Updated version published in: Information modelling in the new millennium, Eds. Matt Rossi & Keng Siau, IDEA GroupPublishing, ISBN 1-878289-77-2, 2001.

Audience-driven Web Design

Olga De Troyer

Vrije Universiteit Brussel
WISE Research group
Pleinlaan 2, B-1050 Brussel, Belgium
Olga.DeTroyer@vub.ac.be

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" was a fact with its legendary stories of delays, unreliability, maintenance bottlenecks and costs.

Now we seem to be starting to deal painfully with a corresponding "web site crisis". Over the last few years, the Internet has boomed and the World Wide Web with it. Web browsers are the basic user platform of the Internet. Because of the immense potential audience, and because publishing on the web is in principle very easy, the number of web applications has exploded. Most of the web sites are created opportunistically without prior planning or analysis. Moreover, even large mission-critical intranet projects are being started without any regard for methodology. The resulting problems of maintenance and development backlog, so well known in "classical" information systems, can easily be predicted, and will happen on a much larger scale. Because web sites are almost by definition required to adapt and grow, and have to interact with other sites and systems unknown at the moment of creation, these problems will also be much more complex and severe.

In addition to the predictable maintenance and development problems, a new problem unknown in classical information systems has emerged: competition for the user's attention. Especially for commercial web sites it is important to hold the interest of the user and to keep them coming back to the site. If for some reason a visitor is not satisfied with the site or he cannot find (fast enough) the information he is looking for, there is a high chance that he will leave the site and not return anymore. Much more than in "classical" software systems, the usability of web applications are primordial for their success.

A new discipline of *web engineering* is required, to provide structured approaches for design, development, interoperation and maintenance of web applications. A number of researchers have already recognised the lack of design methods for web sites, or more in general for web-based information systems. They have proposed methods:

- HDM (Garzotto, Paolini and Schwabe, 1993) and its successors Garzotto, Paolini and Mainetti, 1993) and OOHDM (Schwabe and Rossi, 1995), (Schwabe et al., 1996; Schwabe and Rossi, 1998);
- RMM (Isakowitz et al, 1995);
- W3DT (Bichler and Nusser, 1996);
- the method for analysis and design of web sites in (Takahashi and Liang, 1997);
- SOHDM (Lee et al, 1998).

Older methods (HDM, RMM) were originally designed for hypertext or hypermedia applications and do not deal comfortably with web-specific issues. In addition, these methods are very much data-driven or implementation oriented. Some have their origin in database design methods like the E-R method (Chen, 1976) or object oriented (OO) methods such as OMT (Rumbaugh et al., 1991). These methods may be able to solve maintenance problems to some extent but they do not address the typical usability problems mentioned above.

WSDM, or Web Site Design Method ("WiSDoM") is a new approach (De Troyer and Leune, 1998; De Troyer, 1998; Goedefroy et al., 1998) for designing web sites. Rather than taking an organisation's data or database as starting point and wondering how all of that should be displayed on the Internet (the so-called *data-driven* approach), WSDM takes as starting point the needs and requirements of the intended audience(s) of the web site. This approach we call *audience-driven*[1]. WSDM gives consideration to the fact that the target audiences of a web site may be composed of different “kinds” of visitors/users. Different kind of visitors may have different requirements (information requirements as well as functional- and usability requirements). Therefore the target audiences are classified into *audience classes.* Each audience class has its own requirements and characteristics that may need to be reflected in the content of the site (not all information is relevant for all users) as well as in the “interface” of the site (like the language and jargon used, the look-and-feel, the existing browser capabilities, etc.). Each audience class will be targeted in an appropriate way by the site. This approach leads to web sites that are better tailored to these visitors' needs and therefore may have a higher usability and greater satisfaction.

We also make a clear distinction between the conceptual design (which is free from any implementation detail) and the design of the actual presentation which takes into consideration the implementation language used, the grouping in pages, the use of menus, static and dynamic links, etc. This distinction is similar to the distinction made in database design between the conceptual schema (e.g. an E-R schema (Chen, 1976) or Object-Role schema (Halpin, 1995)) and the logical schema (e.g. a relational schema (Date, 1990)). This distinction has proven its usefulness for more than 15 years. This distinction also allow us to propose a method for web site design which is not biased by the diversity and rapid growing obsolescence of the web technology as well as by current implementation limitations.

