DESIGN AND IMPLEMENTATION OF ADAPTIVE WEB APPLICATION

Raoudha Ben Djemaa, Ikram Amous, Abdelmajid Ben Hamadou
ISIMS, LARIM, Route Mharza Km 1, BP 1030 – Sfax, Tunisie

Keywords: Adaptive web application, MCSD-case, user profile, Java Servlets.

Abstract: Web applications are now widely used for diffusing and processing information over the network. Methodological guidelines which assist web application developers in their task must take into account users' preferences to generate adaptable web interfaces. This paper presents MCSD-case, a generator of adaptive web application, which addresses the issue of designing and automatically deploying such application. MCSD-case's target is to facilitate the automatic execution of the design and the automatic generation of adaptable web interface. Using MCSD-case, designers can specify, at a conceptual level, the features of the web application to be generated. These features are made operational by instantiating different models dedicated to the description of the expected functionalities, the users and some features concerning adaptation. The implementation of MCSD-case is based on HTML and Java Servlets to instantiate the models which are translated in XML files then XSL files to generate the HTML page corresponding to the user aims.

1 INTRODUCTION

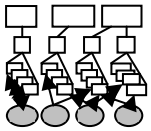
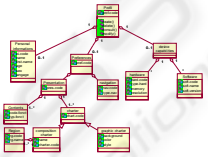
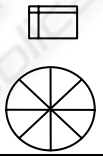
From its introduction in the early 90's the World Wide Web (WWW) is in a continuous development. Its rapid expansion results in an increasing number of Web applications being developed, especially with sources that contain frequently changing information such as databases (relational or object-oriented databases, XML repositories etc.)(Bieber et al., 1998). This leads to the fact that there is a higher need to automate, at least partially, the design process of web application. For this reason, there are many methods like RMM (Isakowitz et al, 1995), OOHDM (Schwabe and Rossi, 1998), WSDM (De Troyer and Leune, 1998), etc. These methods have been originally developed for a manual hypermedia design process; they are not particularly well-suited in the context of automated adaptive web application. In this context, efforts in the design of today's hypermedia system have to be pursued and, especially, to be concentrated on the *adaptation* issue.

Adaptation can be defined as the ability a system has to dynamically adapt itself to any user by giving her/him the feeling that the system has been designed especially for her/him. The need for adaptation arises from different aspects of the interaction between users and hypermedia systems. Users' categories which deal with these systems are increasingly heterogeneous due to their different interests and preferences. The web can be accessed through a number of different devices (PC, WebTV, WAP phone, etc.). Finally taking into account the user preferences (desired layout, navigation patterns, etc.) can lead to a more effective interaction. For this reason, in previous work (Djemaa et al., 2004), we have presented a user driven method for modeling adaptive Web applications, called MCSD. The different models of this method are: the functionalities model, the audience model, the adaptation model, the analysis model, the conceptual model and the navigational model.

In this paper, we will present MCSD-case, a generator of adaptive web application, which addresses the issue of designing and automatically deploying such application. The design approach supported by MCSD-case is presented in section 2. The prototype architecture is detailed in section 3. Then, we compare our approach and prototype with some related works in section 4. Finally, section 5 concludes the paper and suggests future research directions.

Ben Djemaa R., Amous I. and Ben Hamadou A. (2006).
DESIGN AND IMPLEMENTATION OF ADAPTIVE WEB APPLICATION.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - HCI*, pages 151-156
DOI: 10.5220/0002452101510156

Table 1: Design steps in MCSD-case.

N°	Step	Model	Figure
1	Analysis of the functional requirements of the web application	*Functionalities model* based on three concepts: functional class, functional role and Functionalities.	
2	Modeling of users	*Audience model* expressed through a UML class diagram generated by an algorithm of generation of Functional Spaces.	
3	Specification of the user's profile	*Adaptation Model* expressed as a specific UML class diagram.	
4	Description of the presentation	*Presentation model* composed of *Composition charter* and *Graphical charter*	

2 WEB APPLICATION DESIGN WITH MCSD-CASE

MCSD-case is an environment dedicated to the automatic generation of web application given some conceptual specifications. Its target is to facilitate the automatic execution of the design and the automatic generation of adaptive web interfaces. It should be possible to program the web applications in such a way that it can automatically execute the process specified by the designer. The design approach of MCSD-case is resumed in Table 1. Each of these steps is presented in the following section.

