An Approach for Designing Ubiquitous Web Applications: A Case Study

Vito Perrone, Paolo Paolini
perrone|paolini@elet.polimi.it
HOC (Hypermedia Open Center) – Politecnico di Milano
Via Ponzio 34/5 - 20133 Milano
Italy

Abstract

Web applications have become more complex blending together several different features: navigation paradigms to browse across information, operations to manipulate data and to perform actions, interactivity, full-fledged transactions, etc. In addition the need to consider several possible devices (including mobile ones and interactive TV), as ways for to user to access the "same application" and the fact that different user categories need to share the same applications, make the picture even more complex.
Current design methodologies are not satisfactory, since they tackle one piece of the problem (e.g. hypertext features, or transactions, or operations, or data), but they do not cover the whole application.
This paper introduces an innovative design methodology from the UWA project, covering a broad spectrum of features, and based upon UML notation. A "real life" case study (in the Banking sector) is used as an example.

Key Words

Web Application Design, Multi-channel, UML, HDM, Ubiquitous Information Systems.

1. Introduction.

In the last few years, a new generation of Web Applications has appeared with new and sophisticated features and characteristics. These new applications exhibit much more complexity with respect to "old-fashion" Web Applications. Traditional Web Applications, in fact, were basically passive repositories of hypermedia information (organized in an hypertext structure, according to the original Berners Lee definition). Users were able to access, navigate and consume information, and a few interactions with the application, beside navigation, were supported or provided. The new generation of Web Applications, instead, offers new features to web users, increasing the complexity of the applications themselves and also the complexity of their development process. Modern web sites, in fact, offer several different kinds of services, to their users. All kind of operations, and sometimes transactions, are made available to the users: e-commerce, e-business, e-banking, e-government, e-administration, etc. are well known examples of this new generation of web applications.
Beside the increased complexity of the applications, a new integrated environment (of wireless communication devices) allows users to be "online" almost anytime and anywhere. New devices like PDA's (Personal Digital Assistant), last generation cellular phones, portable computers like Palms, and iTV (interactive TV) devices, have come to the stage and are nowadays integrating the "pure" web access. As a consequence, web applications need to be adapted in order to support this new generation of devices, with all the limitations (and constraints) that they carry on with them.
In addition to the above considerations, it is becoming more and more necessary to develop "families of applications" presenting the same content-services to different categories of users in different contexts.

Therefore we can synthesize as it follows the three main reasons for the increased complexity of web applications:

- Integration of several paradigms (e.g. hypertext, multimedia, operation-transaction, services) within the same application.
- Integration of different technologies (e.g. web, PDA's, phones, palmtop, iTV, etc.) within a seamless (for the user) environment.
- Need to adapt applications to different user profiles, different user preferences and different contexts.

The rest of this paper is organized in the following way: in Section 2 we describe what ubiquitous web applications are; section 3 presents a brief survey of actual web design methods and describes their weaknesses when applied to ubiquitous web applications; section 4 outlines the main characteristics of the UWA approach; section 5 exemplifies the usage of this approach in a real life case study; section 6, finally, presents our conclusions the guidelines for future work.

2. Ubiquitous Web Applications.

We call Ubiquitous Web Applications, UWApplications from now on, the kind of applications described in the introduction. This new type of applications involves a specific set of features; some of these features could be also found in traditional environments, but all the features together define a new complex picture. The most important features that we consider (out of the several possible ones) are the following ones:

Large amount of information: this creates a problem of structural-navigational complexity. The concern is not focused upon internal representation of data (as it could be for traditional Data Base applications), but rather upon how to organize information and interaction for an effective "user experience".

Operational and transactional services: we are interested in applications providing operational services and transactions to users. The complexity, with respect to traditional transaction oriented applications, stems from the need for integrating a service oriented approach to design (everything revolves around the service being provided) with an hypertext-hypermedia approach (everything revolves around navigation and browsing), as discussed in the next paragraph.

Hypermedia Interaction Paradigm: many different ways for interacting with the application must be provided to the users. Examples of these interaction paradigms are queries, searches, navigation, multimedia operations (such as play, stop, zoom, etc.), application operations, web services, etc. We assume that the driving paradigm should be hypertext-style navigation, but all the other paradigms must nevertheless be integrated with it.