Another important advantage is that the conceptual schema of the web site, made during conceptual design, can be made available to the web site audience. This may serve a double purpose:

- To prevent that users search for information which is not available in the site.
- To reduce the lost-in hyperspace syndrome (Maurer, 1996).

If given in a standard format (e.g. using XML), the many different search engines may exploit the information in this conceptual schema to enhance their search capabilities. The issue of providing the conceptual schema of a web site inside the web site is outside the scope of this chapter and is described elsewhere, e.g. (Decruyenare, 1999; De Troyer and Decruyenare, 2000).

The rest of this chapter is structured as follows: In section 2 we describe some of the problems that may arise in web sites. In section 3 we give a general overview of the different phases of WSDM. Sections 4, 5 and 6 will describe the most important and innovating phases of the method into more detail: section 4 deals with the mission statement of a web site, section 5 describes the audience modelling and section 6 the conceptual modelling. Section 7 draws conclusions.

DESIGNING WEB SITES

There are different kinds of web sites. Here we will concentrate on what we can call "static" web sites. Although the term static may be a little bit misleading we use it to indicate the difference with the so-called "dynamic" or "active" web sites (British HCI Group, 1999). Although a static web site may contain dynamic elements (like a video or animation) and the pages may be generated on the fly, the view that a user has of a page is rather static. The actual content will not change while he is looking to the page and if he navigates and comes back to the page it will still be the same. In an active or adaptive web site this is not necessarily the case, e.g. navigation may influence the content of a page on the fly.

There is also the distinction between kiosk type and application type (Isakowitz et al., 1995) web sites. A kiosk web site mainly provides information and allows users to navigate through that information, i.e. an information repository. An application web site is a kind of interactive information system where the user interface is formed by a set of web pages. Here we concentrate on the kiosk type of web sites. The design issue of this type of web sites is not only to make them attractive and fashionable, but above all to make maintainable, reliable and usable sites. Therefore, the main issue is the structuring of the (complex) information domain and the presentation to the users in a clear and easy accessible way.

Indeed as any database designer knows, if the represented information is not structured properly, maintenance problems occur which are very similar to those in databases: redundancy, inconsistency, incompleteness and obsolescence. This is not surprising as web sites as well as databases may provide (large) amounts of information that need to be

maintained. But the same phenomena also lead to **usability problems** experienced by the audience of the web site:

- **Redundancy**. Information that is needlessly repeated during navigation is annoying to most users.
- **Inconsistency**. If some information on the site is found to be inconsistent the user will probably distrust the whole site.
- **Incompleteness**. Stale or broken link falls in this category, but users who cannot find the information that they expect to be available on a site also experience a kind of incompleteness.
- **Obsolescence**. Organisations and information are often changing so quickly that the information provided on web sites soon becomes out of date. If a web site has visibly not been updated for a while, confidence of the users in the information provided will likely not be very high.

Other issues that may cause usability problems are:

- **Lack of a mission statement**. If a web site has no declared goal, that goal can quite simply not be reached. The key question therefore that must be answered first is "What do I want to get out of my site?". If there is no such statement, there will be no basis for any evaluation of the effectiveness of the site.
- **Lack of clearly identified target audience**. Every web site has an audience. But in order to create a compelling and effective site, the audience must be target (predetermined) in advance. The target audience is that specific audience who is interested in and will benefit from the content of the web site. Determining who the target audience is will help to communicate to them more effectively.
- **Information overload**. Users typically are not interested in wading through pages and pages of spurious "information". Also, attention spans tend to be short.
- **Visual overload**. To many banners, logo's, graphics, colours and animations will confuse users and distract them from the real important information.
- **The lost-in-hyperspace syndrome** (Maurer, 1996). Hypertext requires users to navigate through the provided information. If the information or this navigation process is not well structured or guided, users may easily get lost. This makes it more difficult and time consuming to locate the desired information.

