Vrije Universiteit Brussel
Faculteit Wetenschappen
Departement Informatica & Toegepaste Informatica
Academiejaar 2002-2003

Het gebruik van UML in WSDM

Frederik Uyttersprot <fuytters@vub.ac.be>

Proefschrift ingediend met het oog
op het behalen van de graad van
Licentiaat in de Toegepaste Informatica

Promotor: Prof. Dr. O. De Troyer

Samenvatting

De laatste jaren zijn zowel het aantal, de hoeveelheid informatie als de complexiteit van de meeste web sites enorm toegenomen. Vooral de toename in complexiteit van deze web sites heeft ertoe bijgedragen dat men is gaan nadenken over de design van deze informatie systemen. Onderzoek heeft enkele veel belovende design methodologieën met zich meegebracht.

Een van deze methodologieën is WSDM, of Web Site Design Method, ontwikkeld aan de Vrije Universiteit Brussel. Tot op vandaag is men steeds tewerk gegaan in WSDM met ORM als modeling tool. Deze thesis gaat na of UML aan waardige vervanger kan zijn als modeling tool voor ORM.

Deze thesis start met een inleiding over WSDM en gelijkaardige initiatieven en gaat dieper in op het gebruik van UML bij deze andere design methoden. Vervolgens worden via enkel voorbeelden het gebruik van UML in WSDM verder toegeligt – ook wordt er dieper ingegaan op de mogelijk voordelen dat UML als modeling tool met zich meebrengt.

Vrije Universiteit Brussel
Faculteit Wetenschappen
Departement Informatica & Toegepaste Informatica
Academiejaar 2002-2003

The use of UML in WSDM

Frederik Uyttersprot <fuytters@vub.ac.be>

Abstract

During the last few years the number, its content and the complexity of web sites has increased enormously. Due to this increase in complexity of those web sites people have started to do research on the design and the maintenance of these information systems. Today, several promising design methodologies already exist.

One of these methodologies is WSDM, or Web Site Design Method, created at the Free University of Brussels. Up till now ORM was used as modeling tool within all aspects of WSDM. This thesis is a first step to start using UML as the default modeling tool for WSDM.

The thesis starts with an introduction on WSDM and similar initiatives that exist today giving an overview of the use of UML in those methods. In a second part, the use of UML within WSDM is presented using a case study – in the end, several possible advantages of using UML as a modeling tool instead of ORM are presented.

Acknowledgements

I'd like to thank Prof. Dr. De Troyer for the changes she has giving me over the past few years and the patients she has shown. As following her courses, it was a pleasure to work with her.

Also, I'd like to thank my parents for the support over all those years, and the fact that they never stopped believing in me – even if some of those years where very difficult in several ways.

My partner in life, Sophie – she made me do this thesis and gave me the courage to go on until the end.

Dedicated to my daughter

- Anaïs -

Born on the 6th of June 2003

Table of contents

List of figures

Abbreviations

- ORM – Object Role Modeling
- UML – Unified Modeling Language
- WSDM – Web Site Design Method
- OCL – Object Constraint Language
- OO-H – Object-Oriented Hypermedia model
- UWE – UML-based Web Engineering
- OOHDM - Object -Oriented Hypermedia Design Method
- WebML - Web Modeling Language
- WAE - Web Application Extension

Introduction

This thesis contains two main parts, the background and the research part.

Background

The first part of this thesis provides information on which the second part is based. Firstly, we discuss the relevant differences between ORM and UML. Next, some important existing web site design methods are described, including their support for UML. Finally, WSDM is discussed in detail.

Research

The second part describes the research that has been done. First, a short problem description is given and a case is used to show UML can be used to replace ORM as a case tool. Next, some specific WSDM details are further worked out using UML. Finally, several advantages that arise when using UML in WSDM are discussed.

1 A view on two modeling notations

This chapter starts with a motivation for this text, then provides a short overview of ORM, which is currently used as modeling tool within WSDM, and UML, a widely used and accepted modeling tool that's constantly evolving. In the end a short discussion is given on ORM vs. UML based on several papers by Dr. Terry Halpin.

1.1 Motivation

One could wonder why this thesis was written. Up to now, WSDM used ORM as its main modeling tool with success and besides the basic foundation of WSDM, extensions to WSDM where successfully added, as we will see later on. Although, parts in WSDM use there "own" notations, which are build on top of ORM, ORM was good enough for WSDM up to this point. Still several points can be brought forward to motivate this text.

Popularity

Compared to UML, ORM is a lot less accepted outside the academic world. UML has become the de facto standard for object-oriented software modeling. Besides that, the explosive growth of the World Wide Web has raised the need for an UML extension[1] to model hypermedia applications.

Software modeling

More and more, web sites are seen as information systems based on a bigger collection of software elements with the need of a good design due to its complexity and size. Also maintenance becomes more and more an issue as the hypermedia applications grow. There for the ideas of "regular" software design are used or applied more and more within the design of a web site. Again, UML can play an important role when it comes to applying patterns within these applications.

Evolution

As we'll see later on, ORM seems to have several advantages over UML when it comes to data modeling. Even so, many are convinced that UML is powerful enough to cover all requirements that arise when modeling Web applications. For most of these requirements we can use the notation and diagrammatic techniques provided by "pure" UML. To cover the more special aspects of Web application design, one can define an extension or profile.

[1] Several examples will be discussed later in this text.

Besides that, UML is still evolving and several issues that currently exist (compared to ORM) will be solved in version 2. Version 2 is nearing completion at the time of this writing.

Standardization

One of the main issues regarding general acceptance of a design method is surely a standard modeling tool known by many. At this point, UML is very well known among developers around the world and a previous mentioned extension that's UML-compliant will be a much smaller barrier to start using any method then learning a whole new modeling notation.

Another important fact is that, thanks to the OMG group[2], UML is now a standard modeling notation supported by a lot of companies and individuals.

1.2 ORM – Object Role Modeling

As explained in [19][20] Object-Role Modeling (ORM) is primarily a method for modeling and querying an information system at the conceptual level. In Europe, the method is often called NIAM (Natural language Information Analysis Method). Since information systems are typically implemented on a DBMS that is based on some logical data model (e.g. Relational, object-relational, hierarchic), ORM includes procedures for mapping between conceptual and logical levels. Although various ORM extensions have been proposed for process and event modeling, the focus of ORM is on data modeling, since the data perspective is the most stable and it provides a formal foundation on which operations can be defined. For correctness, clarity and adaptability, information systems are best specified first at the conceptual level, using concepts and language that people can readily understand. Analysis and design involves building a formal model of the application area or universe of discourse (UoD). To do this properly requires a good understanding of the UoD and a means of specifying this understanding in a clear, unambiguous way. Object-Role Modeling simplifies this process by using natural language, as well as intuitive diagrams that can be populated with examples, and by expressing the information in terms of elementary relationships.

1.3 UML – Unified Modeling language

Presented in [16][21][22] the Unified Modeling Language (UML) is a modeling language for specifying, visualizing, constructing, and documenting the artifacts of a system intensive process.

[2] http://www.omg.org/uml/

It was originally conceived by Rational Software Corporation and three of the most prominent methodologists in the information systems and technology industry, Grady Booch, James Rumbaugh, and Ivar Jacobson (the Three Amigos).

The language has gained significant industry support from various organizations via the UML Partners Consortium and has been submitted to and approved by the Object Management Group (OMG) as a standard (November 17, 1997).

The UML is a modeling language for specifying, visualizing, constructing, and documenting the artifacts of a system intensive process.

- Within a system intensive process, a method is applied as a process to derive or evolve a system. Subject and express knowledge (syntax) regarding the subject for the purpose of communication. The subject is the system under discussion.
- As a modeling language, it focuses on understanding a subject via the formulation of a model of the subject (and its related context). The model embodies knowledge regarding the subject, and the appropriate application of this knowledge constitutes intelligence.
- Regarding unification, it unifies the information systems and technology industry's best engineering practices across types of systems (software and non software), domains (business versus software), and lifecycle processes.
- As it applies to specifying systems, it can be used to communicate "what" is required of a system, and "how" a system may be realized.
- As it applies to visualizing systems, it can be used to visually depict a system before it is realized.
- As it applies to constructing systems, it can be used to guide the realization of a system similar to a "blueprint".
- As it applies to documenting systems, it can be used for capturing knowledge about a system throughout its lifecycle's a language, it is used for communication.

1.4 A comparison

Extensive discussions and comparisons are given in [32][33][34][35]. In the end, the conclusion comes down to the simple fact that both modeling notations serve different purposes. Still, the designer can gain several benefits by using both notations at different stages during the design of the application.

For this text it's interesting to see what the weaknesses are of UML when compared to ORM. The main problem(s) arise during data modeling, which can be considered an important part of web application design. Since web sites have become "online" data repositories, one can't ignore this issue.

From the discussions mentioned above, one can identify the following principles for evaluating modeling languages and applied them in evaluating UML and ORM for conceptual data modeling: expressibility; clarity; semantic stability; semantic relevance; validation mechanisms; abstraction mechanisms; formal foundations.