Ubiquity: ubiquity is the ability of an application to be executed anywhere at anytime. This ability imposes the application to be multi-channel and multi-device increasing its complexity. Different screen sizes and features, different pointing-interaction mechanisms, different memory sizes, different middleware, different performances, etc. within the same application environment, add new complexities to the whole application development process, including requirements analysis, design and implementation.

Personalization: different types of users, for the same application must be considered (see the case-study in a later section, for an example). Each type of user has it own needs and requirements. In addition each single user may have his/her own personal preferences, and may operate in a different context (technological, geographical, etc.). In other words, the application needs to be customized according to all the different factors, requiring a specific approach to maximize reuse and minimize the development effort from one side, and to present a seamless application environment to the user.

Within the UWA project we assumed that the most important aspect to consider was how design UWApplications, rather than how to implement them. This not because implementation is easy, but because usability and effectiveness of applications, originate from a correct approach to design (and requirements); without this approach the best implementation strategy becomes almost useless.

In the next section we will briefly examine current approach to design of web applications.

3. Current web design methods.

While analyzing the current state-of-the-art for web applications development, different aspects must be considered that highlight the lack of a suitable design methodology covering design in all its different aspects:

- From a user point of view, most of the nowadays-available Ubiquitous Web Applications do not provide a satisfactory "user experience". As more applications will be delivered (in the next few years), the situation is supposed to be worst.
- From a developer's point of view, Ubiquitous Web Applications are very difficult and expensive to develop and maintain.
- Improving the design methods is the crucial point, on order to improve both the quality of the resulting applications and the efficiency of the development process.

Let us analyze, more in details, the expected benefits from improving the design methods: Improving the design methods should foster a better understanding of the requirements and hence a better adherence between requirements and implementation. Enhancing the requirements understanding may bring to a substantial decrees of undo and redo of some application components and to a reduction of costs and time required for the future evolution of the application.

The UWA approach mainly bases on the work done by HDM [3] but also considering the results achieved by numerous other research groups that have defined a number of "similar" methods like RMM [4], WebML [9], OOHDM [2][7][8], WSDM [1] and HFPM [5][6]. Lack of space prevents us to report a extended comparison of all the previous methods but in the following we briefly describe the most relevant aspect in respect of the main innovation introduced in UWA:

Transactional and operational services: in HDM original definition, transactional and operational services where not taken into account. New versions of this method, like HDM2 and W2000 arises this problem by providing pre and post conditions for specifying operations in the information, navigation and publishing model. Also specific adaptations of UML artifacts like collaboration diagrams are provided in order to describe the operation's execution flow combined with navigational aspects.

RMM doesn't provide an explicit way for designing operations in the context of hypermedia application. In its

original definition RMM do not deal with operations and their combination with navigational aspects.

WebML presents a solution supporting operations, introducing data entry and operations in the composition and navigation modeling. The composition model is extended with data entry and operation units. The navigation model is extended by adding to links the property of being operation-activating and by distinguishing a special class of links, called KO-links, associated with execution failure. Operation units are used to invoke generic external operations implemented by externally defined WEB pages. It should be noted that operation execution-flow and combination with navigational aspects seems poorly supported. More on this topic can be founded in [14].

OOHDM is an object-oriented based approach, so the natural way to support operation is to attach them to objects in the form of methods. Because nodes in the *Navigational Design Layer* of OOHDM are described with classes, operations are attached to node classes. However, aspects related to navigation and user interaction, like operations suspension and cancellation need to be further specified.

WSDM is focused on what they call "kiosk-like web sites" so no attention is given to operations modeling. In [1], the authors of the methodology explicitly say: "For our purpose, kiosk web sites behavior description is not yet needed."

Again, HFPM bases it operation modeling in that defined in OOHDM, so the same comments apply also for HFPM operation and transactional services support.

Ubiquity: as far as we could investigate, none of the methodologies taken into account here for being compared gives support for ubiquity issues. Only some of them (e.g. WebML) propose some variations for customization according to the device used for access the application. But the problem is quite more sophisticated. The ubiquity characteristic imposes constraints like for example customize the information supplied to the user according to it physical location, or adapting the application to the bandwidth constraints of each device. Although some further works on these methodologies proposes different approaches to some sort of ubiquity support, no one takes ubiquity in consideration as a fundamental issue, or states how ubiquity may influence hypermedia design and each of it design layers.