WSDM OVERVIEW

WSDM is *audience-driven*, not data driven. This means that instead of letting the structure of the available data set drive the design of the web site, as in most methods, we will create a web site based on the requirements of the intended audience(s). To do so, WSDM gives consideration to the fact that web sites usually have different types of visitors that may have different needs.

Consider as an example the web site of a university. Typical users of such a web site are candidate students, enrolled students, researchers and university staff.

- Candidate students are looking for general information about the university and the content of the different programs of study.

- The enrolled students need detail information about the different courses, timetables and contact information of the lecturers (telephone extension, room number, and contact-hours).
- Researchers look for information on research projects, publications and general information on the researchers (full address, research interests, and research activities).

This shows that different types of users (we will use the term audience class) may have different information requirements. This should be reflected in the web site. For example, a student should be able to follow a navigation path that leads him to the information he is interested in without having to travel through pages of other (for him) non-relevant information. Usually, this is not the case, e.g. most lecturers have a "home page" that contains all information they could think of: research projects, research interests, publications, courses, university functions, etc. If a student needs to know the contact hours of a lecturer, he has to go to this home page and scan the page(s) in order to find the contact hours (if the lecturer did not forgot them after all). This is a result of a data-driven approach. Although it may be easy to maintain such a "home page" (all information stands together), this is not efficient from a usability point of view. In addition, providing too much non-relevant information will enhance the lost-in-hyperspace syndrome.

Next to the fact that different types of users may have different information requirements, it may be necessary to represent the same information in different ways to different kinds of users. This depends on the *characteristics* of the users. As an example we again consider the university site.

- Candidate students, especially secondary school students are not familiar with the university jargon. Also, by preference, the information should be offered in the native language.
- The enrolled students are familiar with the university jargon. They also prefer to have the information in the native language. However, foreign students who follow the exchange programs also require the same information. Therefore, in addition, it may be necessary to give the same information in English for the exchange students.

Note that, what we call, an audience-driven approach is not the same as the user-centered approach used in the Human Computer Interaction (HCI) community. In a user-centered approach the requirements of the users (acquired by e.g. interviews and scenario analysis) largely guide the design and users are actually involved in the development process. This is not possible for web sites on the WWW because most of the users are unknown, they cannot be interviewed in advance and they cannot be involved in the development process. However, we can fairly well identify the different types of target users and investigate their requirements.

In figure 1 an overview of the WSDM method is given. The first step is to define the *mission statement*. The mission statement should express the purpose and the subject of the web site and declare the target audience. Based on this mission statement a two-step *audience modelling* phase is performed. In the first step, *audience classification*, the

different kinds of users are identified and classified (i.e. their audience class is formally identified). Members of the same audience class have the same information and functional requirements. In the next step, called *audience class characterisation*, the characteristics of the different audience classes are given. The result of the audience modelling is a set of audience classes together with an informal description of their requirements, *information*- and *functional* - as well as the *usability* requirements, and their characteristics. If within one audience class, members have different characteristics *variants* of audience classes are introduced that group members of the same class with the same characteristics.