Although ORM's richer constraint notation makes it more expressive graphically, both methods extend expressibility through the use of textual languages – in UML this is done by using OCL.

ORM scores higher on clarity, because its structures may be directly verbalized as sentences, it is based on fewer constructs, and it models semantic connections across domains. Being attribute-free, ORM is more stable for both modeling and queries. ORM is easier to validate. Both methods are amenable to similar abstraction mechanisms, and have adequate formal foundations.

UML class diagrams are often more compact, and can be modeled with a vast array of implementation detail for engineering to and from object-oriented programming code. Moreover, UML includes mechanisms for modeling behavior, and its acceptance as an OMG standard is helping it gain wide support in industry, especially for the design of object-oriented software.

Thus both methods have their own advantages. For data modeling purposes, it seems worthwhile to provide tool support that would allow users to gain the advantages of performing conceptual modeling in ORM, while still allowing them to work with UML.

Research is currently being done to add transformations between ORM and UML. Once this support is widely available, studies are planned to study why and how designers choose and/or integrate modeling methods in practice.

2 Overview of existing web site design methods

In this chapter, we discuss some existing web site design methods. For each method, the approach and the different phases are described. Because this thesis is about the use of UML in a web site design method as modeling notation, we shortly discuss the support of UML in each method.

2.1 OO-H - Object-Oriented Hypermedia Method

OO-H as presented in [11][12] extends traditional software engineering approaches, based on UML, with two new models, namely, the navigation model and a presentation model.

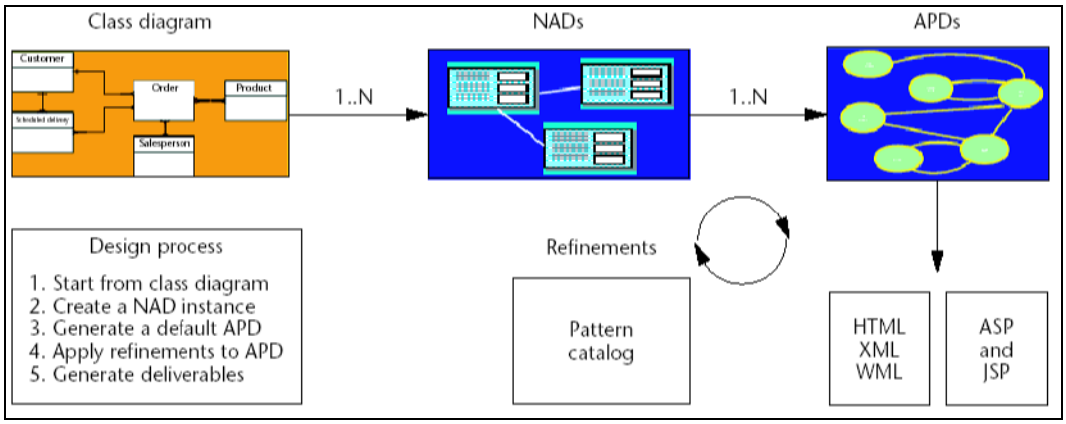

Figure 1 - OO-H design process

Navigation model

The navigation model is defined by one or more navigation access diagram (NAD). The NAD extends the domain view provided by an UML use case and class diagram with navigation and interaction features. There should be as many NAD's as there are views in the system. A view is required for at least each user-type who is allowed to navigate trough the system. The NAD is based on four constructs,

- Navigation classes, are enriched class diagrams; the visibility of the attributes and methods are restricted according to the users access permissions and navigation requirements of the user
- Navigation targets, they group the elements of the model that collaborate in the coverage of each user navigation requirement
- Navigation links, define one or more paths a user can follow when surfing the web site – links can have both navigation patters and/or filters associated (OCL expressions) – six link types are defined
- Collections are (possibly) hierarchical structures defined on Navigation Classes or Navigation Targets. They provide the user with new ways of accessing the available information

Presentation model

The presentation model is created after the navigation model and is built out of one or more abstract presentation diagram (APD). An APD specifies the visual appearance and page structure of the system. A default APD can be automatically created from the NAD diagrams. Refinements and the application of an APD related pattern catalog would produce more useful APD diagrams.

Support for UML

OO-H departs from basic UML use cases and class diagrams to define the domain information structure. With this information it then creates the NAD and the APD using its own notation (or extension). OO-H also uses the UML-compliant OCL language to define constraints, which are associated to the NAD by means of filters. To conclude, OO-H is partly UML-compliant.

2.2 UWE - UML-based Web Engineering

UWE as described in [13][14][15][18] is an object-oriented, iterative and incremental approach based on the Unified Modeling Language [16] and the Unified Software Development Process [17]. The notation used for the models is an UML profile (extension).

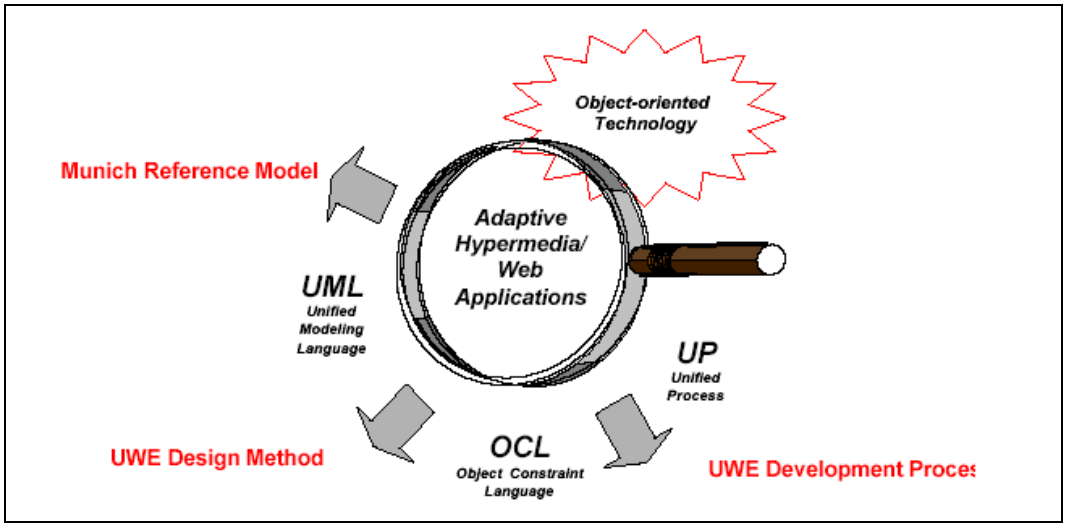

Figure 2 - UWE overview

The modeling activities are the requirements analysis, conceptual, navigation and presentation design. In this work task modeling is included to model the dynamic aspects of the application.

Conceptual modeling

UWE proposes use cases for capturing the requirements. Conceptual modeling is based on these use cases; a conceptual model includes the objects involved in the typical activities users will perform with the application.

Navigation modeling

Navigation modeling activities comprise the specification of which objects can visited by navigation through the Web application and how these objects can be reached through access structures. In the end two models are specified: the navigational class model and the navigational structure model.

The navigational class model defines a view on the conceptual model showing which classes of the conceptual model can be visited through navigation in the application. This model is built with a set of navigational classes and associations, which are obtained from the conceptual model.

The navigational structure model is based on the navigational class model. It defines the navigation structure of the application, i.e. how navigational objects are visited. Additional model elements are required to perform the navigation between navigational objects: menus, indexes, external nodes and navigational contexts[3].

OCL Constraints are used to express the relationship between conceptual classes and navigation classes or attributes of navigation classes.

Presentation modeling

The presentation modeling describes where and how navigation objects and access primitives will be presented to the user. Presentation design supports the transformation of the navigation structure model in a set of models that show the static location of the objects visible to the user.

Support for UML

Because UWE is presented as an UML profile, and this profile is used throughout the whole design cycle, we can conclude that UWE is UML-compliant.

2.3 OOHDM - Object -Oriented Hypermedia Design Method

According to OOHDM [6][7], the development of web-based information systems is presented in four-step process, which is performed in a mix of iterative and incremental styles of development; in each step an object-oriented model is built or enriched.

[3] Navigational contexts have been introduced by OOHDM to permit grouping of navigational objects by user requirements.

Conceptual Design

In this phase a model of the application domain is built using well-known object-oriented modeling principles with a notation similar to UML. The model consist s of classes, subsystems and relationships, which are built using aggregation and generalization/specialization hierarchies.

The most important in this phase is that the domain semantics must be taken as neutral as possible; users and tasks are not yet to be considered. OOHDM doesn't demand to use a particular method to produce the conceptual class schema; any of the well-known methodologies (OMT, UML) may be employed.

Navigational Design

This phase describes the navigational structure of a hypermedia application in terms of navigational contexts, which are induced from navigation classes such as nodes, links, indices, and guided tours.

Navigational contexts and classes take into account the types of intended users and their tasks. Nodes in OOHDM represent logical "windows" (or views) on conceptual classes defined during domain analysis.

Different navigational models may be built for the same conceptual schema to express different views on the same domain. Links are derived from conceptual relationships defined in step 1. By defining the navigational semantics in terms of nodes and links, we can model movement in the navigation space independently of the conceptual model.