An interesting approach to creating WAP applications using WebML can be found in [13].

Customization: about customization, none of HDM, RMM and OOHDM gives support for customization or some sort of personalization for designing a hypermedia web application. Some articles present proposals for incorporating personalization features to some of these methodologies, like for example the proposal presented in [2] for OOHDM.

In WebML, personalization is considered in the Personalization Model design. Users and user groups are explicitly modeled in the form of predefined entities called *User* and *Group***.** The features of these entities can be used for storing group-specific or individual content, like shopping suggestions, list of favorites, and resources for graphic customization. In WSDM customization is centered on personalization. The first phase of WSDM method, User Modeling, concentrates in the potential users of the web site. During User Classification the audience of the web site are identified and classified in *user classes* that are a subset of the potential users who are similar in terms of their information requirements. Next, user classes are analyzed in more detail, focusing in information requirements and in the way that information should be presented to each user class. According to this analysis, user classes are divided in perspectives.

Finally, HFPM doesn't cover aspects relevant to personalization, like user modeling and adaptation issues. As it is based in OOHDM, it lacks for a customization support and integration with the proposed development lifecycle.

4. The UWA approach to design

The specific aims of the UWA approach can be synthesized as it follows:

- Improving the *quality of design*, intending, by quality, the completeness and readability of the design document and the "non ambiguity" of its semantics.
- Improving the *quality and effectiveness of the application*, intending to make the user experience efficient (satisfying his/her goal) and easy.
- Improving the *process of design*, intending making it more organized, with better support by tools, and, therefore, more productive.
- Improving the *efficiency of the overall development-maintenance cycle*, helping the developer to better cope with problems of evolution and changes of the application (dues to changes of context and/or changes of requirements).

In order to satisfy the above objective the project has taken the following approach:

- Devising a methodology to define requirements (intended as operationalization of goals) and organizing them. Requirements in UWA consider, among other parts, the problem of offering to each triple <user profile-task-context> the optimal user experience. Context, as it was explained in an earlier section, must be intended as including all the possible aspects (from technology, to location, to situation, etc.).
- Devising a global strategy for customizing the application that offers to designers a number of different options: families of applications, different user scenarios within the same application, different options within the same scenario, run-time

executable customization features (especially if runtime changes of context are detected).

- Devising a model (set of concepts), a notation (a way to represent concepts), a design methodology (describing the design process), a set of heuristic guidelines (providing practical tips and advices) and a design support tool (making the process efficient and supporting documentation generation) for each aspect of a WA design, i.e.:
  - *Requirements*: defining what the application must do, which are the application goals, and how to refine these goals into categorized requirements. [15]
  - *Hypermedia Design*: defining the information and interaction aspects of the application. Hypermedia design is divided into Information design (describing the entities, relationships and organizational aspect of the application relevant information), Navigation design (describing how the information will be organized in nodes and clusters for being navigated) and Publishing design (describing how the information and navigation schemas will be presented to the user). [16]
  - *Operations Design*: defining the operations made available to the user by the application. These operations are described using pre and post conditions and UML-like diagrams, for establishing how the user perceives the operation behavior and semantic [[16]. Furthermore, as one of the most innovative aspects, operation has been integrated to the interactive infrastructure of the application through appropriate interaction spots.
  - *Transaction Design*: defining the transactions (sequences of operations with specific additional properties), made available by the application. Transactional aspects concerning the ubiquity-ness of the application are considered and explicitly designed [17].
  - *Customization Design*: defining the adaptation of the application to context features, and, in particular to the characteristics of the device, the connection channels, the location, etc. We use customization rules for dynamically adapting the application according to context changes, and different design schemas for static adaptations.[18]

UWA is not concerned with implementation, but with design. Therefore there is no notion of standard UWA architecture. There are several motivations for this choice: lack of time and resources, difficulty in keeping track of the fast evolution of the implementation platforms, difficulty of reaching an agreement on a specific (possibly vendor based) solution, the problem of compatibility of a new application with preexisting technical solutions, etc. We have decided, however, to have a reference architecture, with the following motivations: to show that the design concepts we are introducing can be implemented; to provide a basis for defining runtime semantics for the applications; to provide guidelines to implementers in general and to the prototype applications of UWA in particular.