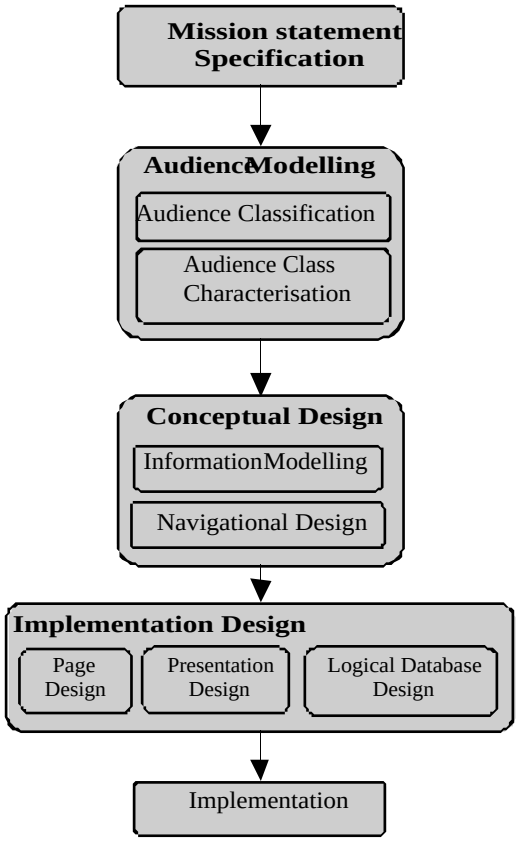

Figure 1: Overview of WSDM

Next we perform a *conceptual design* of the site. The separation of the conceptual design from the actual implementation design, which simply is good practice carried over from classical Information System design, allows designing a site without bias by current implementation technology. The conceptual design phase is divided in two steps: *information modelling* and *navigation design*. During information modelling, *audience object models* are created. These models model the information requirements of the different audience classes. The audience object models are composed in a step-like way by translating information requirements into so-called *object chunks*. *Variant object models* capture the differences between the variants of an audience class. The different audience object models are linked together by a single object model, called the *Business Object Model*. The Business Object Model describes the information available in the organisation, independently of any particular use. All audience object models will be

defined as views on this model. In this way, possible redundancy is described and therefore can be controlled.

During navigation design we describe the (conceptual) structure of the web site and model how the members from the different audience classes will be able to navigate through the site. For each audience class (variant) a *navigation track* is created. These navigation tracks may also be composed in a step-like way again by translating requirements into so-called *navigation chunks*. All navigation tracks together form the *navigation model* of the site.

During *implementation design* we essentially design the (page) structure as well as the 'look and feel' of the web site. The aim is to create a consistent, pleasing and efficient look and feel for the conceptual design made in the previous phase by taking into consideration the usability requirements and characteristics of the audience classes. The page structure is derived from the navigational model. The amount of information on a page should not overwhelm the user. In addition it should also not cause a long download time. On the other hand, too little information on a page may force the user to "click" too much. If the information provided by the web site (or parts of it) will be maintained by means of a database then the implementation design phase will also include the logical design of this database. This logical database schema can be derived from the business object model.

The last phase, *implementation*, is the actual realisation of the web site using the chosen implementation environment. E.g. for an HTML implementation this means that the implementation model must be converted into a set of files containing HTML source code. This step can be largely automated using available tools and environments for assisting in HTML implementations.

THE MISSION STATEMENT

The first step in WSDM is to define the *mission statement* of the web site. The mission statement must answer the following questions: *what is the **purpose** of the web site, what is the **subject*** and *who are the **target audience(s)***.

We believe that without giving due consideration to the purpose of the web site, there is no proper basis for making decisions, or for evaluating the effectiveness of the web site and the visitors will have little idea what the web site is for.

The target audiences are the audiences that we want to address or that will be interested in the site. To be able to address them in an appropriated way in the web site, one need to have a clear understanding of one's target audiences. Therefore we first have to agree on the target audiences of the web site.

Related to the purpose and the target audiences is the subject of the web site. The subject of the web site must allow fulfilling the purpose of the web site and it must be suitable for the target audiences. It is important to identify the subject because this will allow later on

to decide which information must be included in the web site or which not. Not all information is good. After all, no matter how much information is put on the web site, if nobody is interested in, nobody will ever look through it.

As an example we give the mission statement for a web site of a typical university department. This mission statement is twofold:

- Enhance the communication between students and lecturers by providing detail information about the available courses.
- Provide information about the available programs to potential students to attract more students.

The target audiences in this example are students, lecturers and potential students. The purpose is to enhance the communication between students and lecturers and to attract more new students. The subject is courses and programs. Note that by this mission statement, the subject of this web site is deliberately limited to teaching information and researchers are not part of the target audiences.

AUDIENCE MODELLING

In general, the mission statement only gives a general indication of the target audience of the site. E.g. for a commercial site, the target audience may be customers and potential customers. However this specification is too broad to decide whether this target audience should be modelled as one audience class or as two or more audience classes.

Identifying Audience Classes

To identify the different audience classes we look at the activities of the organisation for which the web site is built. These activities are decomposed in order to refine in each decomposition step the target audience given in the mission statement.

