Building dynamic WAP applications using UWE

Ricardo Soto D., Nibaldo Rodríguez A., Wenceslao Palma M.
School of Informatic Engineering, Pontifical Catholic University of Valparaíso, Chile.
E-mail: [ricardo.soto, nibaldo.rodriguez, wenceslao.palma]@ucv.cl.
Av. Brasil 2241 Fax: 56-32-273859 Casilla 4059, Valparaíso, Chile.

Abstract

The great advance of the communications and the importance gained by the mobile systems have increased the development of WAP (Wireless Application Protocol) applications, nevertheless at the present time do not exist methodologies specifically designed for the WAP applications development process.

However, in the last years, several methodologies for the web applications development process have been proposed. One of them is UWE (UML-based Web Engineering) [1]. Although this methodology has specifically been developed for web applications, the similarity between web and WAP systems allows us to propose the use of it, to generate a design stage approach for WAP applications and then to create a specific extension of UWE to build WAP applications.

In order to obtain this, we present a new UWE based model called WAP navigational model for the navigational design and presentational design steps, which we can use to design WAP systems improving the WAP applications development process.

Keywords: *Software Engineering, Web Engineering, Web Systems, Hypermedia, UML, WAP.*

1. Introduction

The great advance of the communications and the demanding people needs have increased the hypermedia field. New areas, that require the flexibility of mobile systems, have been born during the last years. However, the hypermedia applications development is extremely complex. Moreover it exist few methodologies appropriately to this type of software, driving to the developers to the omission of the application structural design. This big trouble generally gives a low quality application and makes it susceptible of later corrections. Without exaggerating, the maintenance stage continues being a problem. Not to have the suitable documentation of the application means to transform this process into an exhausting task.

The solution of these problems starts with the creation of a suitable task planning before the application construction. In order to obtain this appears the necessity to define development methodologies that use models and formal design structures, specially oriented to hypermedia software.

The hypermedia system development process usually is done using tools that support only the implementation stage, neglecting the important previous process of analysis and design of the structural navigation and interface aspects. However, in the last years, several methodologies for the web development process have been proposed, some of them are: UWE [1,2,3], HDM (Hypertext Design Model) [4], EORM (Enhanced Object Relationship Model) [5], RMM (Relationship Management Methodology) [6], OOHDM (Object Oriented Hypermedia Design Method) [7], Conallen [8], WSDM (Web Site Design Method) [9], that consider a design previous to the system construction and offer a group of techniques, to gather, in different abstract models the specifications from the hypermedia system to develop.

Now, due to the importance gained by the mobile systems like cellular phones and PDAs (Personal Digital Assistant), it seems coherent to us transfer this approach to the WAP applications field, where are a great amount of weaknesses, since formal models to design WAP applications practically do not exist. Nevertheless, the similarity between web and WAP systems allows us to propose the use of UWE, to generate a design stage approach for WAP applications and then to create a specific extension of UWE to build WAP applications.

This paper is structured as follows: First, Section 2 presents an overview of the UWE methodology, using a study case. Section 3 presents a design stage approach for WAP applications. Section 4 a sketch of the study case implementation, and finally presents some concluding remarks and an overview of future work.

2. UWE Methodology Overview

UWE is a development process for web applications specially focused in the systematization, personalization and semiautomatic generation of applications [10]. In this methodology graphical UML notations with other own ones are combined to design robust applications in three stages. Figure 1 shows the three design stages of UWE.

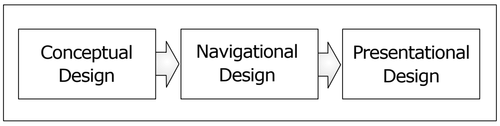

Figure 1 *The three design stages of UWE.*

In order to explain in a better way the use of this methodology, a WAP application called *Teaching WAP System* will be used. The goal of this application is to allow teachers and assistants to publish news and tutorials for its students for each one of the courses that are being taught. This system will be designed using UWE and it will be constructed for mobile systems using wml, wmlscript and php.

2.1. Conceptual Design

The conceptual design result is the conceptual model of the problem domain. To obtain it, it will be necessary to identify classes, attributes, methods and relations. Therefore, by means of well-known object oriented techniques such as, compositions, specializations and generalization a logical structure able to represent in correct way the problem domain it will be defined. Each element identified in this step will be used in the following one, the defined classes and associations will be the base to create the nodes of the navigation structure and the relations will be used to derive the links among them [11].

