UR-WSDM: Adding User Requirement Granularity to Model Web Based Information Systems

Wim Goedefroy
STARLab, Vrije Universiteit Brussel
and
Robert Meersman
STARLab, Vrije Universiteit Brussel
and
Olga De Troyer
Infolab, Tilburg University

In this position paper we argue that the overall development of a Web based Information System can substantially benefit from adopting a user-centered design approach. Such an approach is characterized by an analysis phase identifying relevant *user classes* around which design and development are subsequently organized, as opposed to being driven by the site's data only. We suggest a further improvement on this design method by defining user classes through a finer level of granularity made out of user requirements. We claim this improves both the overall clarity of the models and their suitability for maintenance and reuse.

Additional Key Words and Phrases: Web engineering, user centered, Web design methods, Web based information systems

1. INTRODUCTION

Today Web-related software development seems to be faced with a crisis not unlike the one that occurred a generation ago when in the 70s computer hardware experienced an order of magnitude increase in computational power. While this made possible the implementation of a new class of applications larger both in size and complexity, the methods for software development available at that time were not able to scale up to such large projects. The "software crisis" [Boehm 1981] was a fact with its legendary stories of delays, unreliability, maintenance bottlenecks and cost overruns proportional to software project consultant's fees.

Today we seem to be dealing with a corresponding "website crisis". Over the last few years, the Internet has boomed and the World Wide Web with it. Web browsers have become the basic user platform of the Internet. Because of this immense potential audience, and because publishing on the web is very easy, the number of web applications has exploded. Most of these sites are created opportunistically without prior planning or analysis. Moreover, even large mission-critical company intranet projects are being started while the methodology and theory to underpin such projects is still lacking.

So, analogous to software engineering we may expect a discipline of *web engineering*, to provide structured approaches for design, development and maintenance of web applications [Gellersen et al. 1997].

In this position paper we present some suggested contributions to this field. Our paper is organized as follows: Section 2 shows the application of the most relevant phases of our method, called UR-WSDM on the sample problem given in the CfP for this workshop; Section 3 lists our requirements for a website design method, explains the importance of some of the methodological

Name: Wim Goedefroy, Robert Meersman
Address: STARLab - VUB, Pleinlaan 2, 1050 Brussels, Belgium. Phone: +32-2-6293750; fax: +32-2-6293525; email: (wgoedefr|meersman)@vub.ac.be
Name: Olga De Troyer
Address: Infolab - Tilburg University, PO Box 90153, 5000 LE Tilburg, Netherlands. Phone: +31-13-4662432; fax: +31-13-4663069; email: detroyer@kub.nl

options we took in UR-WSDM and compares it to existing work.

2. SOLUTION TO THE SAMPLE PROBLEM

2.1 Introduction

In this first section we will give a summary of our *solution to the sample problem.* Because our research mainly concerns analysis and design of Web Based Information Systems (WBIS)[Takahashi 1997] we will particularly focus on these aspects of the development cycle.

The topic of designing Web sites is hardly new. Early hypertext systems design evolved into methods and models for designing hypermedia and web applications. The best known of these are RMM [Isakowitz et al. 1995], HDM [Garzotto et al. 1993] and OOHDM [Schwabe and Rossi 1995] but others exist. Our approach starts from the (web-specific) method WSDM [De Troyer and Leune 1998] and adds user requirements as a finer level of design granularity. We use the sample problem to illustrate how this granularity influences the User Modeling and Conceptual Design phases of WSDM. We have called the resulting extended method UR-WSDM.

2.2 User Modeling

Most of the current methods begin with modeling the planned site's data using E-R [Chen 1976] or OMT [Rumbaugh et al. 1991]. Next, a navigation model is designed that describes how users navigate through the site. A so-called *data driven* approach is employed where web applications are created with (access to) the site's data foremost in mind. On the contrary, WSDM adopted a *user centered* [De Troyer and Leune 1998] rather than a data driven approach. In a user centered approach we determine the needs of intended users/visitors, identify *user classes* that "represent" these needs, build a "matched set" of object models each corresponding to a variant on a user class, and essentially generate a navigation and manipulation model from the information gathered. This results in applications that are more tailored to users/visitors.