5. A case Study

5.1. The case study in a nut shell

In this section we show one of the case studies where the UWA methodology has been applied. The case study is a "simplified version" of (a small piece of) the information system of one of the most relevant Italian banks, very active in the on-banking and e-financing field [20]. The main goal of the application is to enable the bank to sell on-line banking services. During requirement analysis and design, 5 different departments of the Bank have been considered. Each department has its own set of requirements, preferences and needs. Moreover, the application has typical aspects of the modern web applications: it is multi-users and multi-device, it supports operations and transactions as long as an hypertext-like "look and feel", it has different execution environments and supports some customization features. Additional complexities were due to the need of taking into account other already existing applications, which the application being designed needed to be interfaced with.

The application covers the entire process of sale and management for debit-cards. Three different main user categories [12] have been considered: *End user (customer)*, interested in the purchase and management of the cards; *Salesman*, the person responsible for the keeping the relationships with a number of customers; *Product Manager*, the person responsible for managing the card catalogue and making sure that new cards are available on-line. Different user categories exhibit different needs. For example, the application must allow the *end user* to choose the card most suitable for his needs, to start a purchase and to follow all the subsequent phases, and to manage the cards already acquired, while for the *salesmen* it must allow the user to monitor the situation of his customers, to monitor the purchase procedure and to "step-in" for any customer request, and must be possible to suggest to customers the best possible options (for choosing a card).

Beside these aspects related to the user type, it must be possible to use the application in diverse conditions, via various devices and in various places, allowing the user to access the service in the best possible way according to their situation. Ubiquity is a fundamental feature of the application that must be accessible via WEB and also via mobile devices such as WAP, PDA, and cellular phones.

5.2 Requirements Analysis

As it was said above, requirements analysis has been performed using an adapted version of the goal oriented methodology [11], modified also to take into account specific aspects of the ubiquitous web applications [15]. Stakeholders we identified are: Generic User, Bank Customer without Card, Bank Customer with Card, Strategy manager, Product Manager, Salesman, Financial Department, External Partner. They exhibit very different goals that sometime conflicted and other times cooperated to arise an application requirement. In figure 1 a sketch of the overall requirements diagram is shown where it is possible to see the part concerning the *Bank Customer without Card* that is defined as "a customer of the bank, who does not own any card yet". In the diagram, Ovals represent the goals while the rectangles the requirements. Lines with dashed ends mean a relation with some other stakeholder's goals. One of the main innovation of the requirements approach within UWA is to classify requirements into Categories concerning with the hypermedia design dimensions. Different categories of requirements are represented by a letter in a small square in the upper right corner of the requirement rectangle. The requirements categories taken into consideration by the UWA methodology are: **C**ontent (concerning which information should be "offered" by the application), **S**tructure (concerning the organization of content), **A**ccess (concerning the strategy offered to the users to locate pieces of content), **N**avigation (concerning hypertext features of the application), **P**resentation (concerning the organization into pages), **U**ser operation (concerning operations available to the users) and **S**ystem operation (concerning operations that the system must perform).

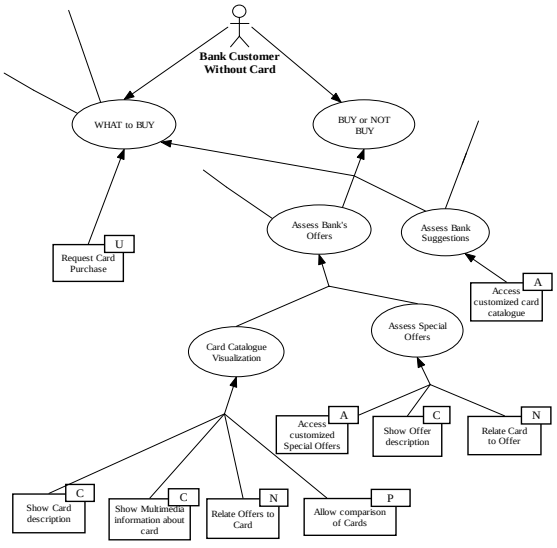

Figure 1: Requirements specification for a "Bank Customer without Card"

5.3 Application Design

A general decision concerns the choice between "static customization" (different design diagrams for different contexts) and "dynamic customization" (variations of design, possible executed at runtime, within the same diagram) [18]. Static customization has been used, for this application, for "customizing" the features relevant to each user category, while dynamic customization (expressed via customization rules) has been used in order to describe how to take into account different devices.