It is important to consider as an essential pre-condition to get an appropriate conceptual design, a careful analysis, ideally done using use cases and scenarios [12]. The conceptual model of the problem domain is shown in Figure 2.

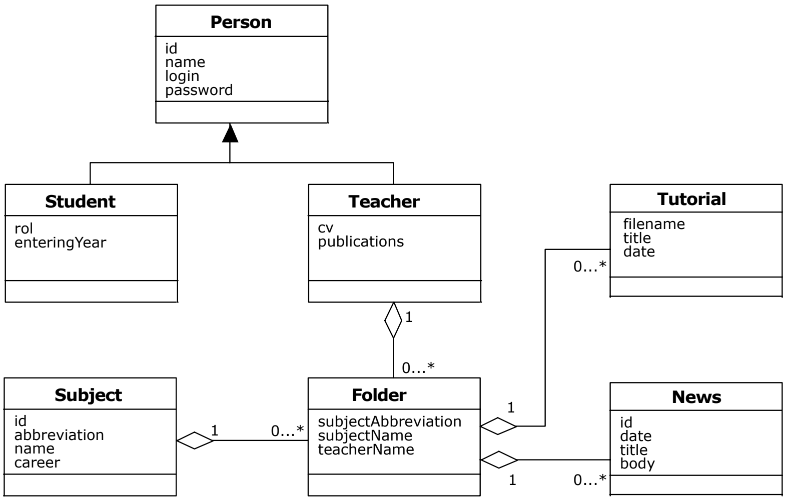

Figure 2 *Conceptual Model of the Teaching WAP System.*

2.2. Navigational Design

Navigational design is a critical step in the design of hypermedia applications. One of the main difficulties is to establish a navigational structure that allows the user to navigate in an intuitive way and to avoid him losing the orientation. In order to obtain this, it is necessary to design a hierarchical navigation structure, being careful with the nodes and navigation among them. Moreover, we must remember that the navigation model is a valuable document of the maintenance step.

The navigational design proposes to build two models: The navigational class model and the navigational structure model.

2.2.1 Navigational Class Model

The navigational class model defines a view on the conceptual model obtained in the previous step, where each class of the conceptual model can be visited through navigation in the application. The model is constructed from a set of navigation classes and associations among them. To build this model it is necessary to define which classes of the conceptual model will be part of the navigational class model, these classes will have to take the stereotype <<*Navigational Class*>>. The irrelevant classes for the navigation of the application can not be included in this model or simply reduced to attributes of another navigation class. Navigation among navigation classes will be obtained from the associations among classes of the conceptual model, for each link; there will be distinguished a navigational source object and a navigational target object.

In our study case, class Person and class Student are not relevant for the navigation of the application. Therefore attributes of the class Person have been added to the navigation class Professor. Attributes derived from eliminated classes are denoted by a slash (/) before its name. Figure 3 shows the attributes derived from eliminated classes [11].

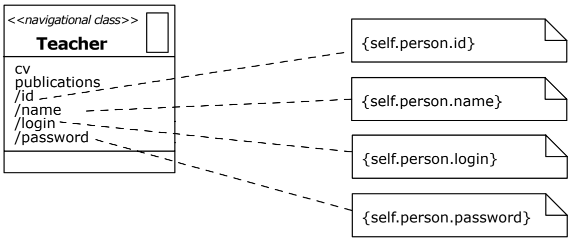

Figure 3 *Derived attributes from eliminated classes.*

Due to the exclusion of class Person and class Student the model of navigation classes results simpler. Figure 4 shows the Navigational Class Model.