The first step in the WSDM method is to define the *mission statement* [De Troyer and Leune 1998]. The mission statement defines the purpose of the site as seen by the provider (server). *As the problem statement for this workshop contained no explicit mission statement, we subsumed one from the CfP.* (See Appendix A)

The next and crucial step is to identify *user classes.* Informally in the approach at hand a user class stands for a group of users sharing the same information requirements. One possible way of doing this is described in [De Troyer and Leune 1998]. In the case study at hand we may identify the following user classes: internal researchers, outside researchers, current students, prospective students, research sponsors.

Note: this is where UR-WSDM refines WSDM, which depended on a declared set of *user class variants* (or *perspectives*) assumed to share a related, but undeclared as such, set of requirements.

The user classes are now analyzed in detail. This leads to a list of *user requirements* where every requirement can be seen as an explicit reason to browse to the site. For simplicity and lack of space, we restrict ourselves here to two representative user classes: outside researchers and current students. Figure 1 and Figure 2 show an analysis of the requirements of the outside researchers and current students respectively. As one can see, these requirements cover both retrieval and manipulation (update) of information.

2.3 Conceptual Design

Knowing the requirements, we represent them in a structured way. In the Object Modeling phase data needed to meet the requirements is modeled as objects. Then, in Navigation Design, we will consider how to present that information and design the communication between user and WBIS that allows him/her to manipulate information. We end Navigation Design with the creation of a navigation model. This describes how users reach the place in the site that meets their requirement.

2.3.1 *Object Modeling.* In a user-centered approach like WSDM we do not model behavior per individual user but instead create a model for (every variant of) every user class. In UR-WSDM, user classes are defined as specifying groups of users that satisfy a same set of identified user requirements.

Outside Professors and researchers

1. View the labs of the department; for every lab, view some information, the members attached and the projects. For every project, get a short description and the sponsors. For every member, view some information and his/her contributions. For the sponsors, get a link to their homepage.

2. View a list of all results. For every result show the authors. Link the author's name to a page showing the author's name, information and results.

3. View a list of all research projects. For every project show the labs conducting the project. The lab is linked to a page describing the lab as in requirement 1.

Fig. 1. Analysis of the requirements of an outside researcher

Current students

(limited)
1. For every prof, get the courses he lectures, for every course get a short description, the prerequisites, the syllabus in electronic form and other material added by the prof

2. Subscribe to a course

Fig. 2. Analysis of the requirements of a current student

Figure 3 and Figure 4 show the object chunks for the outside researchers and current students respectively. One object chunk corresponds to one user requirement. We chose an OMT-like notation primarily because of its compactness. Other techniques like ER [Chen 1976] or ORM (NIAM) [Verheyen and van Bekkum 1982; Halpin 1995] involve more, often unnecessary, detail. We put no restrictions however on which technique to use.

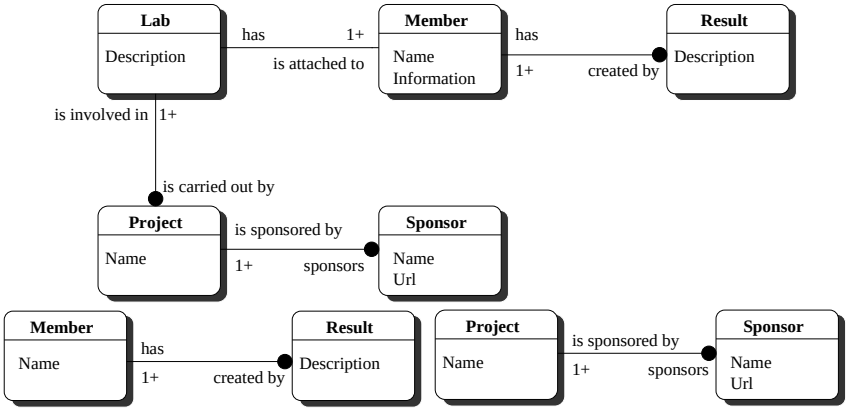

Fig. 3. Object Chunks for the outside researcher class. Top: requirement 1, Bottom-left: requirement 2, Bottom-right: requirement 3

Next, we merge these object chunks into a single model for every user class. The result reflects the view a user of a certain user class has on the information in our WBIS.

Figure 5 and figure 6 show the merged chunks for the outside researchers and current students respectively.

