APPLYING QUALITY HYPERMEDIA DESIGN PRINCIPLES TO A WEB-BASED EDUCATIONAL SYSTEM

Roberto F. Arroyo, Nuria Medina, Miguel J. Hornos and Fernando Molina
Dept. Lenguajes y Sistemas Informáticos, E.T.S.I. Informática y de Telecomunicación, University of Granada, Spain

Keywords: Web-based educational system, hypermedia design, user adaptation.

Abstract: We present a web-based educational system modeled using hypermedia design principles. The design establishes two interrelated and interacting layers: a data storage layer and an adaptation layer, linked by the user model. The data storage layer is structured in a data level and an object level. The first level is responsible for maintaining a relational representation of all data, and the second one converts data from the relational schemes into object oriented structures, used out of the storage layer. The adaptation layer performs an adaptive process in two levels: contents and presentation. The former decides what must be shown and depends on the user's capabilities, while the latter determines how it must be shown and depends on the user's preferences. Moreover, the system uses dynamic role assignments based on users' capabilities.

1 INTRODUCTION

Hypermedia systems, and web systems, considered as the subset of hypermedia systems that operate in specific Web environments (Díaz, 2005), structure the information in conceptual units that are interrelated by means of links, so the selection of a link causes the retrieval of the information contained in the destination unit. This associative organization of the information is similar to the one used by the human mind in order to interrelate memories (Rada, 1991) and has its origin in the paper presented by Bush (1945). Nowadays hypermedia can be considered to be a mature technology in whose theoretical bases the scientific community is still working.

If we focus on theoretical models, we must cite: HAM model (Campbell, 1988) as the first attempt to express a hypermedia system by means of an abstract model; Trellis model (Furuta, 1990), which defines five logical levels and establishes the need of separating the structure from the content elements that will be associated to such structure; and Dexter model (Halasz, 1990), which explicitly separates in different layers the storage and presentation of the information.

Regarding design methods we can find, among others: OOHDM (Schwabe, 1996), which adapts object oriented notation and abstraction mechanisms, and establishes four steps of design: conceptual, navigational, abstract interface and implementation; WSDM (De Troyer, 1998), where data is modeled from each user's point of view in three steps: user modeling, conceptual design and implementation; WebML (Ceri, 2000), oriented to the design of web sites with a large amount of data, where a web site is defined as the combination of structure, composition, navigation and presentation; UWE (Koch, 2001), an object oriented approach based on UML and on the unified process that takes into account user adaptation; OO-HMethod (Cachero, 2000), which extends conventional object oriented models with two new diagrams that specify navigational access and abstract presentation and which, in addition, uses a catalogue of interface patterns; and ADM (Díaz, 2005), which establishes a user-centered iterative process, divided into three phases with a different abstraction level: conceptual design, detailed design and evaluation.

Focusing in the adaptive capabilities of the system, many authors propose models or architectures with the aim of developing a hypermedia system that is able to adjust both the presentation of contents and the way their link structure is navigated to the needs of each user, therefore reducing the problems of disorientation and cognitive overhead usually associated to hypermedia systems. For example, we can cite AHAM model (De Bra, 1998), AHA architecture (De Bra, 2003), or the model and architecture SEM-HP (Medina, 2005).

F. Arroyo R., Medina N., J. Hornos M. and Molina F. (2007).
APPLYING QUALITY HYPERMEDIA DESIGN PRINCIPLES TO A WEB-BASED EDUCATIONAL SYSTEM.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Society, e-Business and e-Government / e-Learning*, pages 458-463
DOI: 10.5220/0001285504580463
Copyright © SciTePress

This brief review shows that quality in hypermedia applications is a growing concern in the present scientific and technologic community. It is also known that the quality in processes affects the quality in the products, and that the quality in the products influences the usage quality (Olsina, 2005). In this way, the concept of Web Engineering arises as the application of a systematic, disciplined and measurable approach to the development, operation and maintenance of hypermedia systems and applications (Lowe, 1999). In addition, specific methodologies and tools to measure the quality of this kind of applications arise, such as WebQEM and WebQEM-Tool (Olsina, 2002).