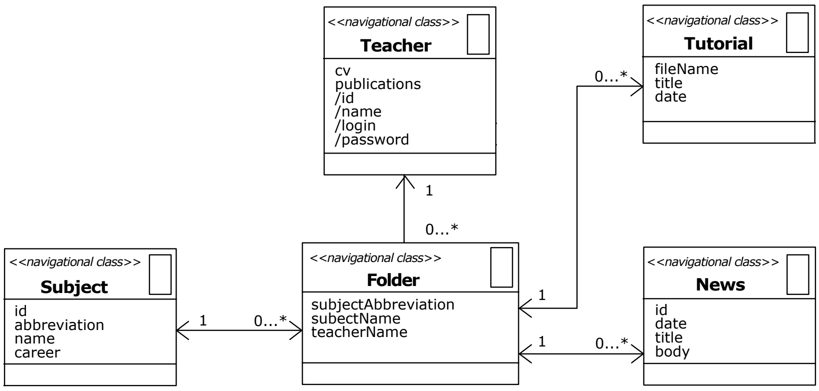

Figure 4 *Navigational Class Model of the Teaching WAP System.*

2.2.2 Navigational Structure Model

The navigational structure model is based on the navigation class model. It defines the navigation structure of the application and how navigation nodes and contexts [13,14,15] will be visited. A navigation context is formed by a set of navigation nodes that have a common condition. OOHDM introduced this concept to allow different groupings from navigation objects, this way a navigation object can be accessed from different navigation contexts.

To define clearly this model, it is necessary to specify the way in which the navigation nodes and contexts will be accessed, for that purpose; we will use access primitives, such as: indexes, guided tours, queries and menus:

- An index allows a direct access to each element within a navigational context.
- Tour guided allows a sequential access, step by step, to the elements of a navigation context. Tour guided can be controlled by the user or the system.
- A query is a set of elements as result of a SQL sentence executed to the data base.
- A menu is an index of a set of navigational nodes.

Figure 5 shows the stereotypes for the Navigational Structure Model.

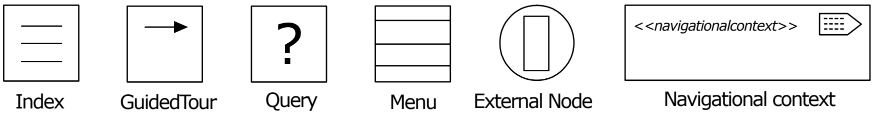

Figure 5 *Stereotypes for the Navigational Structure Model.*

In our study case a main menu is the starting point, from this it is possible to access to a teachers list and a subjects list separated by career (Informatics Engineering-4 year career- and Informatics Engineering-6 year career-). From the index of teachers we access to all the subjects taught by the selected teacher. From the index of subjects we access to all the teachers who teaches the selected subject. Once chosen the subject by teacher or the teacher by subject, we can enter to the navigation contexts of the class folder. Then, located in that place we can access to the news and the tutorials of the folder. Figure 6 shows the navigational structure model of the Teaching WAP System is shown.

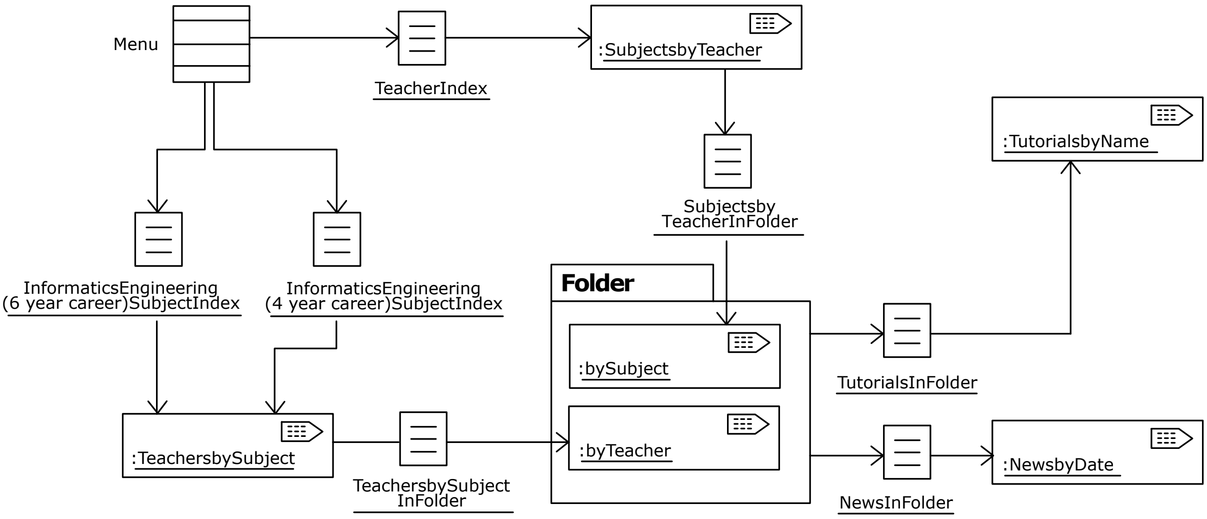
Menu
TeacherIndex
:SubjectsbyTeacher
:TutorialsbyName
Subjectsby TeacherInFolder
InformaticsEngineering (6 year career)SubjectIndex
InformaticsEngineering (4 year career)SubjectIndex
Folder
:bySubject
:byTeacher
TutorialsInFolder
:TeachersbySubject
TeachersbySubject InFolder
NewsInFolder
:NewsbyDate