Abstract Interface Design

In this phase an abstract interface model is built. During abstract interface design, the interface objects which the user will perceive are defined: how will the navigational objects look like, which interface objects will activate navigation and so on.

Thanks to a strict separation between abstract and navigational interface design, multiple (different) interfaces may be build using the same navigational model.

ADV (Abstract Data View) is used to describe the user interface, they're formal models for interface objects. They define how the interface objects are structured, how they're related to the navigational objects and how they react to external events. Interface objects are aggregations of (primitive) classes like buttons and text fields.

Implementation

This phase maps interface objects to implementation objects available in a chosen implementation environment. This phase is very different from the previous ones due to the fact that all the previous phases where independent of the implementation platform.

Support for UML

During conceptual modeling, an UML-like notation is used. OOHDM itself does not demand the use of a particular method to produce the conceptual model and leaves the choice to the designer. In the past, support for UML was very limited but in the last revision of OOHDM [8], a stronger focus on UML is presented and describes an extension of UML.

2.4 WebML - Web Modeling Language

WebML as presented in [9][10] defines a model-driven approach allowing the developer to define a web site on a conceptual level. WebML provides a graphical notation and an XML-based syntax. The last feature can be used within code generators for automatically producing the implementation of a web site.

The specification of a site in WebML consists of four perspectives,

Structural model

The structure model defines or expresses the content of the web site in terms of relevant entities or relationships. It therefore uses classical notations such as the E/R model, the ODMG object-oriented model or UML class diagrams for data modeling. To have access to calculated information a simplified OQL-like query language is offered.

Hypertext model

It describes one or more hypertexts that can be published in the site. Each hypertext defines a site view. A site view in turn has two sub-models; the composition model specifies the pages build from the hypertext and the content units within a page. The navigational model defines how pages and content units are linked together from the hypertext.

Presentation model

The presentation model defines the look and layout of the pages by using an abstract XML-syntax – this allows this process to be independent from the output device or rendering language. Presentation information can be page-specific or generic; page-specific specifications define the presentation of a specific page, including explicit references to page-content. Generic specifications are based on predefined models that are independent of the specific page and include references to generic content elements.

Personalization model

This part defines a model of the users and user groups, specified in terms of predefined entities called User and Group.

Support for UML

WebML shows limited support for UML. As presented in [9] UML can be used for defining the application requirements and for data modeling within the structural model. Later during design, the self-defined WebML-OQL language is used. The hypertext model itself is done using a special graphical notation.

2.5 WAE – Web Application Extension

WAE [1] is an exception, because WAE itself is not a web site design method. It's added to this overview because WAE provides an extension, or profile to UML similar as in UWE. Although WAE provides us a way to represent Web pages and other architecturally significant elements in UML it does not provide a separation between navigation and presentation aspects. Neither does it provide a way to think about the way web engineering should be done. On [2] a roadmap and some recommendation are defined but these can't be seen as a design method.

Support for UML

Because WAE itself is a UML profile, UML support can be seen as complete, but as mentioned before, WAE itself is not a web site design method.

2.6 Other existing methods

Several other design methods exist (e.g. HDM, RMM,) but none fully support UML throughout the design cycle. Another extensive comparison can be found in [3] which shows that support for UML is mostly limited or non-existing in most currently available web site design methods.

Still, several methods are being redefined and we see support and interest for UML (pure or as an extension) grow – the best example up-to-know is UWE. For more details on those design methods mentioned above please refer to [4][5].

3 WSDM - Web Site Design Method

After considering other design methods, we now give description of WSDM [36][37][38]. Because WSDM is the methodology on which the research is based, this description is more extensive than the sections of chapter 2. First, we discuss the most characteristic elements of the method and after that, we describe all WSDM phases in detail using a case study that provides an initial example of using UML in WSDM.

3.1 Characteristics

Audience driven

WSDM has an audience-driven approach in contrast to most other web site design methods, which are data driven. The audience-driven approach takes the requirement s of the users of the web site as a starting point and uses this as basis for the structuring of data afterwards. Data-driven methods take the data of the organization as a starting point: they consider how to structure the web site based on the data.

Audience classes

A website has different 'kinds' of users, e.g. for the web site of a university we may distinguish the following users: professors, student s, future students. Different users have different requirement s, sometimes even different presentational needs, e.g. children will appreciate a flashy presentation, while bank directors maybe prefer a sober and clear interface. WSDM classifies the users of a web site into audience classes: each audience class has it s own requirement s and characteristics. This is reflected in the context (not all info for all users) and in the interface (language, jargon, look-and- feel, etc.). The division into audience classes has as an advantage that the site is more adapted to each user's needs, which enhances the usability and user satisfaction.

Explicit conceptual design phase

The conceptual design, free of implementation details, is explicitly separated from the implementation design, like using a specific implementation language, grouping into pages, use of menus, etc. Hence we are no longer dependent of the current technology (e.g. HTML). Moreover, it is possible to offer different presentations on different devices, e.g. Palm Pilot, PC, WAP, etc.

3.2 An overview

Figure 3 gives an overview of the WSDM method. The first step is to specify the Mission Statement. The Mission Statement Specification

expresses the purpose and subject of the web site, and declares the target audience. Based on the Mission Statement, the Audience Modeling is performed, in two steps: Audience Classification and Characterization. During the Audience Classification phase, the different types of users are identified, while Audience Characterization consists of defining the characteristics of the members of the different Audience Classes.

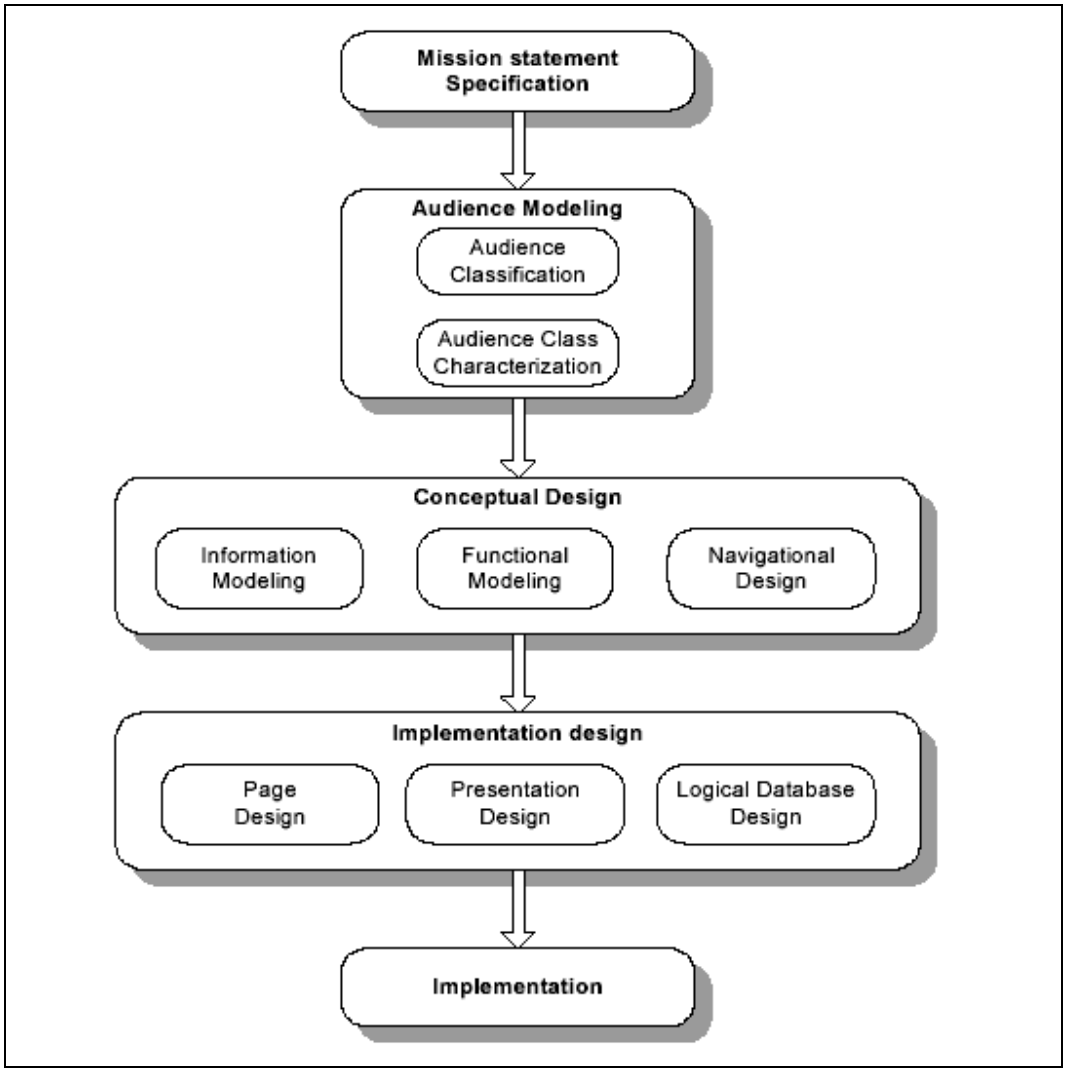

Figure 3 - WSDM overview

Next, we perform Conceptual Design, in three steps: Information, Functional and Navigational Modeling. During Information Modeling we observe what kind of information is apparent, while in the Functional Modeling phase we observe the functionality. The Navigation Design phase, we consider the global navigation through the information and the functionality.