As example of static customization we show, in Figure 2, and Figure 3, the overall information schemas for two user categories, the bank customer and the salesman.

Let us analyze, for example, (the entity type) Customer's Card. For Bank Customers it represents the cards the user owns, while for a salesman it represents all the cards owned by the customers (assigned to that salesman). This difference becomes clear when looking to the entity type cardinality: a customer can own from one (at least) up to five cards (at most), and two as average. For a salesman, instead, the cardinality of accessible cards is a function depending on the number of customers (each one having a number of cards).

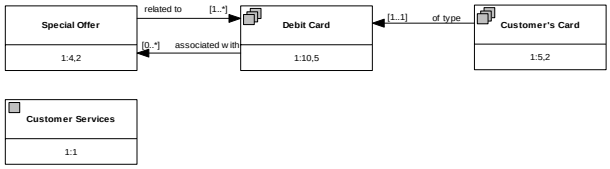

Figure 2: Information Schema for the Bank Customer

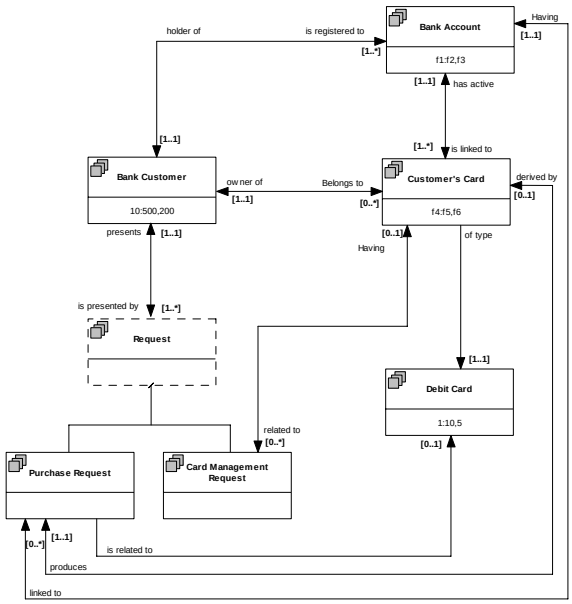

Figure 3: Information Schema for the Salesman

Figure 4 shows an example of customization. If a Bank Customer accesses the application using a PDA (as opposite to a standard PC), a number of features are modified in the information design. Customization rules

(part of what is called “dynamic customization” in UWA) can be applied, in order to modify the design.
The component *Contract*, for example, is disabled for PDA (since designers have decided that it isn’t useful for the user to view the contract on a small screen).
After information design has been completed (for each user category), “navigation design” must be carried on. Within this activity, information is organized into consumption (for the user) units, called “nodes” (using a standard terminology of hypertexts). Nodes are grouped into clusters, and within each cluster they are interconnected via links (again using standard hypertext terminology). Figure 5 shows a small fragment of the navigation design, and specifically it shows the navigation aspects of an access structure [10], that is a structure supporting the location of objects of interest.

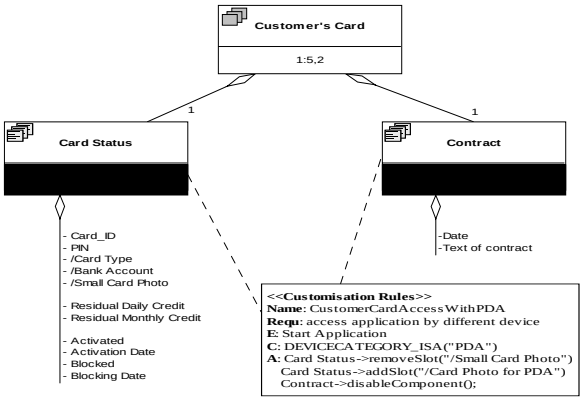

Figure 4: Customization rule applied to Bank Customer for two different devices.