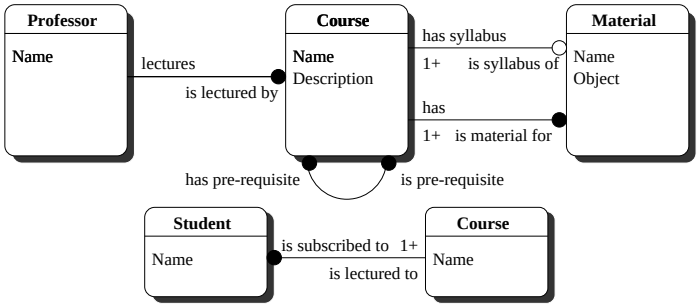

Fig. 4. Object Chunks for the current student class. Top: requirement 1, Bottom: requirement 2

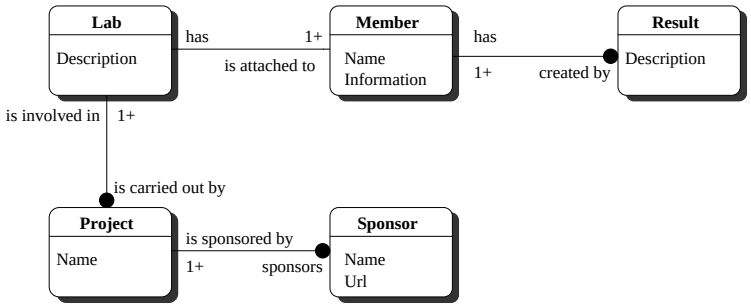

Fig. 5. Model for the outside researchers class

At this point all the models created are merged conceptually into one single model. In a group development environment, it may be necessary to resolve apparent inconsistencies (naming conflicts, ...) at this stage.

Figure 7 shows the resulting model.

2.3.2 *Navigation Design.* A WBIS supports two types of tasks, information retrieval and information manipulation. Each task has two aspects, viz. *what* to handle and *how* to handle it. While Object Modeling covered the *what*-aspect, this phase will treat the *how*-aspect. We start with the information retrieval part, the information manipulation part is modeled in 2.3.2.2.

2.3.2.1 *Navigation Chunks.* In WSDM, at this point, Navigation tracks are created. Such a navigation track describes how to represent the information for a certain variant of a user class. Because we refined our granularity to user requirements, in UR-WSDM we instead now create for every user requirement a *navigation chunk*. Figure 8 and 9 show the navigation chunks for the user requirements of our example. The notation should be self-explanatory. Rectangles around