The next phase, Implementation Design, performs Page, Presentation and Logical Database Design. During these phases, grouping in pages, specifying the look and feel, and designing the database are done. After all design phases done, the web site can be implemented.

All mentioned phases are discussed in detail below. As illustration of the different steps, and because we like to compare the use of UML to that of ORM in WSDM, we'll use the same subject as in [36] – the conference review system, based on [39].

The discussion is mainly based on the use of UML and not on the design method itself.

3.2.1 Mission Statement

In the mission statement, the purpose and subject of the project or web site is defined and the target audience(s) is declared.

For the conference review system, the mission statement is formulated as follows: "support the overall selection process of papers for a conference – this includes submission of authors, evaluation and selection of the committee".

Purpose → the site has to support the paper selection
Target → authors and committee
Subject → papers for a conference

In this first step, no graphical notations are required, and therefore no UML is used here.

3.2.2 Audience modeling

This phase contains two main steps, the audience classification and the audience characterization. Audience classes are defined using the user's information – and functional requirements. A single user can be part of multiple (different) audience classes.

3.2.2.1 Audience classification

Classification identifies the different types of users that will use the web site. This is done based on their "activities" and activities involve people. Depending on what the user will do with the web site will define his/her audience class.

The activities are,

- → Paper submission
- → Paper assignment to reviewers
- → Paper assignment to PC members
- → Entering a review
- → Selecting a paper
- → Notify author

The people involved in these activities are,

→ Author
→ PC Chair
→ PC Member
→ Reviewer

We could see an audience class as a role that a user has within the system. A role gets several actions associated. In UML this can be modeled using use case diagrams. The main ingredients for this type of diagram are use cases and actors, respectively the roles that users can take towards a system. They are associated to the tasks, these are use cases, and they are involved in.

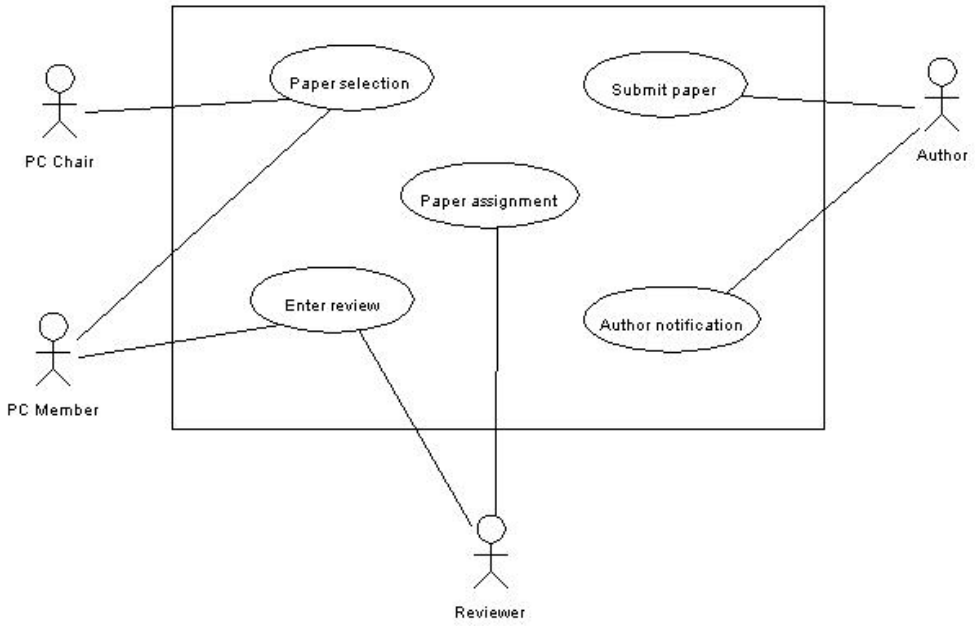

Figure 4 - Activity diagram

Also modeled in this phase is the audience class hierarchy. All described classes are sub-classes of one top class – the visitor. The actors that are part of the use case diagram can model this functionality for us as seen in Figure 5.

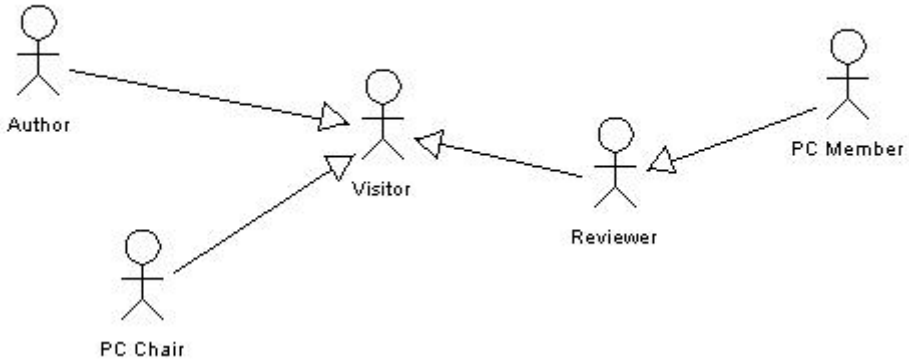

Figure 5 - Audience class hierarchy

Because this site has a protected area, we define a second hierarchy – based on the security aspects needed by the system and user. Here we can also define use cases defining the actions taken by the respective actors.

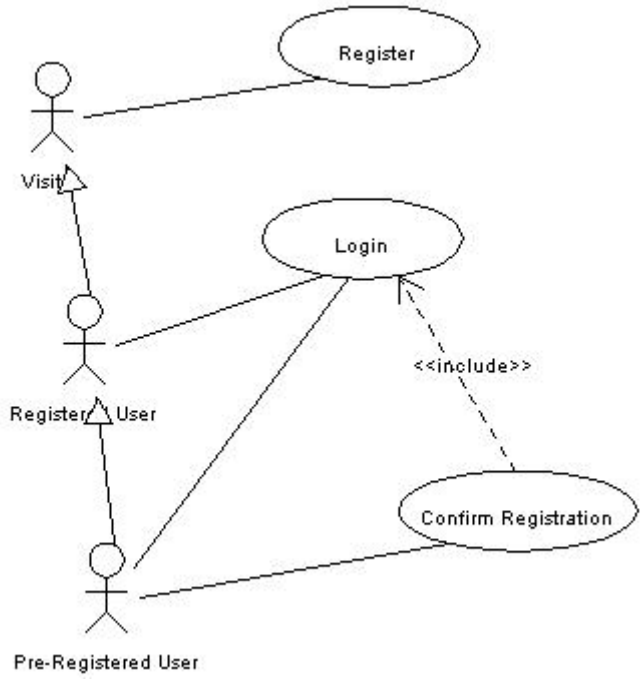

Figure 6 - Security use case

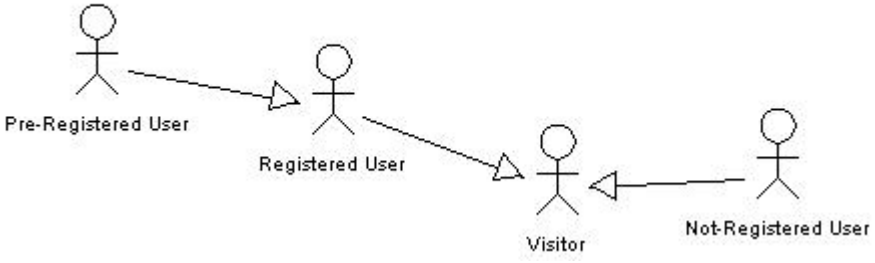

Figure 7 - Security hierarchy

A relation exists between the classes from the audience class hierarchy and from the security class hierarchy.

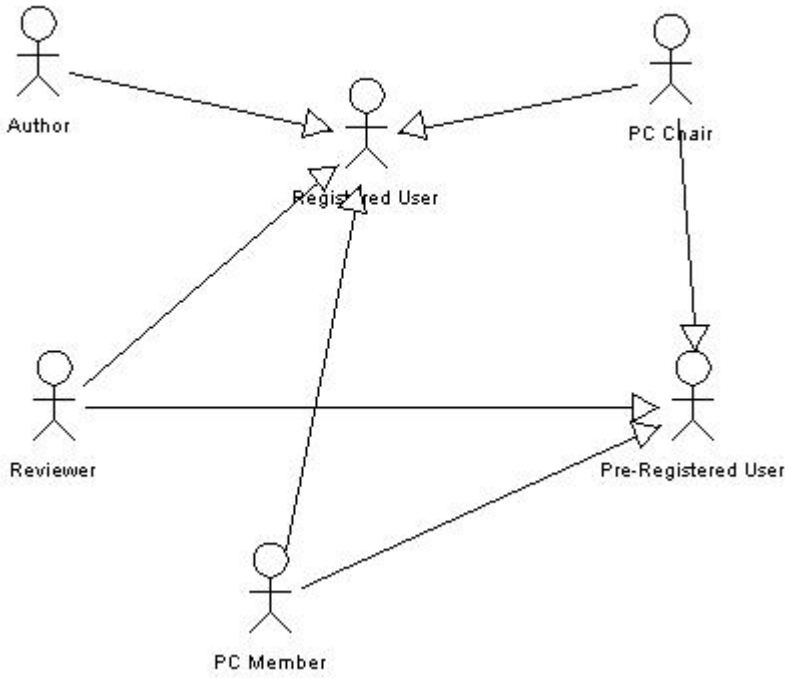

Figure 8 - Audience and Security class relations

3.2.2.2 Audience characterization

In this step, the different audience class characteristics are defined. Those characteristics define usability requirements. They may for instance describe how information should be presented to a certain audience class. Modeling these characteristics at this point is not useful, because they don't add any value to the model – they only add additional requirements to the audience classes. Later during the implementation phase, these requirements can become important.