Nevertheless, in spite of all the efforts done in the scientific environment, the reality is that, nowadays, there are still web applications that are built without taking into account sound design principles and user adaptation. In this sense, in this paper we present our experience in the design of a web-based educational platform, in which initially these aspects were not taken into account. After a period of exploitation and subsequent evaluation, we decided to reimplement the system, this time grounded on quality hypermedia design principles.

This paper is organized as follows: Section 2 presents the principles and context regarding the first implementation of the platform. Section 3 shows the architecture and design principles followed in the development of a new version of the system. Finally, last section presents the conclusions.

2 WORK CONTEXT

In the academic year 2003/04, a group of lecturers at the University of Granada decided to create, in the context of an Educational Innovation Project, a web-based system to support the teaching and learning of the subjects we taught and the management of academic information related to them (Hornos, 2006), but our aim was not to build an e-learning tool, like the existing ones, since in this case we would have opted by one of them. This system, called Tutor (http://tutor.ugr.es), was initially used by about one thousand students and about twelve teachers, and was designed with different free access sections, such as: *Introduction*, with general information about the system; *Staff*, with information about the development team and the teachers using it; *Subjects*, with detailed information about the syllabus, timetables, exam calendars, etc. regarding the subjects managed by the system; *Links*, selected and classified into categories by the teachers, leading to interesting web pages for their students; *Notice Board*, where the administrator or the teachers can place notices or news related to the platform or to any of the subjects managed by it; *Suggestions*, which allows the users to communicate problems or suggestions related to the platform, its subjects and/or teachers, being each received communication automatically forwarded to the most appropriate recipient. In addition, there is a *Restricted* section, which can be used only by registered users that authenticate by properly logging in. This section has more functionality than the others, and provides the information that requires a higher level of privacy. For example, students can register into the laboratory group in which they are most interested, consult their marks, update their personal information, download didactic materials, upload solutions to assignments, etc. Teachers can create theoretical and laboratory groups, upload materials to the download area, propose assignments, control the attendance to classes, and set marks, among many other functions.

Regarding users, four different profiles were created: one for *non authenticated users*, in which the user (who is not registered in the system or has not logged in) can navigate in any section except the Restricted section, and three authenticated user profiles, corresponding to *administrator*, *teacher* and *student*, each of which gives access to the appropriate restricted section. All the registered users can use several communication mechanisms such as: internal mail, discussion forums, chat, IP telephony, instant messaging and access to restricted notices. The shown elements and the set of available actions for a user not only depend on the user profile, but also on the role the user is performing with regard to a specific subject. The system distinguishes between the roles of theory teacher, laboratory teacher and subject coordinator in the teacher profile, and between theory group student and laboratory group student in the student profile.

The number of users and subjects in Tutor increases each year. This academic year the platform is used by 45 teachers and 32 subjects, corresponding to 11 different degree courses, counting more than 5500 registered students. This increase requires a considerable extension of the system functionality, because of specific needs that new subjects and users have. For example, the addition of subjects with a high number of foreign students from the Socrates/Erasmus exchange programme has made us to consider that Tutor provides information in more languages than only Spanish.

The system we initially developed was adequate for a limited number of users and subjects, but, due to its design, its adaptation to such a big growth (in users, contents and specially in functionality) requires an excessive effort. This is mainly because the aspects of information storage and navigation were not separated, a formal user modelling was not performed, and mechanisms to carry out a satisfactory user adaptation were not defined.

For all these reasons, we have decided, in the context of a second Educational Innovation Project, to develop a new version of Tutor which, in addition to improving the user interface and navigation system, allows adaptiveness and adaptability.