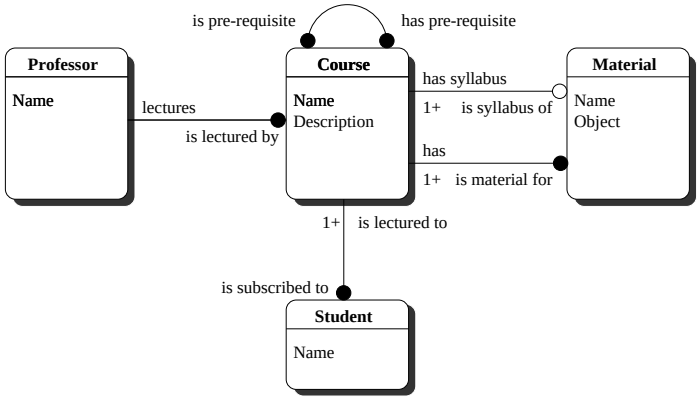

Fig. 6. Model for the current students class

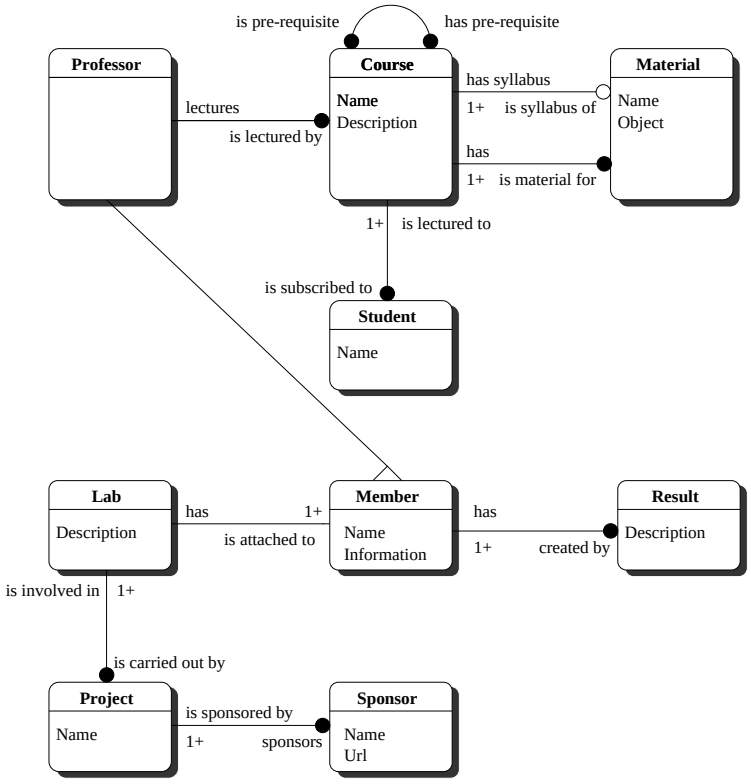

Fig. 7. Data model for the WBIS

object labels denote new components containing the object data. So figure 8 top left reads as:

> List the names of all labs. The name links to a new component containing the description of that lab, a list of members of the lab, and a list of projects the lab is involved in. A member name is a link to a component containing the member's information and a list of his/her contributions. A project name is a link to a component explaining the project and a list of its sponsors with their URL.

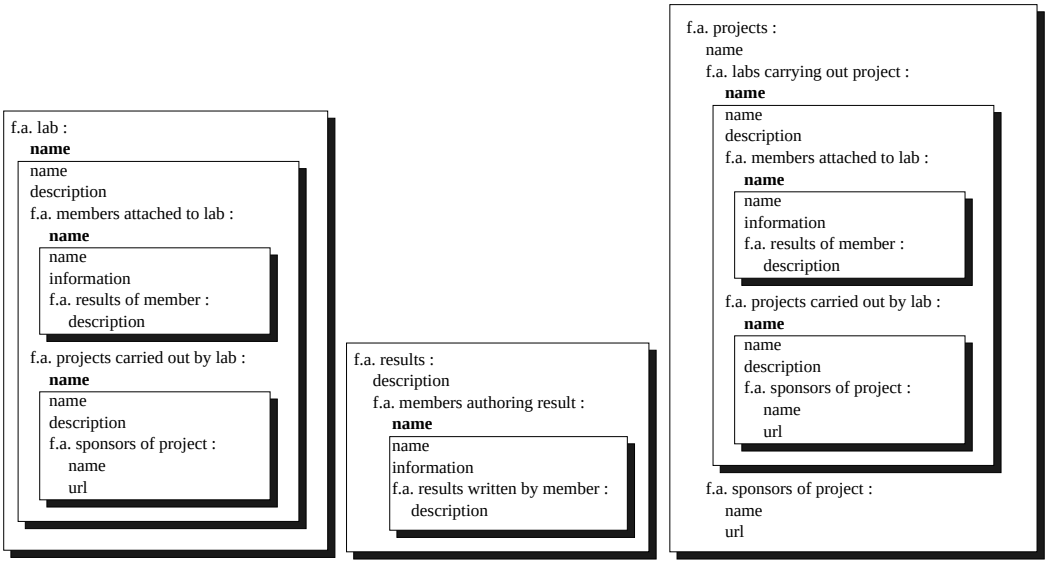

Fig. 8. Navigation Chunks for outside researchers

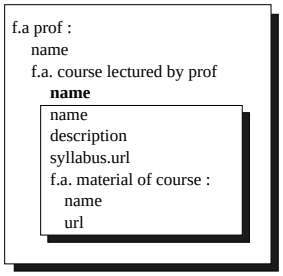

Fig. 9. Navigation Chunks for the current students

Again, we merge all chunks for a single user class. Then we search for data that is represented in different places. We remove possible conceptual redundancies. An example of how to do this is in Figure 10 which shows us the merged model for the outside researcher class. Unlike the current user class which had only one component model the outside researcher class is a little more complicated. There are separate nodes for a lab and a lab member.