These requirements can also be used when building an adaptation model in combination with a user model as explained in 4.1.

3.2.3 Conceptual design

This phase contains three big parts, information modeling, functional modeling and navigational modeling. This phase concentrates on the "what and how" and not on the visual aspect of web site design.

Due to the size of this part, we'll only show parts of the model.

3.2.3.1 Information modeling

The purpose of this step is to model the structured data behind these so-called data intensive web sites. This is done by modeling the information requirements of the different audience classes into so-called information chunks. An information chunk is a conceptual schema that describes the information needed to satisfy a single information requirement.

In UML data modeling is done using class diagrams.

As an example we show the information requirement of the author audience class. The informal description is,

“Information about the author’s submission”

This can be described in the following elementary requirements,

1. “Give paper-id and title of all papers submitted by author as main or co-author and allow a user to make a selection” – shown in Figure 9
2. “For a paper give, paper-id, title, abstract, main author, co-author(s), track, subjects (if available) and the file containing the paper” – shown in Figure 10

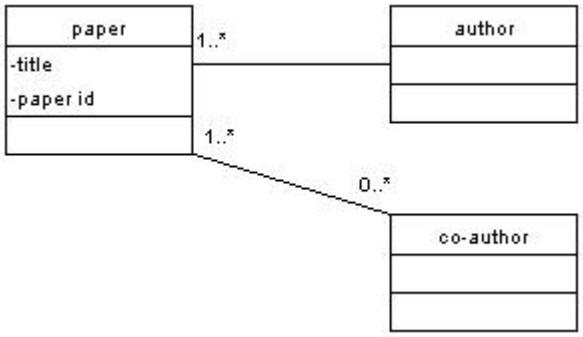

Figure 9 - Information chunk, AuthorSubmission

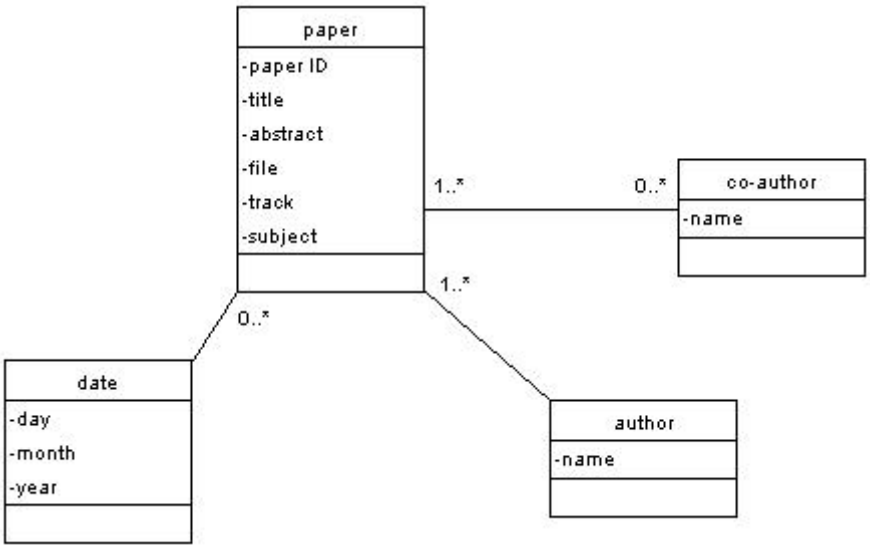

Figure 10 – Information chunk, SubMissionInfo

As the reader can notice, a lot of details have been omitted. UML class diagrams allow a designer to define a lot more detail like parameter types, allowed methods, etc.

3.2.3.2 Functional modeling

This step is very similar to the previous one, expect now, the functional requirements are modeled using functional chunks.

3.2.4 Navigational design

The navigational model describes the structure of the web site and the model how the members of the different audience classes will be able to navigate through the site. For each audience class a navigation track is created. All tracks form the navigational model.

WSDM defines tracks, components and links. On a link a condition can be placed. To model this using UML we can use a state diagrams combined with guards. As an example the navigation track for a registered user is shown in Figure 11.

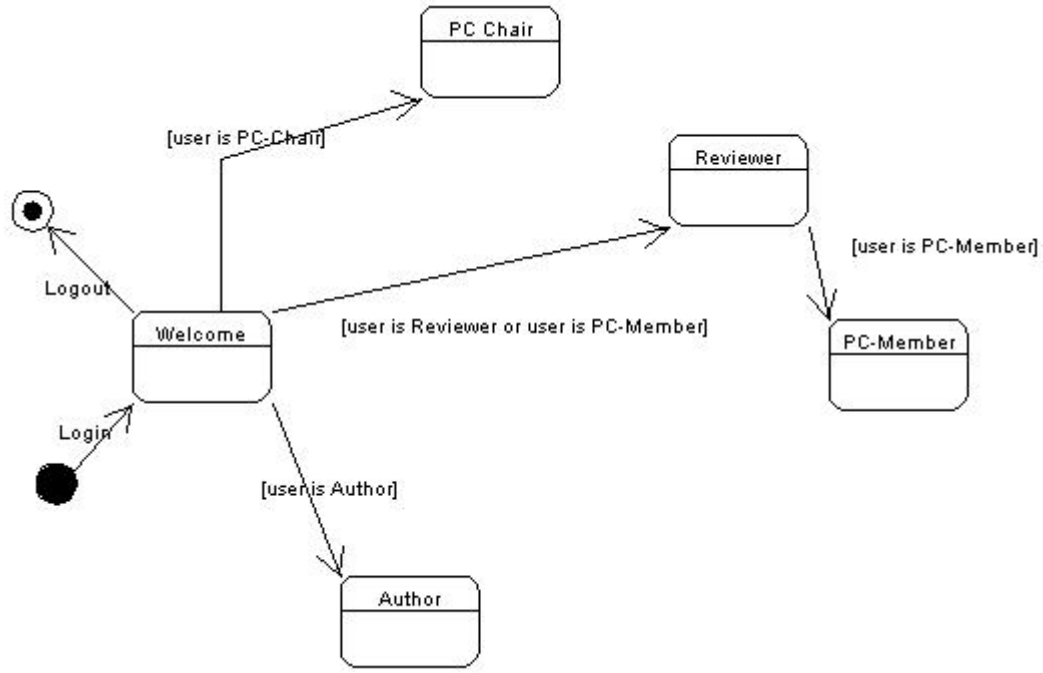

Figure 11 - Registered user navigation track

Figure 12 shows a navigational model for the audience class Author.

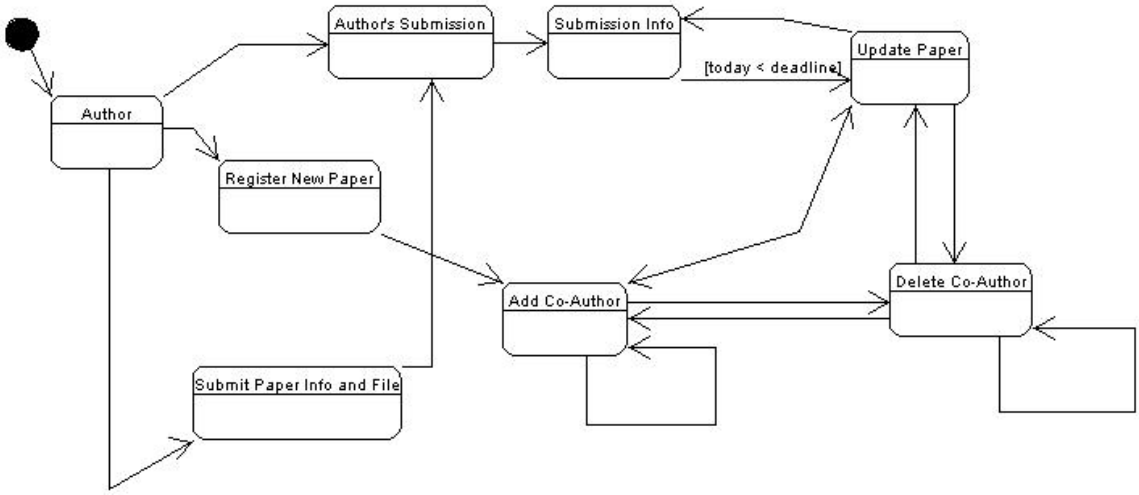

Figure 12 - Author track without chunks

3.2.5 Business Information model

The business information model describes the information needed by the different audience classes in an independent way. In fact it is a conceptual model of the information available in the system. This information of model may already be available within the organization – if not, or if the shape is not usable for our purpose it must be developed.