2.1 Requirements Analysis

Met with the increasing needs of users of Web applications, we propose *a functionalities model,* gathering users' informational and functional needs.

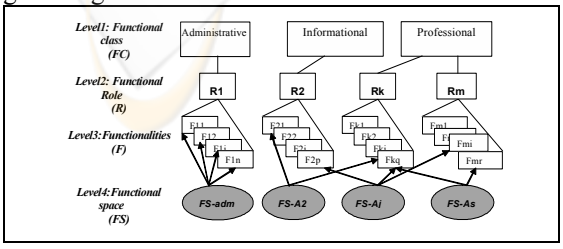


Figure 1: the functionalities model.

This model is to be looked in figure 1.

At the highest level, functionalities are classified into three classes, called *Functional Classes*, summarising the behaviour and the functioning mode of Web applications. These classes are abstracted by *Administrative, Informational* and *Professional* ones.

- The *Administrative* functional class gathers all the actions and the functionalities which take part in the configuration of the system management, relative to the system administrator.
- The *Informational* functional class gathers all the functionalities enabling the users to have access to the system to acquire specific information. This class translates the set of the static and dynamic informational users' needs.
- The *Professional* functional class is devoted to the representation of the actors' functional needs. Indeed, Web applications are based on technologies which make their contents dynamic, enabling, thus, the user to modify the applicative state of the server by carrying out a set of functionalities.

Each of these functional classes, represented above, will be subdivided into sub parts called *Functional Roles,* fixed by the designer.

At the following level, the "Functional Role" concept will be decomposed by the designer into a set of elementary functionalities representing both informational and functional needs. The choice of the decomposition parameters is left to the designer.

At the lowest level of the Functionalities Model, the concept "Functional Space" provides each actor with a list of authorized functionalities.

In our approach, the Functionalities Model implementation is carried out in two stages. During the first stage, the designer is invited to define the first three levels of the model. The fourth level, relating to the functional space of the actors, is approached only after having defined the list of the actors involved in the application. However, this list is defined in the following section.

2.2 Modeling of Users

Web applications are intended to be used by a variety of individual users, not necessarily presenting the same needs, or having the same rights of access to data. Users, in fact, show varied knowledge and experience, and express various preferences, etc. Therefore, the identification of the groups of actors, called in our approach *Audience class* (De Troyer and Leune, 1998), is a means of reducing complexity subjacent with such heterogeneity.

By definition, an audience class is a potential user group which belongs to the application target audience and which has the same informational and functional needs. These classes are not necessarily disjoined (a user can belong to several classes of audience).Nevertheless, despite belonging to the same audience class, the actors show individual characteristics sometimes varying to a subset of individual class or to a particular individual of this class. Thus, the system must be able to take into account various audience classes and their derived sub-classes.

In our approach (Djemaa et al., 2004), we have proposed an algorithm which generates the list of the actors of the application starting from the concept of Functionality. This algorithm makes it possible to specify all the classes of audience, to represent them in a hierarchy and to determine the concept "Functional Spaces" relating to each class. We have presented in (Djemaa et al., 2004), the process of operation of this algorithm.

2.3 Adaptation Model

A web application can be accessed through a multitude of devices and by different users. Each device has its own capabilities (display size, memory size, network speed, etc...). Every user has his specific preferences (desired layout, navigation patterns, etc...).

An intelligent web application needs to take into account these constraints (abilities) coming from both devices and users, and adapt the presentation preferences.

In fact, two types of adaptation or personalization are distinguished: adaptability (Atezni et al., 1997) and adaptativity (Frasincar and Houben, 2002). Adaptability (or static adaptation) means that the generation process of web interfaces is based on available information that describe the information about device capability and user preferences in which the user will use the generated presentation. Adaptativity (or dynamic adaptation) is the kind of adaptation included in the generated adaptive hypermedia presentation that itself changes while being browsed. In this paper, we focus on generating web intrfaces by using only static adaptation.