Figure 6 *Navigational Structure Model of the Teaching WAP System.*

2.3. Presentational Design

The presentational design gives an abstract representation of the final interface and defines the interaction between the user and navigational nodes. In order to obtain this, UWE proposes to build two diagrams: The static presentational model and the dynamic presentational model.

2.3.1. Static Presentational Model

The static interface model gives an abstract model of the user interface. It allows to show the way in which the navigational structure is presented to the user. The interface model is a sketch of the final interface, decisions about details like sizes, colors, fonts or other specific objects of interface are not considered, because they belong to the implementation stage. In order to obtain this representation, UWE propose, to use user interface object such as: *presentational objects, anchors, buttons, texts, forms, images, audio, video, anchored collections and collections*. Figure 7 shows the user interface objects.

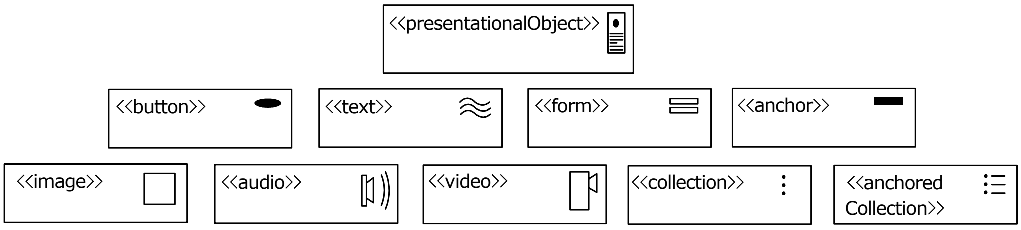

Figura *7 User interface object proposed by UWE.*

Figure 8 shows three interfaces of our system, in which some of the objects defined in figure 7 are identified. The Menu interface corresponds to the main menu. This interface is composed by a *presentacionalObject*, a *text* and three *anchors*, one to accede to the Informatics Engineering (6 year career) subjects, other to accede to the Informatics Engineering (4 year career) subjects and the last one to accede to the teachers list. SubjectsbyTeacher interface has an object *text* with the name of the teacher and an anchored collection that represent a linked subject list, which allows navigating to an interface where it appears a list of professors who teach the selected subject. TeachersbySubject interface has an object *text* with the name of the subject and an *anchoredCollection* that represents a linked teachers list, which allows to navigate to an interface where appears the list of subject taught by the selected teacher.

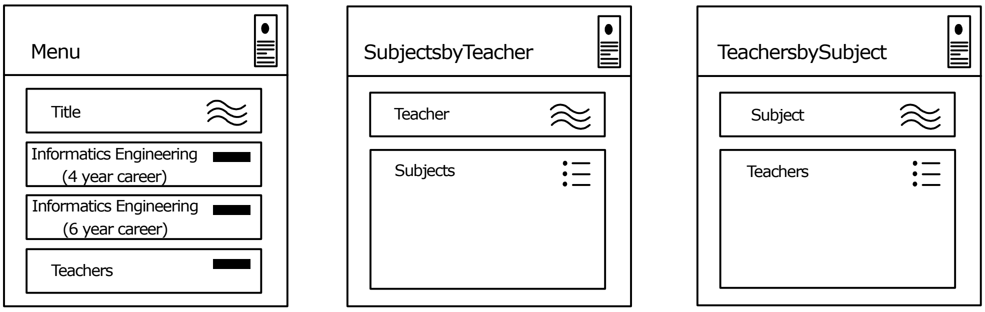

Figure 8 *Static Interfaces of the Teaching WAP System.*

2.3.2. Dynamic Presentational Model

Dynamic presentational model describes the behaviour of the interface objects, i.e. when a user trigges the change of information in frame or a complete page, when a video or sound is executed, or when the system gives back an answer after having completed a form. UWE proposes to use UML state diagrams [16] to represent the behaviour of these objects. A state is specified by a name, entry and exit actions, internal transitions, and/or substates.