Here as during the information modelling phase, UML class diagrams are proposed. They allow us to model the audience classes and there attributes and they allow us to define or model the relations with the other information (chunks). As an example, the relation between a user, his role and the login is given. As mentioned in 3.2.2.1 any user who has a role (who is able to perform any activities) has to be a (pre-) registered user and therefor requires a login.

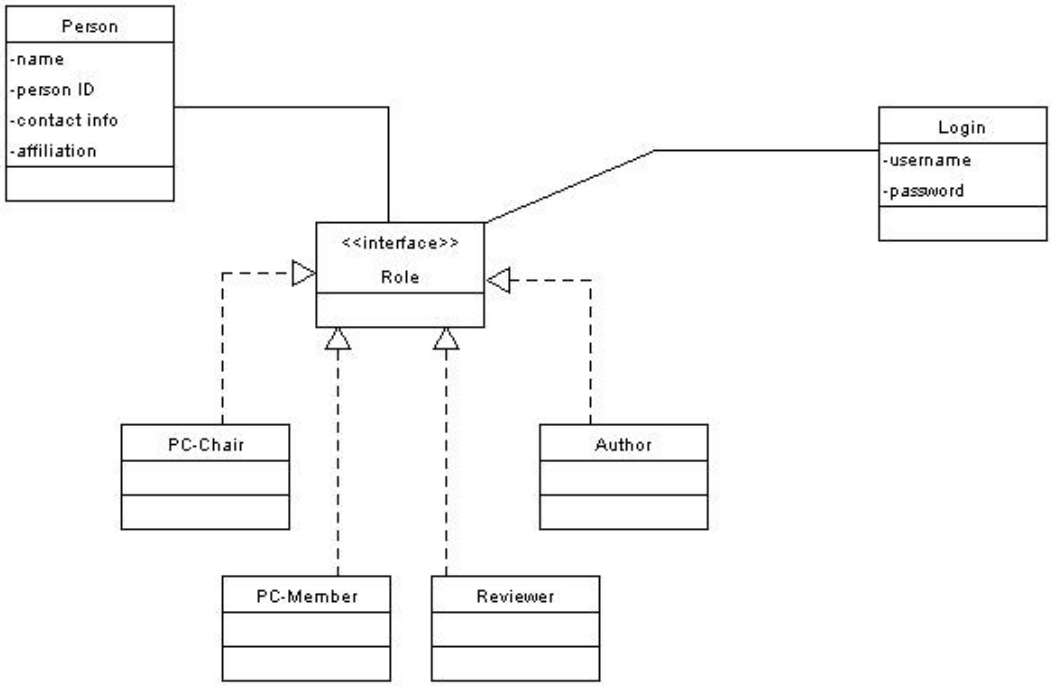

Figure 13 - Person, Role, Login business model

3.2.6 Implementation model

This phase takes up the following three parts; Page design, Presentation design and Logical database design. As mentioned in [36] is this phase still open for research, still some ideas will be given concerning the use of UML during this phase.

3.2.6.1 Page design

The page structure is derived from the navigational model. For the components and links in the navigational model we define pages and hyperlinks. By default we can assign each component to a page and each link to a hyperlink.

Due to the fact that the page structure is based on the navigational model, one can design this using UML sequence diagrams. A sequence diagram is a very easy to understand visualization of single scenarios or examples of business processes regarding their behavior in time. It focuses on when

the individual objects interact with each other during execution, and therefore essentially includes a time-line. The other feature that comes with a sequence diagram is that it defines the “flow” within the web site a user can follow. A quick example shows a login sequence for a registered user.

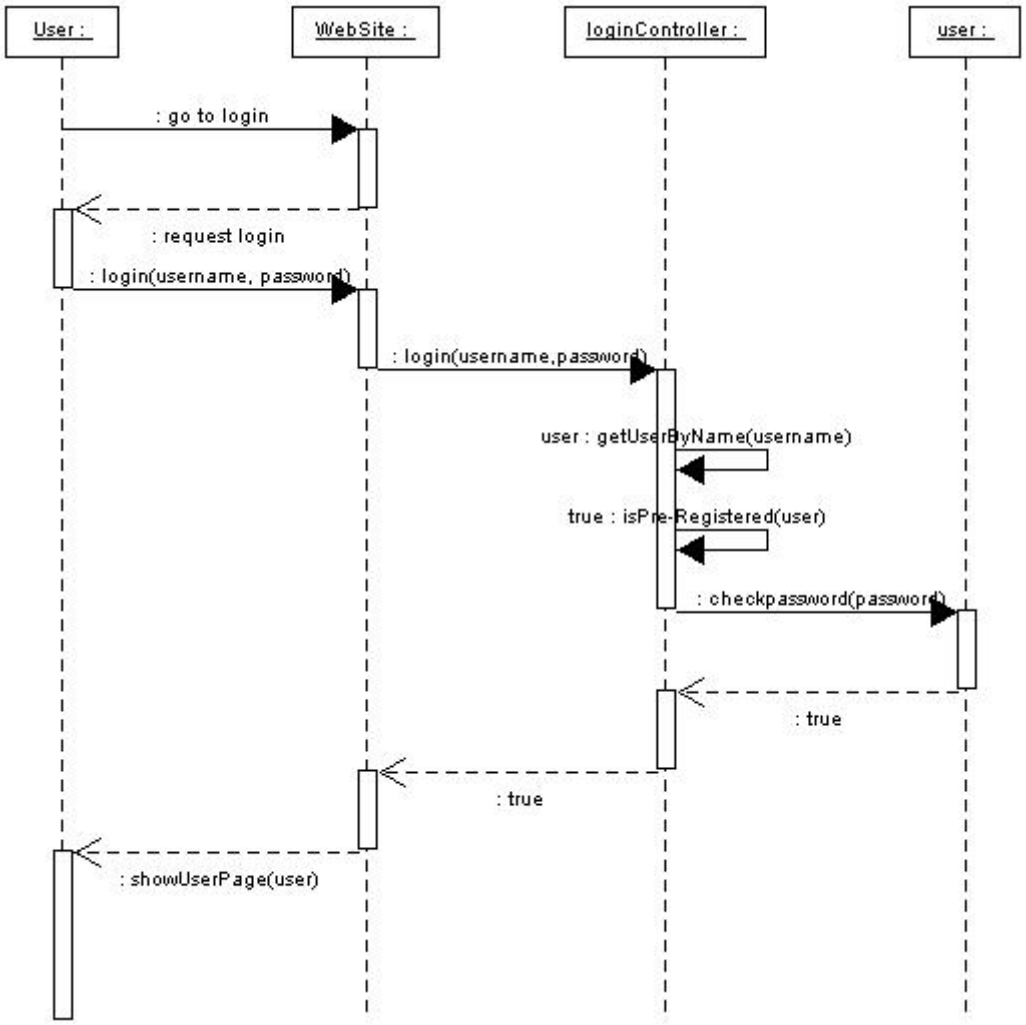

Figure 14 – Login flow for registered user

Based on these flow-charts, one can define what screens should be shown and when they should be shown. The example also shows any underlying software components that might be needed, these are parts of the page design that are not shown but are vital to the whole structure.

3.2.6.2 Presentation design

The presentation model should enable the designer to take the characteristics of the different audience classes into consideration. This may result in different styles for the same navigation track. Pure UML is not capable of doing this. This can only be done when extending UML as discussed in 5.2. Also, one could argue if this even has to be modelled

using UML. Another way could be using business rules to define the look and feel of the web pages using style sheets for example.

3.2.6.3 Logical database design

As presented in [40][41] again UML is found suitable to build a data model. This allows a designer to map the database structure to UML class diagrams. The proposed profiles allow the support of software development and data modeling with one unified language.

3.3 Implementation

This phase will create the actual construction of the web site. Depending on the chosen implementation environment one will be able to use semi-automatic code generators. There exist several case-tools that use UML as a modelling notation and have support for automatic code generation, see Chapter 5 for more information on this topic. Most of these tools generate code based on an object-oriented language like Java. Those languages itself provide a lot of functionality and possibilities when it comes to web based projects.

4 Extensions

Since the first definition of WSDM, several additional features have been added to WSDM. One addition is discussed in this chapter to show more possibilities of UML as a modeling notation when used in WSDM.

4.1 Adaptive behavior

Presented in [23][24], syntax for adaptive behavior was added to WSDM as part of the design cycle. This was done by changing the navigational design and adding rules to define this behavior. This text will not provide a similar proof as in [24] but an almost equal solution can be given using OCL, which is part of UML. OCL is the standard syntax for defining additional constraints on the UML notation and is there for suitable for this task.