In order to specify adaptation in MCSD, we have extended our method by an adaptation model. In (Djemaa et al., 2006) we have presented different dimensions treated for this proposition. This model takes into account the user's devices capabilities (hardware and software), the user preferences presentation (desired layout, navigation patterns, etc.) and personal informations (e.g. Age, language, etc).

2.4 Presentation Model

The presentation model is intended to create the models of the Web pages. Two dimensions are considered:

1. the organization of the Web pages through the definition of composition charter (Djemaa et al., 2006).The composition charter describes the various regions of a page and their composition thus allowing homogenizing the site pages in terms of contents.
2. their appearance based on the specification of graphic charters (Djemaa et al., 2006). In fact, in order to adapt the presentation of the web application pages, we propose in MCSD-case, Java Servlets interfaces which require users information's concerning their preferred choices about:

− Visual characteristics: police (color, size, style), background (color or image).

− Navigation patterns: absolute or relative.

− Type of media used for each web page (representing functionality) and the corresponding joint file.

This information serve to instantiate the profile model which is exported by MCSD-case in XSL files (cf section 3).

The concept of charter is to be brought closer to a presentation model. Our objective is to guarantee coherence between various pages of the web application. This cohesion of the interface is, according to us, particularly necessary in the adaptive web application. The final step, before the generation of web interfaces, consists in filling the different models with information. This step is detailed in the following section.

3 ARCHITECTURE OF MCSD-CASE

The previous section dealt with the engineering process of MCSD-case methodology. This section focuses on the process of generation of adaptive web applications and describes how the system is dynamically adjusted to varying audience classes. The architecture of MCSD-case is shown in figure 3.

For a designer modeling an adaptive web application using MCSD-case consists firstly, in instantiating the functionalities model. Once instantiated, this model is translated into XML files to be stored in the data repository ❶ and the system execute the algorithm of generation of audience classes ❷ which built the audience model. The last is also translated into XML file ❸ which contains the functional space for each audience class.

At this stage, the content is adapted to each audience class. So, to adapt the user's presentation preferences, the designer is invited to instantiate the profile model using three java Servlets interfaces that treat respectively the following axes:

1. Personal information about the user like name, age, sex, language.
2. User's presentation preferences defined in term of web pages composition and graphical aspects specified by two charters called composition and graphical charters.
3. User's devices capabilities which define some characteristics of the hardware and software environment.

After instantiation, the profile model is exported in XSL files in the data repository❹. Finally, XML and XSL fies are extracted from the data repository ❺ and they are sent to the parser ❻ in order to publish the HTML page corresponding to the audience class❼. In figure 3 we show the interfaces of MCSD-case used to adapt the content of the web application represented by the steps of the instantiation of the functionalities model, the execution of the algorithm of the generation of audience classes and finally by the instantiation of the audience model to generate us results functional spaces for each audience class.

In figure 4, we show some results generated by MCSD-case used to adapt the presentation of the web application and we present the interface of administration of the prototype. At this stage, MCSD-case is able to assure the adaptability of the the content and the adaptability of the preferences presentation.