3 DESIGN AND ARCHITECTURE

The new design, which has been done taking into account the specific directives of hypermedia systems, encompasses most of the features proposed by the methods reviewed in the Introduction section: object orientation, distinction between conceptual and navigational modelling, user modelling, user adaptation, and separation of the information storage from its subsequent presentation. Moreover, the performed development process follows the basic principles proposed in the SEM-HP model (Medina, 2005), adapted in each case to the specific features of Tutor. In addition, the final design tries to provide an adequate navigability, not forgetting the need of an easy and fast upgrading and extensibility of the system, since it is predicted that the new system will grow even more that the initial version. It also provides specific user adaptation mechanisms, and a simpler and more intuitive user interface.

The previous design was oriented to the role (or user profile), directing both the way in which the data was stored and the form of presentation and navigation to the role played by the user navigating the system, independently on the part of the system being visited. The new design completely changes this orientation, proposing an object oriented design (data and operations), where it is also important what is being visited, and not only who visits it. The first-level objects we have considered include the system users, subjects, degree courses, groups, schools and departments, as well as notices and internal messages. All these objects constitute the system navigability core, and are the main information structure of the system, so their design deserves a major consideration. The new design is divided into two layers (see Figure 1): *data storage* and *adaptation* to each user. The connection between them is the *user model*, which stores, maintains and updates the user's main features. In the following subsections we describe in detail the different parts of this design, commenting the objectives we want to achieve.

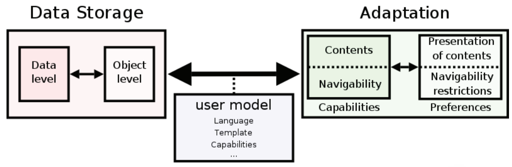

Figure 1: Model based in a clear separation between information storage and display, linked by the user model.

3.1 Data Storage Layer

Data storage is structured in two levels: a relational level and an object oriented level. We store all the data in a set of relational tables at the least abstract level of the information storage, called *data level* (see Figure 1), and use an object-relational mapping (Stonebraker, 1999) for linking data in relational databases to object-oriented language concepts. We opted to code our own tailored object-relational mapping instead of using an existent package that performs this technique.

In addition, we provide a more abstract level inside the data storage layer (see Figure 1): the *object level*, which gives an object oriented view using the underlying relational database. This level builds an *extended object* from the information stored in the data level, taking into account the information associated to that object.

The adaptation layer, which always works with objects, transmits its requests to the object level, which is in charge of converting object oriented requests to relational queries, returning an answer consisting on an extended object constructed from a set of relational data. Figure 2 shows the scheme of this conversion, where the conversion module hides the relational nature of the storage. This allows designing the adaptation layer independently from the data storage support.

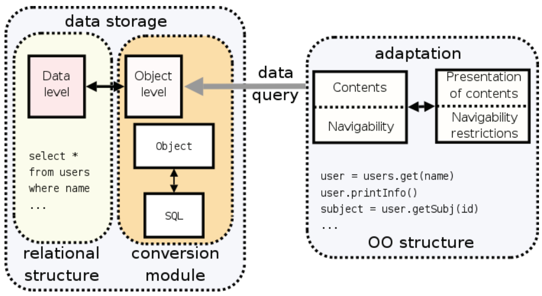

Figure 2: Scheme of our object-relational mapping.

3.2 User Model

The user model, which links the data storage and the user adaptation layers (see Figure 1), establishes some basic patterns that affect the final presentation of the information. The user model is used to control several options, such as the language in which the pages are displayed or the general aspect of the user interface. It also captures the user's capabilities, which are fundamental for the adaptation phase. A capability is an ability or responsibility associated to a user, which allow the user to play roles and to do tasks, subactivities or actions.

Among other things, the user model stores the following items: the language in which the information must be displayed; a customized template to change the general aspect of the platform; the capabilities which cannot be determined automatically by the system using the stored information about the users; special accessibility needs, like a high contrast color scheme; and other preferences, like maximum width, text-only representation, etc.