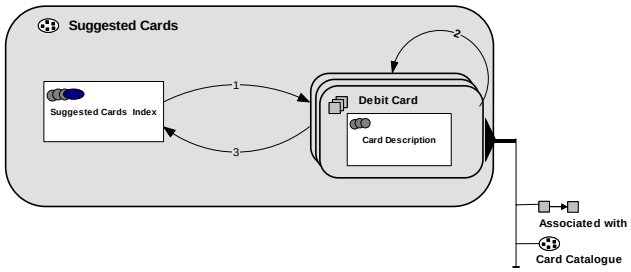

Figure 5: Navigation Cluster for the Collection “Suggested Cards”

Access structures within UWA are expressed via the notion of *Collection*, and Figure 5 shows how navigation is organized for the collection used to “suggest” possible cards to the user[1]. A collection defines a cluster (the big rectangle with rounded corners), i.e. a navigational context, in which the user can access to the suggested cards through an *index* (the “collection center” modeled by the left rectangle inside the big rectangle). Once the user selects, using the index, a card, the link (represented by the arrow) takes him/her to the information concerning the card itself. Since several nodes are used to describe a card, the specific node *Card Description* is indicated as starting point. It is important to point out that the same navigation cluster is used for PC and PDA; the only difference is that the content of the node “Card Description” is larger for PC, and smaller for PDA.

The next activity of the design is the “Publishing Design”, i.e. the overall organization of the application into pages. In most cases this activity can take technology into account, say the sizes of the screen, the pointing devices, etc. In figure 6 we show a page designed for PC and the customization rule for the corresponding version for PDA.

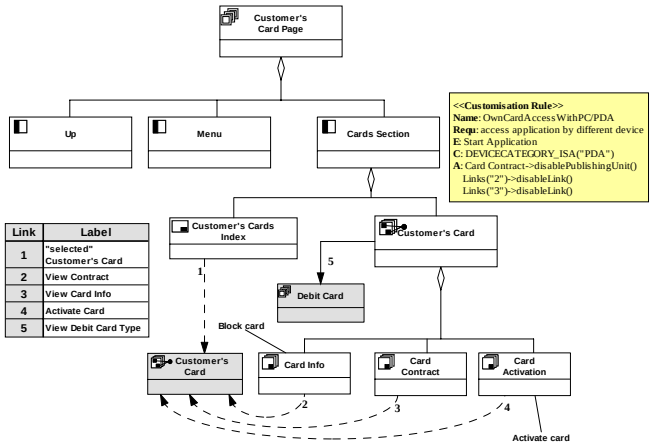

Figure 6: Customer’s card Page with customization rule

For a PC the page is organized into three main sections; the third section is organized in two subsections and so on. Some sections incorporate directly nodes (derived from navigation design) while other sections represent links across pages, defined within the publishing design itself (a statement like “all pages are linked to the home page”, for example, is dealt with within the publishing design). In most cases a page will contain a single node; in a few cases a page may contain a whole cluster, or a piece of it. As a consequence, some of the links, defined in the navigational schema, become links across pages, while some others will become links within the same page (if the corresponding nodes are stored in the same page).

In order to give an idea about the complexity of the prototype application, the following numbers can be useful: three different design documents (one for each user category), 13 entity types, 20 semantic associations (navigation relationships driven by semantics), 15 collections (access structures, as in the example of Fig.6), 40 navigational clusters (as in the example of Fig.6), 3 page templates, 50 page types (as in the example of Fig. 7).

6. Conclusions and Future Work.

[1] The reader is deferred to [16] for details.

The main advantage of the UWA methodology, with respect to "competition", is the integration of different factors (requirements, hypermedia, operations-transactions, customization, publishing, etc.) that are otherwise scattered in different and not integrated methodologies. In UWA requirements are directly connected to design aspects like information structures, navigation structure, operations and so on, moreover they are explicitly mentioned in the customization rule definition. The customization aspects have been tailored upon the design primitives provided by hypermedia model, and operations and transactions fully integrated to the interaction paradigm through attaching them to navigational elements and providing appropriate navigational contexts.