3. Design Stage Approach

In the previous section we used UWE to design a WAP application. The aim was to demonstrate, there exists a well-founded way to design WAP applications using a specific methodology for web applications. Now giving an additional value to this work it seems interesting to us to make some adaptations to UWE, to orient it specifically to the development of WAP applications. These adaptations are centered in:

- Develop models and specific diagrams for WAP applications using UWE notation.
- Reduce the total amount of documentation generated in the design stage, orienting the models and diagrams specifically to the development of WAP applications.

3.1 Why is it important to avoid excessive amounts of documentation?

A well-known trouble, mainly in the development of hypermedia applications, is the generation of excessive documentation. In our example, this problematic is noticed in the excessive amount of diagrams generated in the application design step, considering that the other steps also produce a significant bit of documents, the resultant amount is extremely exaggerated and in most of cases generates huge difficulties [17,18], due to this it is necessary to consider some ideas proposed by Agile Software Development [19]:

- The documentation must be created and maintained only if it provides an important additional value for the project.
- The redundant documentation must be eliminated.
- The development of models and documentation requires a concerted effort that often it inhibits to reach the primary target of the project *to obtain a quality product in a short time.*
- When the documentation is excessive the maintenance of it self becomes a too expensive and bureaucratic task
- While greater amount of documentation, more effort is required to find the precise information.

3.2 Is it possible to reduce the total amount of documentation?

In our case, it is really possible to reduce the documentation, essentially because a WAP application is composed by a smaller amount of multimedia elements and internal interactions in comparison to a web system (due to the limitations related with bandwidth and size display). These limitations allow us to design a WAP application, developing a smaller amount of documentation than the required for the development of a web system.

3.3 How can we reduce the total amount of documentation?

In the design stage proposed by UWE, we need to build the conceptual model plus four UWE design models. Figure 9 shows the models built in the design steps proposed by UWE.

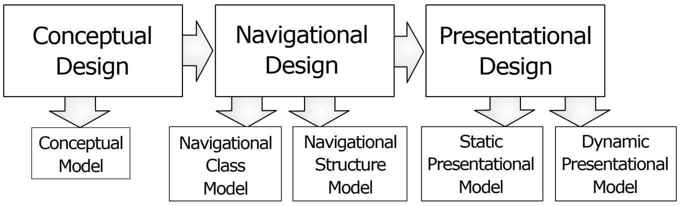

Figure 9 *Models built in the design steps proposed by UWE.*

Our proposal gives a specific extension of UWE for the development of WAP applications and related to this we obtain a reduction in the total amount of documentation generated in the design stage. To get this, we propose the design in three steps, but with the use of only two diagrams, the conceptual model plus our new WAP navigation model. Figure 10 shows the models built in the design steps of our proposal.

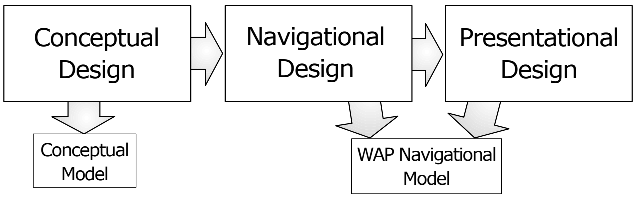

Figure 10 *Models built in the design steps of our proposal.*

3.4 Conceptual Design

The conceptual design proposed fundamentally does not present variations with UWE, nor to the other web development methodologies, due to the fact that the conceptual model is the product obtained in this stage. This model already is the standard in the applications development and represents the base for the classes development and the system data base.

3.5 Navigational Design and Presentational Design

In these steps we propose to replace the four UWE design models (shown in figure 9) for a new unique UWE based model called WAP navigation model (shown in figure 10). This model will serve us to understand the navigation and existing relations among each one of the nodes of the system. Within this model we will also include the interfaces abstract design. Figure 11 shows the navigation model proposed, applied to the same study case.