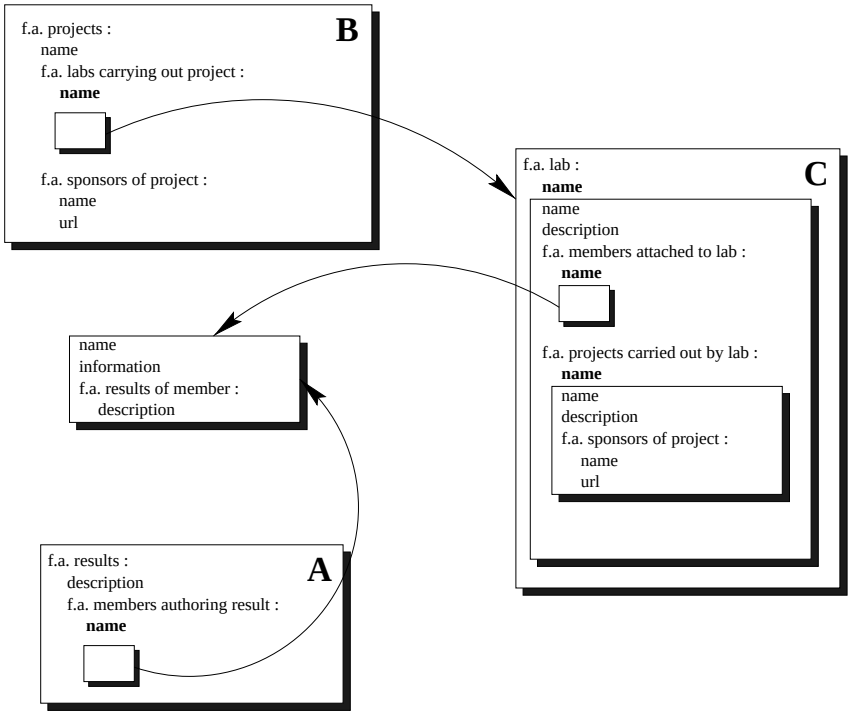

Fig. 10. Model for outside researchers

Now combine the models for all users. In our example, the result is both models of the previous step.

2.3.2.2 *User scenarios.* Now it is time to model the part of the WBIS that manipulates data. This task can be divided in two subtasks. One task models the communication with the user, the second describes what happens inside the WBIS. Only the first subtask is specifically relevant to this paper, the second is typically done with well-known traditional techniques like DFDs [Gane and Sarson 1979] but is omitted here.

To model the user–WBIS communication we make use of semi-formalized scenarios (also found in OO system design e.g. as *use cases* in UML [Jacobson et al. 1992]). For the sample problem however we merely use an ad-hoc technique, inspired by a very much simplified version of the scenarios in [Takahashi 1997]. We create two columns, one for the WBIS and one for the user.

On subsequent lines we write the interaction between the WBIS and the user and summaries of what happens inside the WBIS. The scenario that describes how a user subscribes to a course is depicted in figure 11.

Student	**WBIS**
	Ask student name
Enter student name	
	get all possible courses for this student
	show list
Select choice and submit	
	Ask confirmation
Confirm	
	Process data

Fig. 11. Scenarios for the current students

2.3.2.3 *Navigation Model.* To make the site easier to use, we add navigation components. Their purpose is to lead visitors to the information they need without having to memorize addresses. The whole of navigation chunks, scenarios[1] and navigation components is called the navigation model. For the sample problem we created a main component, showing the Department description plus the list of labs. The main component is linked to a research component and an educational component. The research component contains leads to information about projects and to the part about project results. The educational component will lead the current students to where they can see the courses, and eventually subscribe to a course. See figure 12.