When the information is obtained with a personalized format, the users feel more comfortable, and it is easier for them to understand the provided information. Tutor allows this, since, for example, the page in Figure 3 shows the contents in the middle, with two side-bars containing a calendar and quick links to interesting sections for that user, but users can determine, through their user model, if they want only one side-bar (as Figure 4 shows), and choose its contents and aspect.

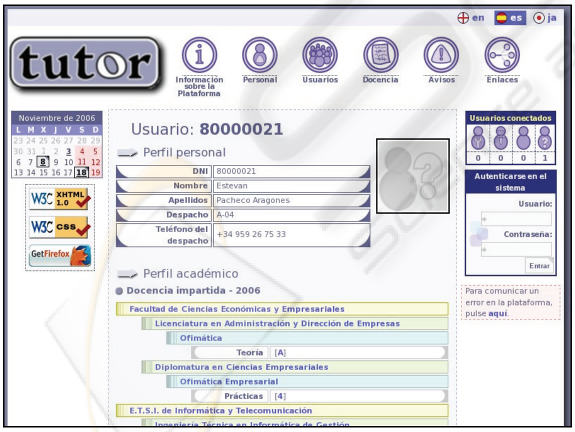

Figure 3: Tutor's user interface showing a three-column purple colour scheme, in Spanish.

Templates regarding user's capabilities or the selected language are not stored in the user model. This kind of templates is stored in the data storage layer, as they are common for every user with the same capability or language, unlike the custom template, which could be unique for every user.

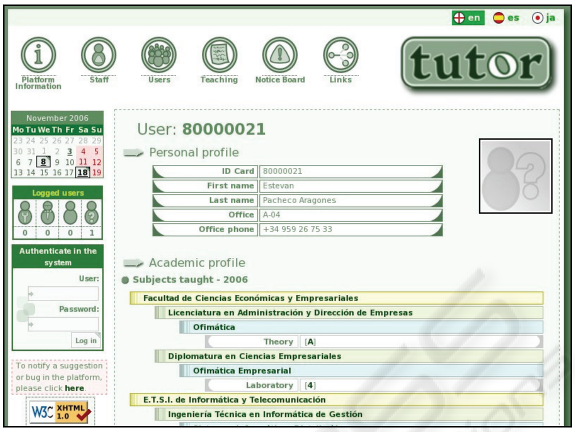

Figure 4: Tutor's user interface showing a two-column green colour scheme, in English.

Our objective is to provide all the contents in any of the available languages. Therefore, it is not enough to translate only the system labels, but we have also to translate the contents stored in the database into different languages. This makes us to add a localization system in the database, which is done in three phases. Firstly, the system tries to obtain the information in the language requested by the user. Secondly, the parts that are not available in that language are requested in the language established in the user model. Thirdly, if there are parts that are still not available, the system will obtain them in the platform default language.

3.3 Adaptation Layer

This layer is separated into two levels, as Figure 1 shows. The first one, where the contents and navigability are decided, depends on the user's capabilities, while the second level, where the presentation of the contents and the additional navigation restrictions are established, depends on the user's preferences. Consequently, the adaptation of the information to the user includes two phases, where what must be shown and how it must be shown is respectively decided. What must be shown is determined mainly by the user's *capabilities* regarding the item being accessed. These capabilities not only determine the shown information but also the allowed navigation, since they select which contents and related items must be accessible. For example, all the users can visit any subject stored in the system, but depending on their capabilities they could perform different actions on it and access different information about it. How the information is shown is determined by the user's *preferences*. Additionally, these preferences restrict the navigability, avoiding an excess of unused links and promoting a set of more useful links according to the user's criteria. For example, the teachers could choose their preferred language (for displaying the information) and set the actions they frequently use on a quick bar, hiding the least used ones.

The information objects are common to all the users in the platform, but both the operations the user can perform and the degree of completeness of the extended object (which can be partially built) are determined by the user's capabilities. The roles and user profiles present in the previous version do not exist now in the same manner. Therefore, the users do not have to choose which profile they want to use when authenticating, as it was done in the previous version, but they have just to log in, and the system will check their capabilities regarding each visited item, and it will assign them the corresponding roles according to these capabilities.