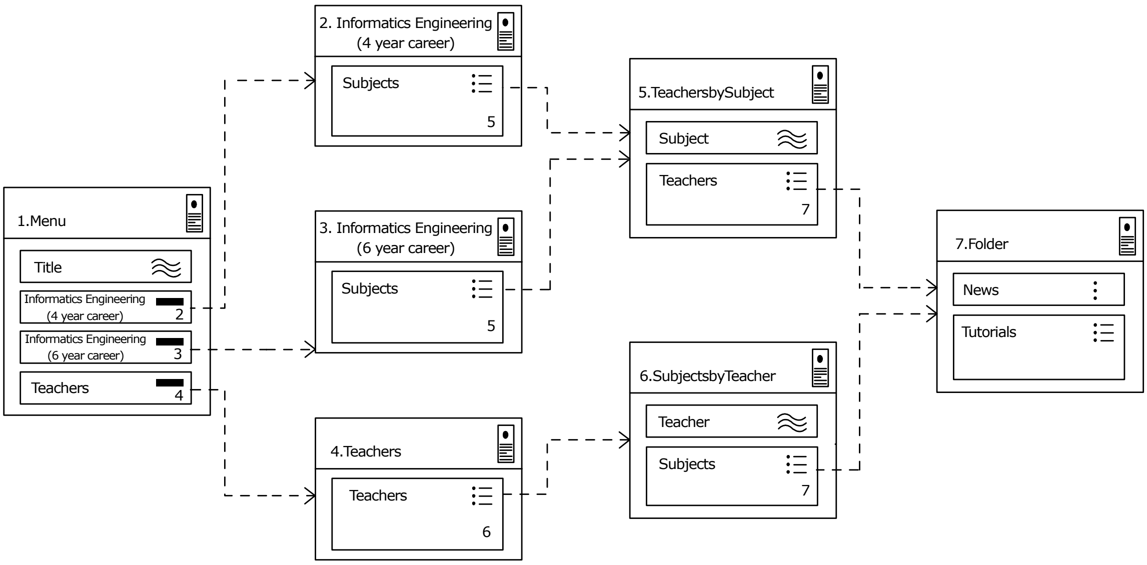
2. Informatics Engineering (4 year career)
Subjects
5
1.Menu
Title
Informatics Engineering (4 year career)
2
Informatics Engineering (6 year career)
3
Teachers
4
3. Informatics Engineering (6 year career)
Subjects
5
4.Teachers
Teachers
6
5.TeachersbySubject
Subject
Teachers
7
6.SubjectsbyTeacher
Teacher
Subjects
7
7.Folder
News
Tutorials

Figure 11 *WAP Navigational Model of the Teaching WAP System.*

This model can support two steps: navigational design and presentational design. In this model each interface and each anchor or anchored collections has an identifier. This way it is possible to recognize the target interface when starting navigation, for example, in the interface menu the anchor *Informatics Engineering (4 year career)* it is identified with 2, which indicates that anchor carry us to the interface with identifier 2 (*Informatics Engineering -4 year career-).* Therefore the complete information of all the system navigation is contained in the model. Furthermore, using the UWE notation, the abstracts interfaces also are contained in the model.

It is possible that the model turns out to be very huge when the system has many nodes, but we can construct a summarized model [20] using only the identifiers. Figure 12 shows the corresponding reduced navigation model.

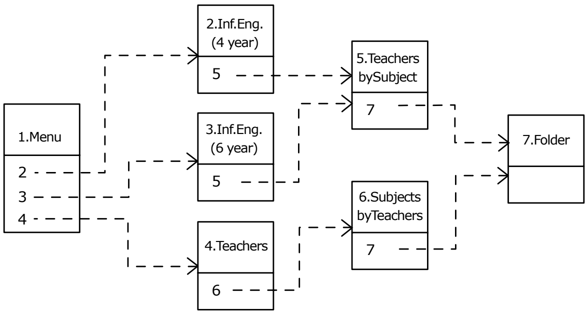

Figure 12 *Reduced Navigation Model of the Teaching WAP System.*

4. Implementation

In this section a sketch of the implementation of our study case will be given, mainly to understand some advantages that our model provides at the implementing time of the application:

- The clarity of the navigation model allows us to easily understand the system navigation and therefore a fast construction of the links between nodes.
- The similarity between the abstract interfaces and the final interfaces allows us to easily understand the objects layout inside the screen
- The simplicity of the model and the abstract interfaces reduces the learning period in being able to use the methodology
- The previous aspects aim to reduce resources needed and time used in the application development.

Figure 13 shows the similarity between abstract interfaces and final interfaces. In addition we can easily compare and understand the navigation depicted by the proposed navigational model.