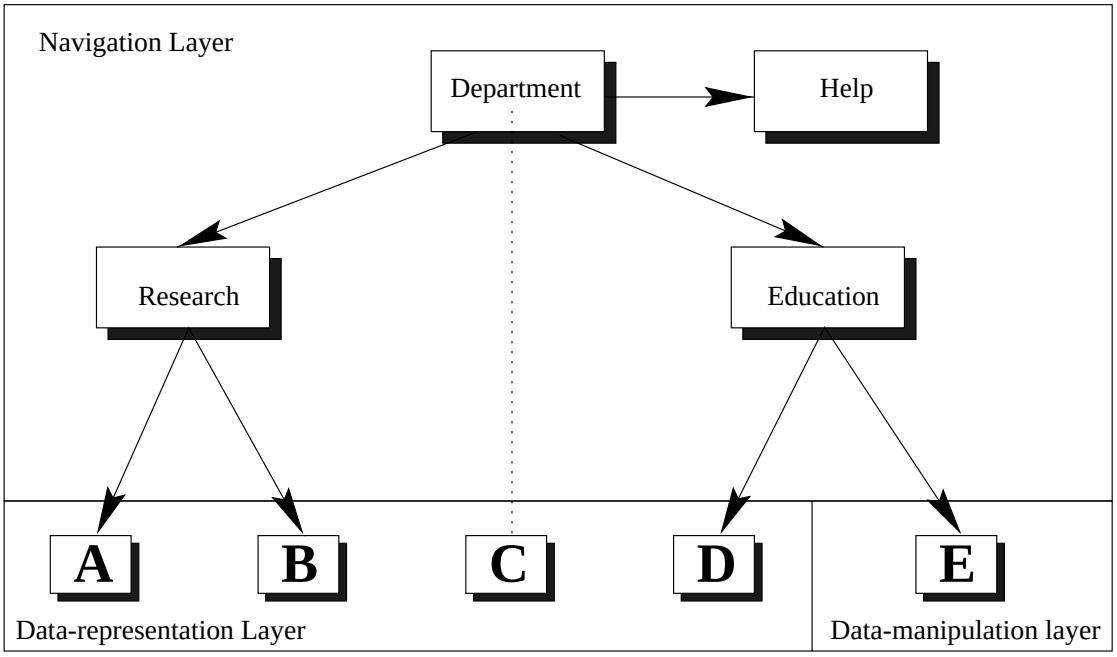

Fig. 12. The navigation model

[1] For a clear organization we use boxed letters to represent the navigation chunks and scenarios.

3. TOWARDS A USER-CENTERED APPROACH BASED ON USER REQUIREMENTS

3.1 Introduction

In the introduction of this paper we mentioned the need for new models and methods to support the development of large Web sites. The question arises why we cannot use the theory available for traditional software engineering? There are many differences between regular system development and the creation of a web site. Dissimilarities that are discussed in the literature are :

—Contrary to a homogeneous group, a web project involves people with different skills like authors, programmers, graphical artists, etc. [Isakowitz et al. 1995]. Models and methods must provide the ability to serve all those people.

—The design of hypermedia applications involves capturing and organizing the structure of a complex domain and making it clear and accessible to users [Garzotto et al. 1993]

—multimedia aspects of hypermedia applications raise numerous difficulties [Hardmanand et al. 1994]

One may also observe that in web applications the emphasis often is on information retrieval. Updates occur only occasionally, certainly much less frequently than in "classical" transaction-oriented information systems. This means that web site design methods tend be oriented towards modeling how to *represent* and *access* information rather than how to *manipulate* it.

A Web based information Systems (WBIS, [Takahashi 1997]) is a kind of information system that resides on the Web (usually in a distributed fashion) and uses a web browser as user interface. A WBIS also differs from a "classical" information system among other aspects by its frequent emphasis on information *publication*.

The next section describes what we like to see in a design method and model. 3.3 quotes interesting ideas from the models and methodologies presented in the literature. 3.4 explains why we added user requirement granularity to WSDM.

3.2 Ingredients of a design method and model

There appears to be a striking scarcity of research literature on the fundamental aspects of web-engineering for WBIS. Drawing on earlier experience with database design, methodologies and tools for these, we decided that our first effort should be aimed at a (formalizable) design model with the following primary requirements in mind:

—Both model and methodology have to be easy to use for many kinds of people. Remember that a web project typically involves non-computer educated people, and that a methodology has to be teachable. This makes a (partly) graphical modeling technique highly preferable. (However, for the purpose of subsequent theoretical foundation, textual representations are often more appropriate; for an example, see e.g. the use of the situation calculus in [Scherl et al. 1998] and [Bieber and Kimbrough 1994].)

—The same teachability argument and extensive experience with IS design methods require also that the adopted model be comprehendible, i.e. preferably using only a small number of primitives and components.

—We preferred a method which starts from user and usability requirements, in contrast to many existing methods which often start by modeling the data to be made available in the site. This constitutes one of the most fundamental differences with database and IS design methodologies. We felt indeed that the perceived quality of a website is determined not so much by the information a supplier wants to publish than by what and how its users find it.