As far as notation is concerned, UML has been the first choice, for a number of reasons, the most important one being "marketing" concern: the first segment of the market we wish to address is represented by those designers already accustomed to complex design and already using UML. We have defined an extension to standard UML (a UML "profile", with a number of "stereotypes") for representing all the different notions of the methodology.

Using this UML-based notation we also developed a set of tools, devised as a plug-in for RATIONAL ROSE [22] design support tool. Again the market, being ROSE the best selling toolbox in the UML community, has driven the choice. The final version of tools with a demonstration tutorial and the complete example of pilot application is currently available to download [23]. In the meantime we are also working on a different version of tools, based on our own metamodel (in order to ensure flexibility to the evolution of the methodology). This additional version of the tools allows us to make the methodology available, with a different simpler notation, not related to UML. This simplified notation will be suitable for communities of designers not familiar with UML (an important segment of the market for Web Applications).

Acknowledgements

The work has been partially funded by the European Commission, as part of the UWA project. The authors express their sincere thanks to all the partners of the UWA consortium, for their support and collaboration in carrying on the project in all its different aspects. A special tanks is due to Mauricio Sansano who has carried on the investigation regarding the usability of the UWA methodology.

References

[1]. De Troyer and C. J. Leune, WSDM: a user centered design method for Web sites, *In proceedings of the 7th International World Wide Web Conference. 1997*

[2]. Rossi G., Schwabe D., Guimaraes R., Designing Personalized Web Applications, *Proc. of the WWW10*, Hong Kong, May 2001.

[3]. Garzotto F., Paolini P., Schwabe D., HDM: a model-based approach to Hypertext application design, *ACM Transactions of information systems. 11(1),* 1-26, 1993.

[4]. Isakowitz T, Stohr E, Balasubraimanian P., A methodology for structured hypermedia design, *Communications of the ACM 8(38)*, 34-44, 1995.

[5]. Olsina, L., Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy, *1st International Workshop on Hypermedia Development in conjunction with ACM Hypertext98*, Pittsburgh, US, 1998

[6]. Olsina L., Functional View of the Hypermedia Process Model, *Fifth International Workshop on Engineering Hypertext Functionality at IEEE ICSE´98*, Kyoto, Japan.

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

[8]. Schwabe D., Rossi G, Barbosa S., Systematic Hypermedia design with OOHDM, *Proceedings of the ACM International Conference on Hypertext, Hypertext96.* 1996.

[9]. Ceri, S., Fraternali, P., A. Bongio, Web Modeling Language (WebML): a Modeling Language for Designing Web Sites, *Proceedings of the WWW9 Conference*, Amsterdam, 2000.

[10]. L. Baresi, F. Garzotto, and P. Paolini, Extending UML for Modeling Web Applications, *Proceedings of 34th Annual Hawaii International Conference on System Sciences (HICSS-34).* IEEE Computer Society, 2001.

[11]. Dardenne, A. van Lamsweerde, and S. Fickas., Goal directed Requirements Acquisition, *Science of Computer Programming*, 20:3–50, 1993.

[12]. A. C.W. Finkelstein et al., Ubiquitous Web Application Development - A Framework for Understanding", *6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, US, July 2002

[13]. http://webml.elet.polimi.it/webml/readings/WAP.html

[14]. http://webml.elet.polimi.it/webml/readings/WebML2.html .

[15]. UWA project Deliverable D6, Requirements Elicitation: Model, Notation and Tool Architecture, http://www.uwaproject.org/

[16]. UWA project Deliverable D7, Hypermedia and Operation Design: Model, Notation and Tool Architecture, http://www.uwaproject.org/

[17]. UWA project Deliverable D8, Transaction Design: Model, Notation and Tool Architecture, http://www.uwaproject.org/

[18]. UWA project Deliverable D9, Customization Design: Model, Notation and Tool Architecture., http://www.uwaproject.org/

[19]. UWA project Deliverable D11, Requirements and Design Specification for Bank121 Pilot Application, http://www.uwaproject.org/

[20]. Banca 121's official site www.banca121.it

[21]. Mauricio Sansano, Critical success factors for the UWA Project, *Internal Report*, Politecnico di Milano 2001

[22]. http://www.rational.com/products/rose/index.jsp

[23]. http://www.uwaproject.org