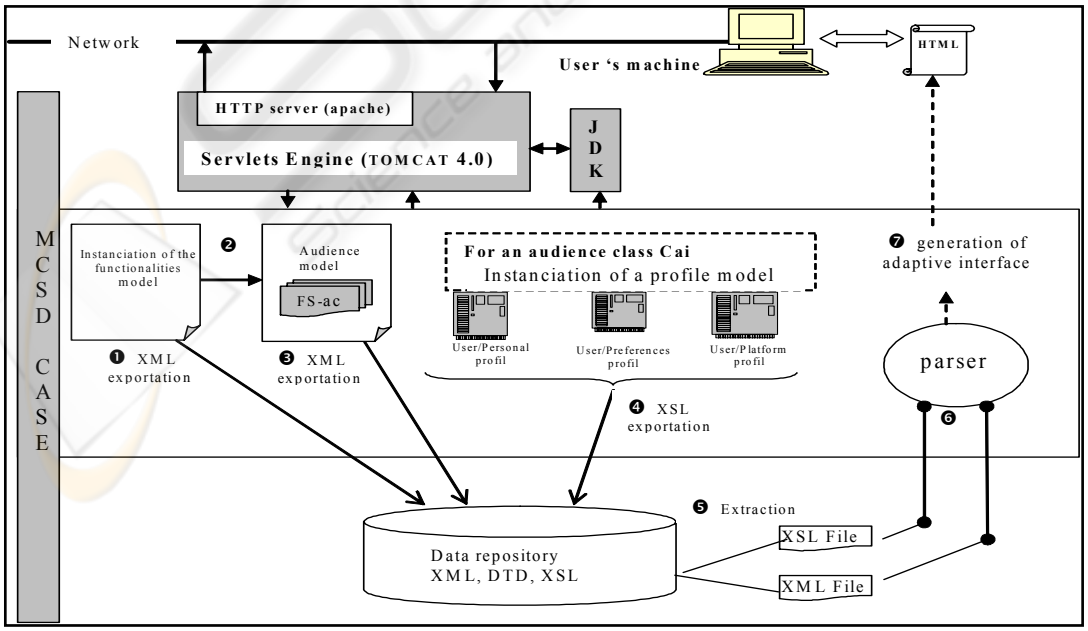
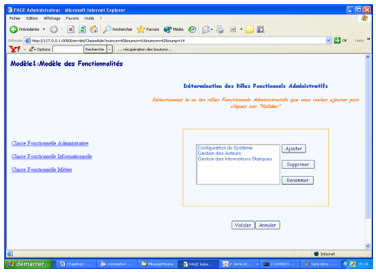
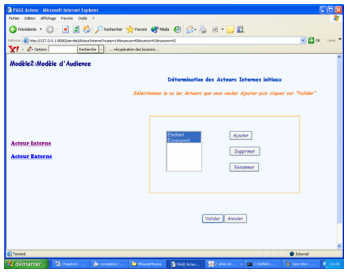


Figure 2: Architecture of MCSD-case.

a. Home page of MCSD-case

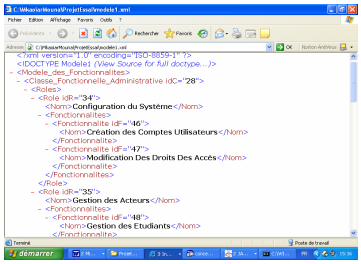


b. How MCSD-case instantiate the functionalities model?

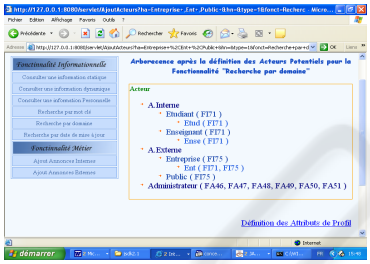


c. How MCSD-case instantiate the audience model?

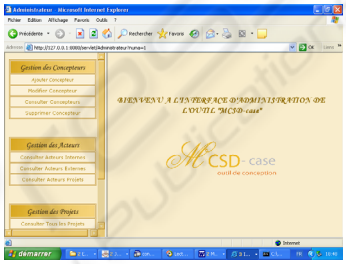
Figure 3: Adaptability of the content in MCSD-case.



d. XML file of the functionalities model



e. Result of the algorithm of generation of audience model



f. Administration of MCSD-case

Figure 4: Administration and results of MCSD-case.

4 RELATED WORKS

Most of the Web application design approaches proposed at the end of the 90's are inspired from earlier hypermedia design methodologies (RMM (Isakowitz et al, 1995) or OOHDM (Schwabe and Rossi, 1998)) which address content, presentation and navigation issues. These three dimensions are also taken into account in our approach, with a special emphasis put on adaptability to user. As a consequence, we propose an explicit modeling of users as opposed to Araneus (Atezni et al., 1997). WSDM (De Troyer and Leune, 1998) also represents users explicitly, but does not uses algorithm of generation of functional space for each user. The last concept allows us to adapt the content to user which constitutes a particularity of MCSD-case.

Adaptability also appears through the choice of a graphical charter, and through the composition of pages which can be define in MCSD-case for each audience class. The independence between content and presentation, notably allowed by the XML/XSL technology, promotes adaptability since it makes possible the association of different presentations for a same content. WebML (Ceri et al., 2000) and KIWIS (Villanova-Oliver et al., 2002) adopts a similar approach, but with lesser possibilities for user for instantiate his profile.

Similarly to MCSD-case, tools for Web application development support the steps of their associated methods, and generally perform the Web application generation. A particularity of MCSD-case is the ability of its use through the network.