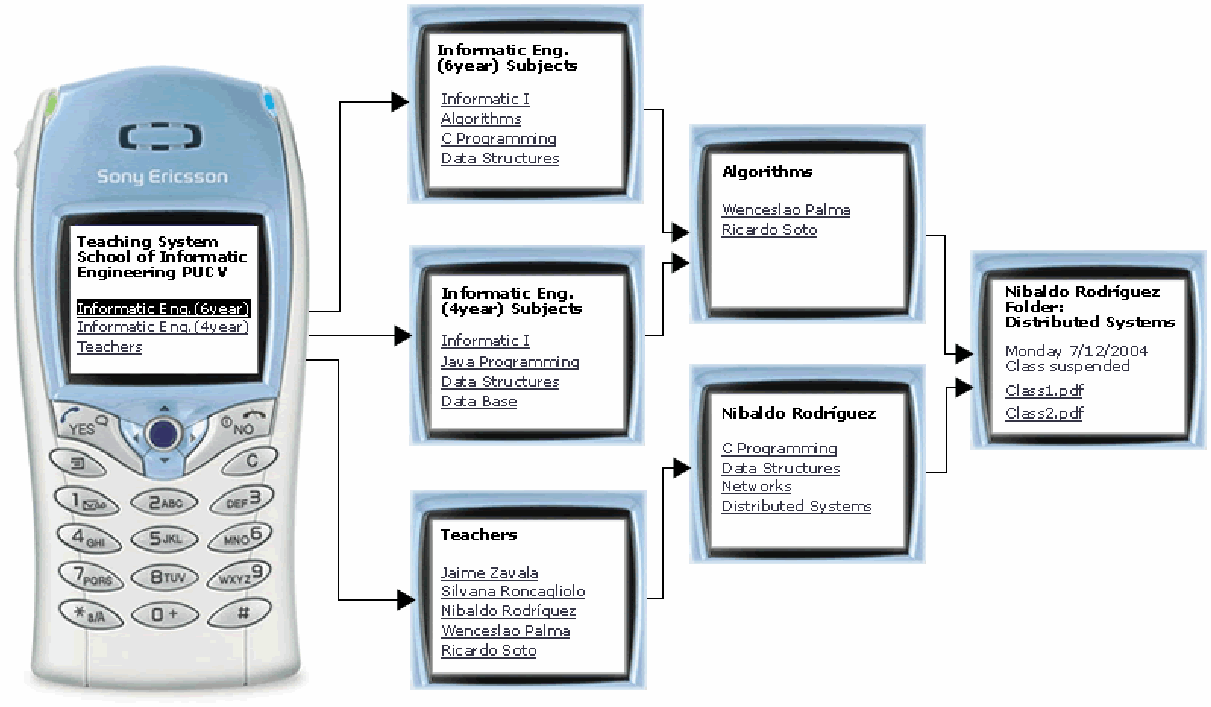

Figure 13 *Reduced Navigation Model of the Teaching WAP System.*

Figure 14 shows wml and php codes of the subjectbyTeacher page (deck), in which we can see the data base connection, the objects layout using tables and the links to other pages (decks).

```
<?php echo "<?xml version='1.0'?>";?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<?php
$cone = mysql_connect ("localhost", "<user>", "<password>") or die ("Can't Connect");
$desc = mysql_select_db("infoweb", $cone) or die ("Can't Open DataBase");
$n = 0;
$sqlres = "select * from folder where user = '".$teacher."'";
$res = mysql_query($sqlres,$cone) or die ("Query Error");
?>
<wml>
<card id="t1" title=<? echo $teacher ?>>
<p>
<?
while ($res = mysql_fetch_array($res)){
?>
<table>
  <tr>
   <td ><a href="docalu3_wap.wml?asig=<?echo $res["subjectCode"]?>&teacher=<? echo $teacher
?>"><? echo $res["subjectCode"]?><? echo $res["subjectName"]?></a></td>
  </tr>
<?}?>
</table>
</p>
<anchor>
   Informatics Engineering (6 year career)
<go href="docalu_wap.wml?career=ici"/>
</anchor>
<anchor>
   Informatics Engineering (4 year career)
<go href="docalu_wap.wml?career=inf"/>
</anchor>
<anchor>
   Teachers
<go href="docalu2_wap.wml"/>
</anchor>
</card>
</wml>
```

Figure 14 *Code of the subjectbyTeacher deck.*

5. Conclusions and future work

In this paper we prove, using a study case, that we can design WAP applications using UWE, a specific methodology for the design of web systems.