The idea of using OCL for this issue is not solely mine and has been proposed already a few times in the past. In [14] an adaptation model is proposed. This model, based on pre-defined rules, is build on top of UML collaboration diagrams. Three rule types are defined,

- Construction rules – finding the correct concepts
- Acquisition rules – acquiring information from/about the users and update the user model
- Adaptation rules – adapting content, link and/or presentation

In [24] adaptive behavior is mainly modeled on a base of “number of visits” to a certain part of the web site. Based on this, one can use the following rules[4],

- Promotion and demotion
- Linking and unlinking
- Clustering

The above three can be grouped as adaptation rules and therefore a similar solution is proposed as in [14] using an UML collaboration diagram.

[4] For more details on this classification please refer to [24].

5 Features

This chapter discusses some additional features that come for “free” when using UML as a modeling notation. Some of these features already exist for standard software design and some are currently being developed or researched on.

5.1 Code generation

Semi – or full automatic code generation is one of the features of today’s existing UML case tools, e.g. [26][27]. Most tools are able to generate full code listings and some are even capable of reverse-engineer code back to UML diagrams.

Today research is being done on adding this feature to web design methods like UWE. As mentioned earlier UWE is based on an UML profile, which is used in the case tool ArgoUWE [28] which itself is based on ArgoUML [29]. Considering this research, one could easily see this being used in the near future for WSDM – when UML is chosen as a design notation.

This and other initiatives [30] are based on UML’s own XMI format. XMI [31] is a model driven XML Integration framework for defining, interchanging, manipulating and integrating XML data and objects. XMI-based standards are in use for integrating tools, repositories, applications and data warehouses. Provides rules by which a schema can be generated for any valid XMI-transmissible MOF-based Meta model.

5.2 UML profile

As shown in chapter three, “pure” UML can already fill a lot of requirements for web site design. But UML can be taken one step further as shown in several initiatives discussed in chapter 2. An UML profile allows the designer to define special stereotypes allowing him to create a better model of the system.

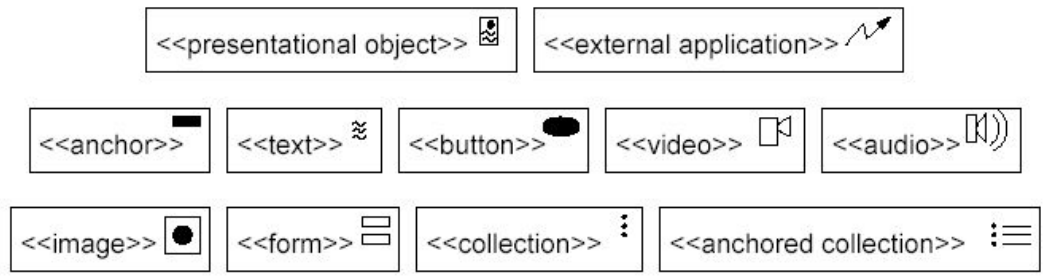

Figure 15 - Example of UML profile

An example is taken from UWE (Figure 15 and 16). Concerning web modeling, stereotypes for HTML elements can be created. This allows the designer to create a better presentation model for example.

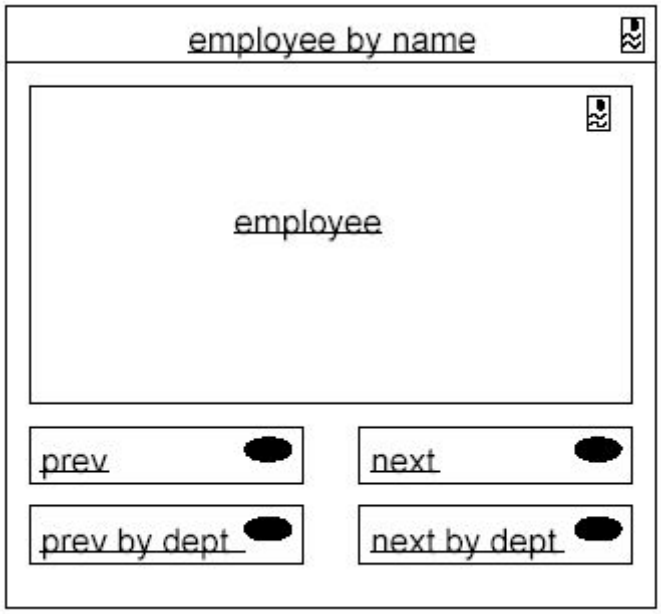

Figure 16 – Presentation model for "employee by name"

5.3 Design Patterns

Because UML is focused towards object-oriented programming, design patterns have been researched and defined in UML already many times in the past. Still it's one of the strong points of UML when looking at a hypermedia application as a normal software application. Design patterns were first defined as solutions to recurring architectural problems. Similar problems or solutions can be provided for within web applications and allow the designer to create a "cleaner" design and code.

An example is given as illustration[5]: Observer pattern

Problem: Suppose you are building a system to handle auctions. In any given auction, you have one object to represent the Item, and any number of objects to represent the Bidders. When the price of the auction changes, how do you notify each Bidder of the new price without coupling the Bidders too closely to the Items?

Solution: The Observer pattern can be applied in a situation such as this. This pattern proposes a collaboration where changes in one object (the subject) cause notifications to be sent to other objects (the observers). A

[5] Taken from http://www.jepstone.net/papers/web_uml_patterns/slides/toot-1.html

typical way to implement Observer is to define a Subject and Observer class, and let your implementations inherit from those classes.

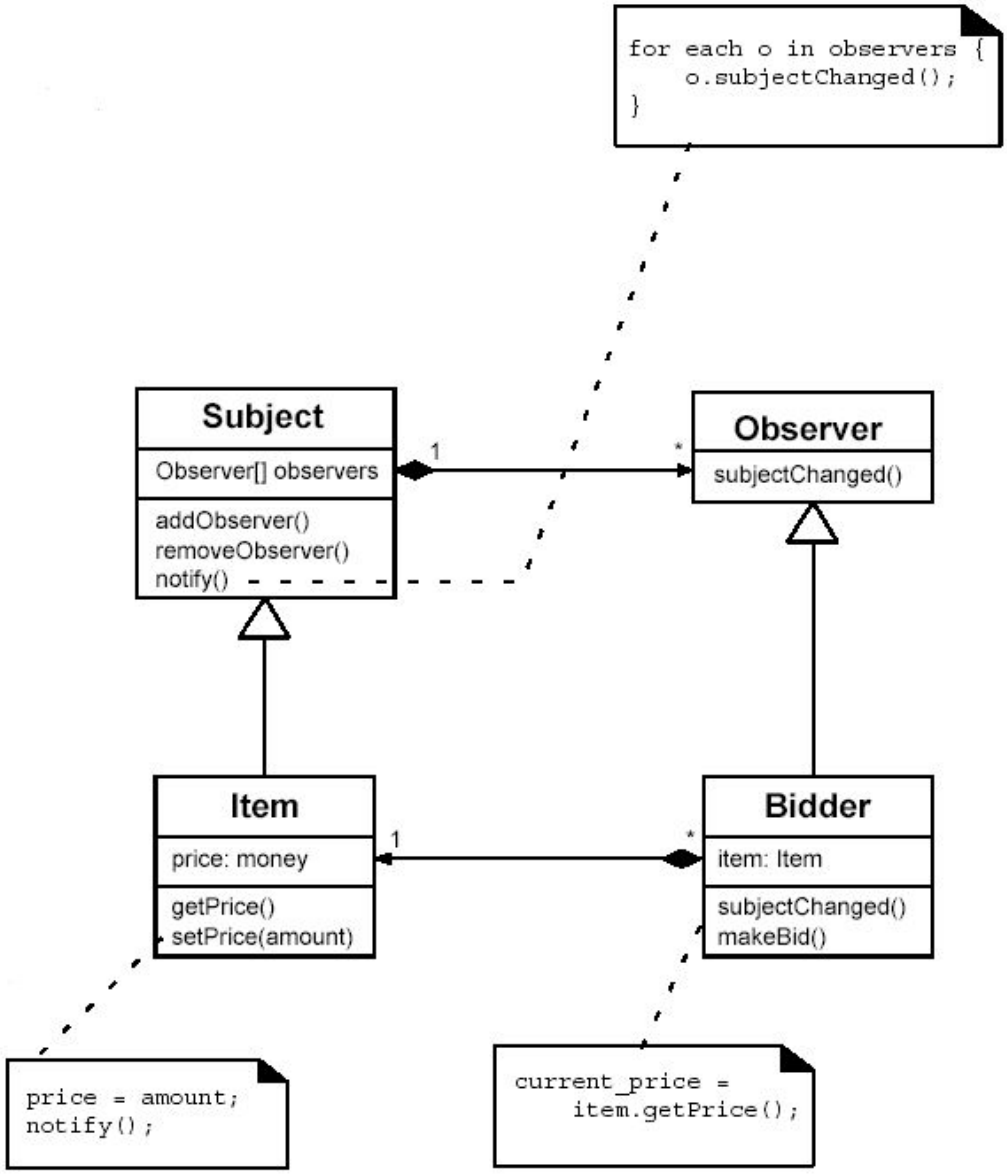

Figure 17 - Observer design pattern

Subject: The Subject class should define the following methods:

AddObserver(Observer): Attaches an observer
DropObserver(Observer): Removes an observer
notify(): For each observer, send a subjectChanged() message

Observer: The Observer class only needs a subjectChanged() method. One way to implement this is to define a class, such as Bidder, as a subclass of Observer. It's up to you to define how the Observer will respond to the subjectChanged() message. More likely than not, the Observer will invoke one of the subject's

methods to obtain the value of any attributes that may have changed (such as the price of an item in an auction).