5 CONCLUSION

In this paper, we have presented MCSD-case, a generator for adaptive web application. We have demonstrated how this generator can facilitate the automatic execution of the design and the automatic generation of adaptive web interfaces.

During the different activities that compose the engineering process of MCSD-case there are a lot of ontologies involved: semantic level (Functionalities model and audience model), adaptation level (profile

model) and presentation level (presentation details in terms of composition and graphic).

The implementation of MCSD-case is based on java Servlets interface to instantiate the models which are translated in XML files. MCSD-case uses then XSL files to generate the HTML page corresponding to the user. As future work, we mention:

– The extension of the MCSD-case implementation to take into account users' devices capabilities (hardware and software).

– The construction of a graphical tool that supports the presentation of the other models of MCSD and outputs XML files.

– The inclusion of the mecanism of the dynamic adaptation (adaptativity) in MCSD-case in order to dynamically elaborate and modify both the functional space and the navigation patterns, learning from the user's behavior.

REFERENCES

Bieber, M., Isakowitz, T., Vitali, F., July 1998. Web Information Systems. *In Communications of the ACM, 41(7),* pp. 78-80.

Isakowitz T., Stohr A. & Balasubramanian E., August 1995. RMM: A methodology for structured hypermedia design. *In Communications of the ACM 38(8),* pp. 34-44.

Schwabe D. & Rossi G., 1998. Developing hypermedia application using OOHDM. *In Proceedings of Workshop on Hypermedia development Process, Methods and Models,* pp. 85-94.

De Troyer O., Leune C., April 1998. WSDM: a User-Centered Design Method for Web Sites, *In Proceedings of the 7th International World Wide Web Conference (WWW7), Computer Networks*, 30(1-7).

Ben Djemaa R. Jmaiel M., Ben Hamadou A., Mai 2004. Vers une méthode centrée-acteur pour la conception des applications Web. *Eighth Maghrebian Conference On Software Engineering and Artificial Intelligence. Mcseai'04*.

Frasincar F. & Houben G-J., 2002. Hypermedia Presentation Adaptation on the Semantic Web. *In Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*, Malaga, Spain, May 29 - 31, , LNCS 2347, pp. 133-142.

Ben Djemaa R. Amous I., Ben Hamadou A., February 2006. GIWA: A generator for adaptive Web applications. In *ICIW'06, IEEE International conference on Internet and Web Application and Services.*

P.Atezni, G. Mecca and P. Merialdo, August 1997. To Weave the Web, 23rd Conf. *On Very Large Databases (VLDB'97)*, pp 206-215, Athens, Greece.

Ceri, S., Fraternali, P., Bongio, A., May 15-19, 2000. Web Modeling Language (WebML): a modeling language for designing Web sites. *9th WWW Conference, Amsterdam*.

Villanova-Oliver M., Gensel J., Martin H., Erb C., April 8-10, 2002. Design and Generation of Adaptable Web Information Systems with KIWIS. *3th IEEE Conference on Information Technology ITCC-2002*, Las Vegas, USA.

R. Vdovjak, F. Frasincar, G. J. Houben, and P. Barna. Engineering semantic web information systems in hera. *Journal of Web Engineering, Rinton Press*, 2(1&2): pp 003--026, 2003.

Fiala Z., Hinz M., Houben G.J., and Frasincar F., March 2004. Design and Implementation of Component-based Adaptive Web Presentations. *Sac'04*, Nicosia, Cyprus.

Brusilovsky P., 2001. User Modeling and User-Adapted Interaction, *Ten Year Anniversary Issue 11 (Alfred Kobsa, ed.)*, pp 87-110.

Fiala Z., Hinz M., Meissner K., and Wehneer F., September 2003. A Component-based Component architecture for adaptive dynamic Web documents. *Journal of Web Ingeineering, Rinton Press, 2(1&2):* pp 058-073.

Henze, N., and Nejdl, W*., AH* 2002. Knowledge modeling for open adaptive hypermedia. *In Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*.

Rossi G., Schwabe D., and Guimaraes R., Designed personalized Web application. *In WWW10, The Tenth International Conference on the world Wide Web*, Hong Kong, 2001.