—The model obtained should allow maximal reuse. Obtaining correct and complete user requirements turns out to be an expensive exercise, not unlike a product marketing project.

—The methodology should decompose into different design stages, allowing different levels of abstraction. We do not want e.g. to talk about pages and CGI scripts while we are designing an object model or when creating navigation patterns.

—A complete model and method must provide the ability to model two-way communication with the user. It must be possible for a WBIS to act on user-supplied data.

—It should be formal at least in the sense that the representation of an obtained model has a unique semantics, i.e. a unique interpretation of the representation in terms of a universe of discourse (application domain).

Following is a brief (and incomplete) summary of design models and methodologies for hypertext and web applications we considered relevant to ideas in UR-WSDM.

3.3 Existing models and methodologies

It is not our aim to explain the models/methods in detail; for this see the literature indicated.

3.3.1 *The Dexter Reference Model.* The Dexter Reference Model [Halasz and Schwartz 1994] is not a design model *per se*, but a generic model for hypertext systems. It consists of three layers. A storage layer describes the structure of the hypertext and how the components are connected. The "within-component" layer describes the structure of the components themselves and their contents. Finally, a run-time layer models the hypertext system as the user experiences it. The division in these three layers is important and may be found at the basis for most design models. Although the Dexter model in itself does not imply a method, the central role of the storage layer might at first sight favor data driven approaches; nevertheless UR-WSDM may also easily be mapped to these layers.

3.3.2 *HDM: The Hypertext Design Model.* HDM [Garzotto et al. 1993] introduced many new fundamental concepts into the world of hypermedia engineering. *Entity* and *entity type* became first-class citizens. HDM also pioneered *perspectives* to represent the same information in several alternate ways, but did not make the jump to different user classes as WSDM does. The concept of *access structures* was defined. These function as entry points for the application, helping the visitor to initially locate information of interest and therefore should not be confused with the more elaborate Navigation Tracks in WSDM or the Navigation Chunks in UR-WSDM.

3.3.3 *RMM: The Relationship Management Methodology.* RMM [Isakowitz et al. 1995], based on HDM, contains both a model and a powerful design methodology. The notion of *slices* allow to limit the view on information about "rich" entities with many attributes, and may be seen as a precursor to the *variants* in UR-WSDM and WSDM (called *perspectives* in the earlier paper [De Troyer and Leune 1998]). Isakowitz et al. however do not make use of these slices to explicitly model what the *user* wants to see of the entity. Their five-step method (E-R design, slice design, navigation design, conversion protocol design and user interface design) is data driven and quite completely covers the life cycle of a typical website, and has served as a basis for many subsequent methods. E.g. Takahashi has adapted RMM specifically for WBIS, and added *scenario analysis* to determine and specify how users interact with the planned WBIS [Takahashi 1997].

3.3.4 *WSDM: Website Design Method.* In [De Troyer and Leune 1998] the concept of a *user-centered approach* to web design was introduced and elaborated. As described above, this process starts with a user classification to identify the different kinds of users and a user class description that specifies the information requirements and characteristics shared by users of the same user class. WSDM also uses a different object model and navigation model, which are obtained by merging the different variant models of a user class and those of the different user classes, respectively.

3.4 Reuse and maintenance using UR-WSDM: a summary

WSDM takes the user centered approach and this has been found easy to use and teach. Recapitulating so far, in WSDM Object Modeling we first model the information in which users/visitors are interested. User classes are defined as specifying groups of users that satisfy a same set of identified user requirements. Navigation tracks (which express user requirements) define how to present them with the information. Scenarios describe the interaction between user and WBIS to manipulate data.

As sites designed with WSDM grow older and larger however, the problem of maintenance, redesign and hence reuse will pose itself. Since in WSDM the granularity for modeling is a single user class (possibly differentiated by variants), changes occur "a class at a time". However, when

modeling sites we noticed that as user classes evolve or become better circumscribed, often just one or a few identified requirements change. User class granularity makes it difficult to reuse parts from one user class to another.

In this paper we therefore argue that the finer granularity offered by the user requirements offers an approach fully compatible with "user-centrism" yet much better behaved for maintenance and reuse. Navigation chunks corresponding to these user requirements in UR-WSDM replace navigation tracks. Indeed, object model and navigation model corresponding to one requirement may be reused in different user classes. And the (formal) mapping of these user requirements onto navigation chunks will allow CAWE tools (*Computer-Aided Web Engineering*) based on UR-WSDM to control and manage the resulting redundancy, well out of sight or concern of designer and user.