In addition, we presented a new UWE based model called WAP navigational model for the navigational design and presentational design steps, which synthesize the navigational and presentational aspects in only one diagram. The use of this model has demonstrated that it provides advantages such as clarity, simplicity to understand the system domain and to reduce resources needed and time used in the application development.

The proposed navigation model is the beginning of a complete study of the WAP applications development process, which will be applied in the development of different applications for mobile systems, to obtain experience and develop patterns and a tool CASE that allows us to simplify still more, the whole life cycle of WAP systems.

6. References

[1] Rolf Hennicke, Nora Koch. *"A UML-based Methodology for Hypermedia Design"*, Proceedings of the Unified Modeling Language Conference, UML '2000 Evans A. And Kent S. (Eds.). LNCS 1939, Springer Verlag, 410-424.

[2] Nora Koch. "*Software Engineering for Adaptive Hypermedia Applications"*, 2001, PhD. Thesis, Reihe Softwaretechnik 12, Uni-Druck Publishing Company, Munich.

[3] Nora Koch, A. Kraus and R Hennicker. "The Authoring Process of the UML-based Web Engineering Approach", In *First International Workshop on Web-Oriented Software Technology IWWOST'2001*, 2001, Valencia.

[4] Franca Garzotto, Paolo Paolini, Daniel Schwabe. *"HDM - A Model-Based Approach to Hypertext Application Design"*, ACM Transaction on Information Systems, vol. 11, nº 1, January 1993, pages 1-26

[5] D.Lange. *"An Object-Oriented Design Approach for Developing Hypermedia Information Systems"*, Journal of Organizational Computing and Electronic Commerce, ol 6(3), 269-293, 1996

[6] Tomas Isakowitz, Edward A. Stoh, P. Balasubramanian, *"RMM: A Methodology for Structured Hypermedia Design"*. Communication of the ACM, August 1995

[7] Schwabe D., and Rossi G. *"An object-oriented approach to Web-based application design. Theory and Practice of Object Systems (TAPOS)"*, (October 1998), 207-225.

[8] J. Conallen. *"Building Web Applications with UML"*, Addison Wesley, 1999

[9] O. De Troyer and C. Leune. WSDM . *"a User-Centered Design Method for Web Sites"*, Proceedings of the 7th International World Wide Web Conference, 1997.

[10] Nora Koch and Andreas Kraus. *"The Expressive Power of UML-based Web Engineering"*,

[11] Nora Koch and Luis Mandel. *"Using UML to Design Hypermedia Applications"*, Technical Report 9901, Ludwig-Maximilians-Universität München, March 1999.

[12] D.Leffingwell & D.Widring, "*Managing Software Requirement. A Use Case Approach",* Pearson Education, Boston, 2003.

[13] D.Schwabe and G. Rossi. *"Developing Hypermedia Applications using OOHDM",* Workshop on Hypermedia Development Process, Methods and Models, Hypertext´98, 1998.

[14] D.Schwabe, G Rossi and S. Barbosa. *"Systematic Hypermedia Design with OOHDM"*, Proceedings of the ACM International Conference on Hypertext, 1996

[15] D.Schwabe, G Rossi and F. Lyardet. *"Web Applications Models are More than Conceptual Models"*, Proceedings of the Web Enginnering Workshop, 2000.

[16] The Object Management Group (OMG). *"Unified Modeling Language"*.http://ww.uml.org, last updated March 2004.

[17] Kelter U., Monecke M., Schild M. "*Do we need 'agile' Software Development Tools?"*; p.408-421 in: Proc. Net.ObjectDays 2002 (NODe'02), 7.10.-10.10.2002, Erfurt; 2002

[18] Wendorff, Peter. *"An Essential Distinction of Agile Software Development Processes Based on Systems Thinking in Software Engineering Management"*; XP 2002 Extreme Programming; page218.

[19] K. Beck et al. *"The Manifesto for Agile Software Development"*, http://agilemanifesto.org/, 2002

[20] Ricardo Soto, Wenceslao Palma, Silvana Roncagliolo. "*Propuesta de un Modelo Navegacional para el Desarrollo de Aplicaciones basadas en OOHDM"* X Encuentro Chileno de Computación, Copiapó, Noviembre 2002