Our platform uses the features of W3C's Cascade Style Sheet (CSS), along with a set of interchangeable templates and a localization system that facilitates the adaptation to the user's preferences. This allows, for example, to define templates with bigger fonts and high contrast colours in order to make easier the accessibility for users with sight deficiencies. These templates also contain keywords that will be replaced either by more complex structures or by the final contents.

The adaptation phase is based on what has previously been established in the user model. As Figure 5 shows, adaptation is progressive. A multi-template system is used to refine the user adaptability. Initially the system employs a template, taken as the default template, which is replaced, modified and/or complete in accordance with the user model. Afterwards, the system completes the resulting template with fragments relative to the user's capabilities. Finally, details about the user's preferences are changed or added, and the template is filled in with the corresponding information, before the resulting web page is ready to be shown.

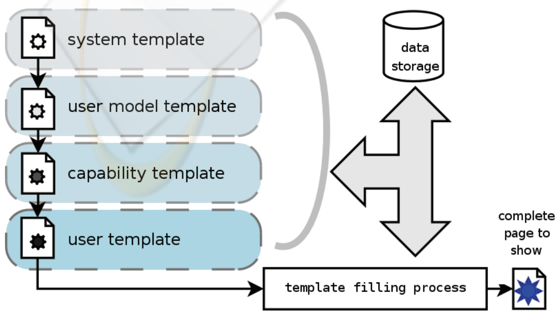

Figure 5: Adaptation based on template composition.

The presentation of information can be carried out in up to three definition phases: firstly, a default configuration is established, which is initially used; secondly, the system determines the modifications to apply according to the user's capabilities regarding the visited element; and thirdly, the user chooses the final details about how the selected information must be displayed.

This division of concepts allows an expandable and adaptable control of navigability, with a clear separation between the available information and how it has to be shown to a specific user.

4 CONCLUSIONS

We have presented the design of a web-based educational platform as an example of applying quality design principles to a hypermedia system. The evaluation of the first version of the platform showed that it lacked the essential groundings to be easily expandable and adaptable to users, being necessary to spend a considerable effort to add new functionalities. In addition, our experience showed the need of a more intuitive navigation system, and that the orientation to roles, not taking into account the visited information, did not contribute to improve the user's navigation.

The main problems detected in the previous version have been solved with the new design, while its positive aspects have been kept. The change of perspective from role orientation to object orientation allows a more intuitive and controlled navigation, with a better structured presentation of the information. This permits an easier expansion and upgrading of the platform to new requirements, being at the same time more scalable, not negatively affecting the provided functionality. The active role of the system in taking decisions has also been increased, replacing the manual selection of the user profile with an automatic query of the user's capabilities while the user visits any page of the system.

The new design establishes two interrelated and interacting layers: the data storage layer and the adaptation layer. The link between both layers is the user model. The data storage layer is structured in two levels: data level and object level. The former maintains a relational representation of the information and communicates with the DBMS, while the latter builds extended objects in accordance with the existing relations between data and communicates with the adaptation layer. The adaptation layer performs an adaptive process in two levels, deciding respectively what to show (i.e. the contents and navigability) and how to show it (i.e. the presentation of the contents and the navigability restrictions). The first level depends on the user's capabilities, while the second one depends on the preferences established by each user.

Consequently, this new design, which is based on the aspects proposed by the majority of hypermedia design methods and follows the basic principles of SEM-HP model, has shown to be more robust and future-proof than the previous design, which was done nearly ad-hoc, according to a traditional methodology that did not take into account the hypermedia features of the system.

The implementation of the new version of the platform is in an advanced state, but not finished, so we are currently working in order to finish its development, with the aim of replacing the previous version. We also plan to increase the functionalities of the new system, extending the available tools with new options and operations, and adding new tools to the ones already included in the platform at the request of users.