Change in one or a few requirements impacts the original design gracefully, as user classes and object models change incrementally.

One may easily trace back design decisions to the requirement level, e.g. using a user-centered CAWE tool, under development at present.

APPENDIX

A. SAMPLE PROBLEM DECLARATION

You are to design a site for an Academic Department. A department has professors, students, technical and administrative personel. It offers graduate (Msc, PhD) and undergraduate (BSc) degrees.

Professors teach courses every semester, attended by students. They also participate in research projects. Professors produce results, sometimes in connection to a research project, sometimes not. Results are journal and conference papers, books, technical reports, software and hardware. Professors advise graduate students, who produce Master's and PhD theses. They also advise undergraduate students, guiding them in developing a graduation project .

Projects may be externally funded by sponsors, and are conducted in laboratories. The labs have computing equipment, and several professors and students may belong to it.

All degrees are obtained by taking a set of required courses and a set of elective courses. The total number of courses to be taken depends on the degree. In addition, a thesis or final project must be done for the degree to be granted. Quite frequently, student's theses are conducted as part of research projects carried out in labs.

Each course has a short description, a possibly empty set or pre-requisite courses, and a syllabus. For each course offering, the professor teaching it may add complementary material to the course, such as slides, workbooks, exercises, etc...

REFERENCES

Bieber, M. P. and Kimbrough, S. O. 1994. On the logic of generalized hypertext. *Decision Support Systems 11*, 241–257.

Boehm, B. W. 1981. *Software Engineering Economics*. Advances in Computing Science and Technology. Prentice Hall.

Chen, P. P.-S. 1976. The entity-relationship model : toward a unified view of data. *ACM Transactions on database systems 1*, 1 (Mar), 9–36.

De Troyer, O. M. F. and Leune, C. J. 1998. WSDM : A user-centered design method for web sites. In *Proceedings of the Seventh International World Wide Web Conference* (1998).

Gane, C. and Sarson, T. 1979. *Structured Systems Analysis: Tools and Techniques*. Prentice Hall, New York, NY.

Garzotto, F., Schwabe, D., and Paolini, P. 1993. HDM - A model-based approach to hypermedia application design. *ACM Transactions on Information Systems 11*, 1 (Jan), 1–26.

Gellersen, H.-W., Wicke, R., and Gaedke, M. 1997. WebComposition: An object-oriented support system for the web engineering lifecycle. In *Proceedings of the Sixth International World Wide Web Conference* (1997).

Halasz, F. G. and Schwartz, M. 1994. The Dexter Hypertext Reference Model. *Communications of the ACM 37*, 2 (Feb), 30–39.

Halpin, T. 1995. *Conceptual Schema and Relational Database Design*. Prentice Hall.

Hardmanand, L., Bulterman, D. C. A., and Van Rossum, G. 1994. The amsterdam hypermedia model. *Communications of the ACM 37*, 2 (Feb), 50–62.

Isakowitz, T., Stohr, E. A., and Balasubramanian, P. 1995. RMM: A methodology for structured hypermedia design. *Communications of the ACM 38*, 8 (Aug), 34–44.

Jacobson, I., Christenson, M., Jonsson, P., and Overgaard, G. 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley, Wokingham.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. 1991. *Object-Oriented Modelling and Design.* Prentice-Hall Inc. ISBN 0-13-630054-5.

Scherl, R., Bieber, M., and Vitali, F. 1998. A situation calculus model of hypertext. In D. R. Dolk Ed., *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences*, Volume 5 (1998), pp. 205–214. IEEE.

Schwabe, D. and Rossi, G. 1995. The object-oriented hypermedia design model. *Communications of the ACM 38*, 8 (Aug), 45–46.

Takahashi, K. 1997. Analysis and design of web-based information systems. In *Proceedings of the Sixth International World Wide Web Conference* (1997).

Verheyen, G. and van Bekkum, J. 1982. NIAM : aN Information Analysis Method. In T. Olle, H. Sol, and A. Verrijn-Stuart Eds., *Proceedings of the CRIS-1 Conference* (1982). Elsevier North Holland.