⁶ (the rules needed in this example are described in the table in section 4.2). We can see how the structure of this file follows the organization of the personalization model in OO-H. This file has four big sections.

- The **acquisition rule section** defines the rules needed to gather the required information to classify the user into user groups. In our example we have an acquisition rule to capture the device from which the user is connecting to the website (device context). This rule gets the device context using a method defined in the personalization framework.
- The **profile rule section** contains the profile rules. The profile rules for this example, defining “Minors”, “BrowseBooks” and “smallScreen”, are all triggered by the *Start* event. When a user enters the system (i.e. starts a session), s/he is attached to the profile group(s) s/he belong to (based on the profile condition).

⁶ Some attributes of the rules have been omitted for simplicity reasons.

- The **personalization rule section** contains the personalization rules which describe the effect of personalization in the website. In our example, we have three personalization rules. All of them are triggered by *Navigation* events, but of course other events could also trigger personalization rules.
 - *The first rule* is triggered by the activation of the link “*ViewProducts*”. Each product in the site has an attribute “*allowedFromAge*”, denoting the required age to view the product. On basis of this attribute, we will only show products allowed from age greater than 18 (i.e. the action changes the property *Visible*⁷ of the Product class to *false*).
 - *The second rule* personalizes the website by omitting pictures to be shown. It is triggered by the *Navigation* event of activating the “*ViewProducts*” link. The action of not showing images (of products) is achieved by setting the property *Visible* of the picture attribute to *false*.
 - *The third rule* is triggered when the user consults the new products activating the link “*ViewNovelties*”. It specifies that the category of products to be shown should be ‘books’. The other condition checks that the selected products are “new”⁸, checking the date of addition of the product to the website.
- Finally, every PRML configuration file has an **Init section**. The deployment of the website causes the rule engine to be set up with all rules, and the launch of an INIT event). Upon this INIT event, user profiles are defined and personalization rules are associated to them. In this way we specify a personalization strategy for the defined user profiles. The acquisition rule and the profile rules are attached to the special profile OOH:all (i.e. applied to all users).

```
# ACQUISITION SECTION
#RULE:"AcquireContext" priority="high"
When start do
DeviceContext=getDeviceContext ()
endWhen

# PROFILE SECTION
#RULE:"defMinors"
priority:"medium"
When start do
  If (charHasValue (Age, <, 18))
  then
    AttachUserToPGroup ("Minors")
  endIf
endWhen

#RULE:"defSmallScreen"
priority:"medium"
When start do
  If (deviceContext="PDA" or
deviceContext="MP3" or
deviceContext="WAP") then
    AttachUserToPGroup ("smallScreen")
  endIf
endWhen

#RULE:"defBrowseBooks" priority:"medium"
When start do
  If (userhasReq ("brosweBooks")) then
    AttachUserToPGroup ("browseBooks")
  endIf
endWhen
```

⁷ In OO-H, every class and attribute has a *visible* property, denoting if it should or shouldn't be shown

⁸ We consider products new if they have been added at most one week ago.

```

# PERSONALIZATION SECTION
# RULE:"restrictProducts"
When Navigation.ViewProducts do
  If
    (Products.allowedFromAge>18)
  then
    Product.Visible=false
  endIf
endWhen

# RULE:"PersonalizeDisplay"
When Navigation.ViewProducts do
  Product.picture.Visible=false
endWhen

# RULE:"ShowNewBooks"
When Navigation.ViewNovelties do
  If (Products.category='Books' and
    Products.dateOfAddition=currentDate-week) then
    Select (name, description, price) in Products
  endIf
endWhen

# INIT SECTION
When init do
  AttachRuleToPGroup("AcquireContext", "OOH:all")
  AttachRuleToPGroup("defMinors", "OOH:all")
  AttachRuleToPGroup("defSmallScreen", "OOH:all")
  AttachRuleToPGroup("defBrowseBooks", "OOH:all")
  AttachRuleToPGroup("restrictProducts", "Minors")
  AttachRuleToPGroup("ShowNewBooks", "browseBooks")
  AttachRuleToPGroup("PersonalizeDisplay", "smallScreen")
endWhen

```

We have introduced the presented approach in the context of a CAWE tool to design web applications called VisualWADE [6] (developed by the web engineering research group of the University of Alicante) that follow the OO-H design method presented in section 4. We have extended VisualWADE with new properties in the OO-H personalization framework to be able to gather the needed information to personalize web applications based on the criteria presented in section 3. Also we have incorporated a PRML parser to specify the acquisition rules, profile rules and personalization rules. The NAD and the PRML can be compiled by means of an advanced model compiler that was presented in [6]. The compilation process produces an XML default presentation that fulfils the conceptual specification.

7 Conclusions and future work

In this paper, we have presented a personalization framework for (general) personalization support during the web site design process, in the scope of the OO-H method. We have focused on the user model of the framework, and distinguished three different criteria upon which personalization support can be built: characteristics, user requirements and context. Based upon these three features, personalization rules can be defined. By means of a running example, we have shown how these rules can be specified using PRML, a high level language easy to use by web designers. Finally, we have discussed the VisualWade tool that

supports generating websites with personalization based on an OO-H design. Future work includes elaborating personalization based upon a fourth criterion: behavior. Although support for limited behavior based personalization is already present in our framework (i.e. based on browsing events), we intent to extend this capability, so that more complex navigation patterns can give rise to personalization.

References

1. Ceri S., Fraternali P., and Bongio A: "Web Modeling Language (WebML): a modeling language for designing Web sites", WWW9 Conf, 2000.
2. Dayal U.: "Active Database Management Systems", In Proc. 3rd Int. Conf on Data and Knowledge Bases, pp 150–169, 1988.
3. De Bra, P., Aerts, A., Houben, G.J., Wu, H: Making General-Purpose Adaptive Hypermedia Work. In Proc. WebNet World Conf on the WWW and Internet, AACE pp 117-123, 2000.
4. Franciscar, F., Houben G.J.: "Hypermedia Presentation adaptation on the Semantic Web", Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conf, AH 2002, Vol. 2347, LNCS, Springer, pp 133-142, 2002.
5. Garrigós, I., Gómez, J. and Cachero, C.: "Modelling Dynamic Personalization in Web Applications", 3rd International Conf on Web Engineering (ICWE), LNCS 2722, pp 472-475. Springer-Verlag, 2003.
6. Gómez, J., Cachero, C., and Pastor, O.: "Conceptual Modelling of Device-Independent Web Applications", IEEE Multimedia Special Issue on Web Engineering, pp 26–39, 2001.
7. Kappel, G., Retschitzegger, W., Poll, W., & Schwinger, W. "Modeling Ubiquitous Web Applications - The WUML Approach". In: Proc. of the International Workshop on Data Semantics in Web Information Systems, 2001.
8. Klyne, G., Reynolds, F., Woodrow, C., Ohto, H.: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies. W3C(2001), <http://www.w3.org/TR/CCPP-struct.vocab>
9. Nielsen, J.: Finding usability problems through heuristic evaluation.. In Proc. of the SIGCHI Conf on Human factors in computing systems. Monterey, California, United States pp: 373 – 380. 1992.
10. Schwabe, D. and Rossi, G. A Conference Review System with OOHD. In First Internacional Workshop on Web-Oriented Software Technology, 2001.