ACKNOWLEDGEMENTS

This work is financed by an Educational Innovation Project from the University of Granada (code 05-03-46) and by a Spanish CICYT Project (code TIN2004-08000-C03-02).

REFERENCES

Bush, V. (1945). As We May Think. *Atlantic Monthly, 176*, 101-108.

Cachero, C.; Gómez, J.; Pastor, O. (2000). Object-Oriented Conceptual Modeling of Web Application Interfaces: the OO-HMethod Abstract Presentation Model. *Electronic Commerce and Web Technologies (EC-Web)*, 206-215.

Campbell, B.; Goodman, J. (1988). HAM: A General Purpose Hypertext Abstract Machine. *Commun. ACM, 31(7)*:856-861.

Ceri, S.; Fraternali, P.; Bongio, A. (2000). Web Modelling Language (WebML): A Modelling Language for Designing Web Sites. *Conference WWW9/Computer Networks 33 (1-6)*, 137-157.

De Bra, P.; Aerts, A.; Berden, B.; De Lange, B.; Rousseau, B.; Santic, T.; Smits, D.; Stash, N. (2003). AHA! The Adaptive Hypermedia Architecture. *Proceedings of the fourteenth ACM Conference on Hypertext and Hypermedia*, 81-84.

De Bra, P.; Houben, G.J.; Wu, H. (1998). AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. *InfWet'98 Conference*, 51-76.

De Troyer, O.; Leune, C. (1998). WSDM: A User-Centered Design Method for Web Sites. *Computer Networks and ISDN systems. 7th International World Wide Web Conference*. Elsevier, 85- 94.

Díaz, P.; Montero, S.; Aedo, I. (2005). *Ingeniería de la Web y Patrones de Diseño*. Prentice Hall (in Spanish).

Furuta, R.; Stotts, P.D. (1990). The Trellis Hypertext Reference Model. *NIST Hypertext Standardization Workshop*, 83-93.

Halasz, F.; Schwartz, M. (1990). The Dexter Hypertext Reference Model. *NIST Hypertext Standardization Workshop*, 95-133.

Hornos, M. J.; Abad, M. M.; Hurtado, M. V.; López-Sánchez-Huete, M. I.; Molina, F.; Muñoz-Ropa, A.; Martínez, J. J. (2006). Web-Based Learning-Support System for Computer Science Applied to Business. *International Journal of Learning, vol. 12, nº 11*, 145-155.

Koch, N.; Kraus, A.; Hennicker, R. (2001). The Authoring Process of the UML-based Web Engineering Approach. *Proceedings of the 1st International Workshop on Web-Oriented Software Technology*. http://www.dsic.upv.es/~west2001/iwwost01//WWOSTContent.htm.

Lowe, D.; Hall, W. (1999). Hypermedia and the Web. An Engineering Approach. *John Wiley & Son*.

Medina, N.; Molina, F.; García, L. (2005). Diversity of Structures and Adaptive Methods on an Evolutionary Hypermedia System. *IEE Proceedings – Software, vol. 152, nº 3*, 119-126.

Olsina L.; Rossi G. (2002). Measuring Web Application Quality with WebQEM. *IEEE Multimedia, vol. 9, nº 4*, 20-29.

Olsina, L. (2005). Conferencia: Una Estrategia de Evaluación de Aplicaciones Web. *I Jornadas de Ingeniería Web* Retrieved October 28, 2006, from http://www.informandote.com/jornadasIngWEB/programa.asp (in Spanish).

Rada, R. (1991). *Hypertext: from Text to Expertext*. McGraw-Hill.

Schwabe, D.; Rossi, G.; Barbosa, S. D. J. (1996). Systematic Hypermedia Application Design with OOHDM. *Hypertext '96. Proceedings of the Seventh ACM Conference on Hypertext*, 116-128.

Stonebraker, M.; Brown, P.; Moore, D. (1999). Object Relational DBMS: Tracking the Next Great Wave. *Morgan Kaufmann*.