Several larger and web site related projects exists that are using patterns at the very center of their design, like [42] where the well-known MVC (model-view-controller) design pattern has been used.

6 Conclusion and future work

This should be seen as a first step towards using UML in WSDM. A lot of ground still has to be covered and will probably take another few years before a similar level is reached as today with ORM. Still, due to all the parallel initiatives that exist today one could use a lot of work that's currently done by others. For example, the UML profile developed for UWE will surely prove to be useful even for WSDM.

As we've seen in this text, UML has positive as negative aspects when used as a modeling notation for hypermedia applications.

The main negative aspect comes from a data modeling perspective; ORM shows to be a better notation (see 1.4) and allows building a better model then when using UML class diagrams. Using ORM as a front-end to UML could solve this, this way modelers can exploit the benefits of both approaches [33]. Also, future revisions of UML could solve several problems that currently exist.

Due to the previous mentioned aspect, "pure" UML is not able to create a model of the same quality as ORM. Extending UML – which was done with ORM also when defining WSDM, solves this. These extensions or profiles give UML a lot of advantages over ORM when it comes to modeling specific domains.

Other aspects that might prove useful when considering UML as a modeling notation for WSDM are:

- Wider acceptance within the developer community
- UML profiles which extend the UML notations and enable the designer to create a better model – this model can include every aspect of the design method or parts like the presentation design or the (logical) database design phase
- When seen as a software project, UML easily allows modeling of design patterns and/or object-oriented code
- Support within case tools is growing everyday, those case tools come with a lot of features such as semi-automatic code generation
- Parallel research (based on UML) might help increase the speed of the development of WSDM as a design method (e.g. on could propose to start using the UWE profile)

Future work will surely involve an UML profile that fits the needs of WSDM. Also it might be interesting to redefine additional research in terms of UML notation as presented in 4.1 for example. By extending UML and/or using UML as a modeling notation for WSDM, one can think of an (semi-) automatic code generation tool as in [28].

Bibliography

[1] J. Conallen (2002) Building Web Applications with UML Second Edition – Addison Wesley ISBN 0-201-73038-3

[2] WAE-UML home page - http://www.wae-uml.org

[3] N. Koch (1999) A Comparative Study of Methods for Hypermedia Development - Technical Report 9905, Ludwig-Maximilians-Universität München

[4] Garzotto F., Paolini P and Schwabe D. (1993) HDM --- a model-based approach to hypertext application design. ACM Transactions on Information Systems, 11(1): 1--26, Jan. 1993 - http://www.inf.udec.cl/~yfarran/HDM.htm

[5] Isakowitz T., Stohr E. and Balasubramanian P. (1995). A methodology for the design of structured hypermedia applications. Communications of the ACM, 8(38), 34-44.

[6] Daniel Schwabe and Gustavo Rossi, "An Object Oriented Approach to Web-Based Application Design", Theory and Practice of Object Systems 4(4), 1998. Wiley and Sons, New York, ISSN 1074-3224)

[7] D. Schwabe and G. Rossi (2001), “The Object -Oriented Hypermedia Design model (OOHDM) ” - http://www.telemidia.pucrio.br/oohdm/oohdm.html

[8] Fernanda Lima and Daniel Schwabe (2003) - "Modeling Applications for the Semantic Web", Extended version of paper in ICWE 2003 – Second International Conference on Web Engineering, Oviedo, Spain, 2003 - Lecture Notes in Computer Science.

[9] S. Ceri, P. Fraternali and A. Bongio (2000), “Web Modeling language (WebML): a modeling language for designing Web sites”, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy.

[10] S. Ceri, P. Fraternali, M. Matera, A. Maurino (2001) "Designing Multi-Role, Collaborative Web Sites with WebML: a Conference Management System Case Study". IWWOST'01, Valencia, Spain, June 2001

[11] J. Gomez, C. Cachero and O. Pastor (2001), “Conceptual Modeling of Device- Independent Web Applications”, http://www.dsic.upv.es/~west2001/iwwost01/files/contributions/JaimeGomez/ieeeMultimedia.pdf.

[12] J. Gomez, C. Cachero and O. Pastor (2001), “Extending a Conceptual Modeling Approach to Web Application Design”, http://www.dsic.upv.es/~west2001/iwwost01/files/contributions/JaimeGomez/caise00.pdf

[13] Koch N., Kraus A. and Hennicker R. (2001) The Authoring Process of the UML-based Web Engineering Approach, In First International Workshop on Web-Oriented Software Technology IWWOST'2001, Valencia, to appear.

[14] Koch N. (2001) Software Engineering for Adaptive Hypermedia Applications, PhD. Thesis, Reihe Softwaretechnik 12, Uni-Druck Publishing Company, Munich. http://www.pst.informatik.uni-muenchen.de/personen/kochn/PhDThesisNoraKoch.pdf

[15] Koch N. and Kraus A. (2002). The expressive Power of UML-based Web Engineering. Proc. of IWWOST´02, CYTED, pp. 105-119.
[16] UML Version 1.5 (2003) Unified Modeling Language. The Object Management Group (OMG) - http://www.omg.org
[17] Booch G., Rumbauch J. and Jacobson I. (1999) The Unified Modeling Language User Guide, Addison Wesley
[18] Baumeister, H., Koch, N., and Mandel L.: Towards a UML extension for hypermedia design. In proceedings «UML»'99, France, R., Rumpe, B. (eds), LNCS, Vol. 1723. Springer-Verlag (1999) 614-629.
[19] Terry Halpin (1998) Object-Role Modeling (ORM/NIAM) -appeared as chapter 4 of Handbook on Architectures of Information Systems, P. Bernus, K. Mertins & G. Schmidt, Springer-Verlag, Berlin.
[20] ORM home page - http://www.orm.net/
[21] Fowler M., Kendall S. (1997) UML Distilled, Applying the standard Object Modeling Language – ISBN 0-201-32563-2
[22] Sinan Si Alhir (1998) What is UML? - http://home.earthlink.net/~salhir/WhatIsTheUML.PDF
[23] Casteleyn, S., De Troyer, O., Brockmans, S.: "Design Time Support for Adaptive Behavior in Web Sites", In Proceedings of the 18th ACM Symposium on Applied Computing, Melbourne, USA (2003)
[24] Brockmans S. (2002) Thesis - Designing Adaptive Web Sites using WSDM – Free University Brussels
[25] Papasalouros A. et al - An Integrated Model for the authoring of Web-based Adaptive Educational Applications - http://wwwis.win.tue.nl/ah2003/proceedings/paper3.pdf
[26] Rational Rose homepage - http://www.rational.com/products/rose/index.jsp
[27] Poseidon for UML homepage - http://www.gentleware.com/products/index.php3
[28] ArgoUWE homepage - http://www.pst.informatik.uni-muenchen.de/projekte/argouwe/
[29] ArgoUML homepage - http://argouml.tigris.org/
[30] UML to PHP homepage - http://www.phpedit.net/products/xmi2php/
[31] UML XMI homepage - http://www.omg.org/technology/documents/formal/xmi.htm
[32] Dr. Halpin T. (1998-99) UML data models from an ORM perspective Part 1 to 10 - http://www.orm.net/uml_orm.html
[33] Dr. Halpin T. (2001) Integrating Fact-oriented Modeling with Object-oriented Modeling – Chapter IX, Information Modeling in the New Millennium by Rossi M. et al. – Idea Group Publishing
[34] Dr. Halpin T. (2001) Augmenting UML with Fact-orientation – In workshop proceedings: UML: a critical evaluation and suggested future, HICCS-34 conference
[35] Dr Halpin T. (2002) Evolving UML: Opportunities and Challenges – http://www.orm.net/IRMA2002.html
[36] De Troyer, O., Casteleyn, S.: "The Conference Review System with WSDM", In IWWOST 2001, Valencia, Spain (2001) http://www.dsic.upv.es/~west2001/iwwost01
[37] De Troyer, O., Leune, C.: "WSDM: A User-Centered Design Method for Web Sites", In Computer Networks and ISDN systems,

Proceedings of the 7th International World Wide Web Conference, Elsevier, pp. 85 - 94 (1998)
[38] O. De Troyer (2001), “WiSDoM Short descript ion”, http://wsdm.vub.ac.be/ShortDescription.htm, Brussels, Belgium.
[39] Schwabe D. (2001), A Conference Review System - http://www.dsic.upv.es/~west2001/
[40] Scott W. Ambler (2002) A UML Profile for Data Modeling - http://www.agiledata.org/essays/umlDataModelingProfile.html
[41] Davor Gornik (2002) UML Data Modeling Profile - http://www.rational.com/media/whitepapers/tp162.pdf
[42] Struts framework - http://jakarta.apache.org/struts/