We only consider the activities that are related to the purpose and subject of the web site. In our university department example this is only the activity “Provide Education”. Activities such as “Perform Research” and “Advise Companies” are not (directly) relevant for the purpose of this web site.

Each activity involves people. These people are potential users if they belong to the target audience given in the mission statement. In our example, the activity “Provide Education” involves lecturers, enrolled students, potential students and staff members (see figure 2).

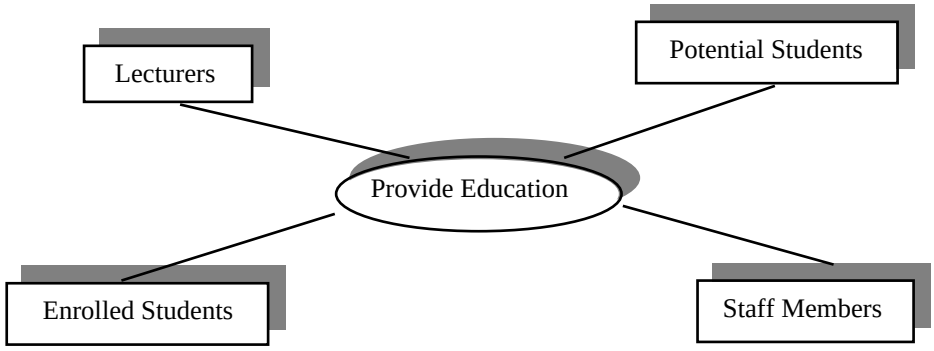

Figure 2: Activity Diagram “Provide Education”.

Staff members do not belong to our initial target audience. Given the purpose of the web site, there is also no reason to include them in the target audience; therefore we do not consider them further on in the design of the web site. This leaves us with the lecturers, enrolled students and potential students. To decide whether these can be in one audience class or we need several audience classes we look to their requirements. By definition we state that all members of an audience class have the same information and functional requirements. For our example we can formulate the following requirements:

- Enrolled students: require detail information about courses.
- Lecturers: need to be able to distribute detail information about their courses and like to know who is following their courses.
- Potential students: require general information on study programs and courses.

The requirements of these different groups are sufficiently different to put them in different audience classes. This results in the initial audience classes: **Enrolled Students**, **Lecturers** and **Potential Students**.

In order to check whether we have to refine these audience classes, we check if we can decompose the activity “Provide Education” into sub-activities. For our university department we can decompose this activity into “Provide Graduate Education” and “Provide Master Education”. This result in a new activity diagram (see figure 3).

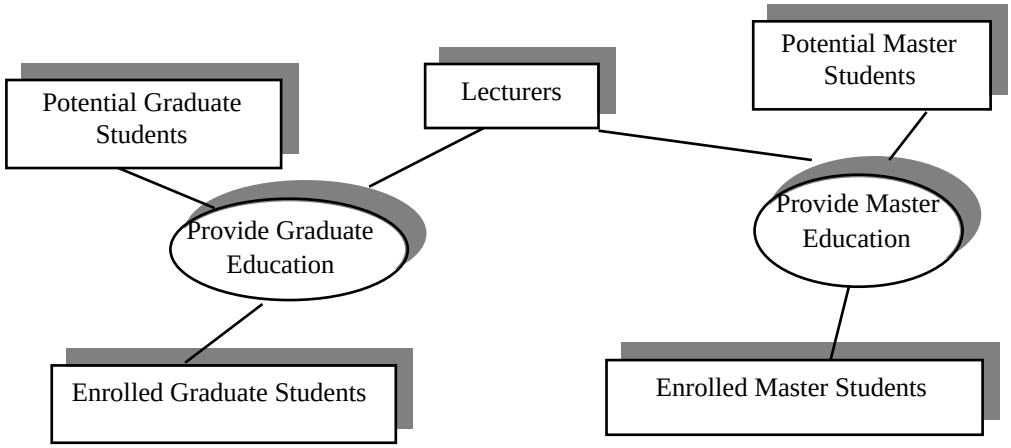

Figure 3: Decomposition of Activity Diagram “Provide Education”.

However, enrolled graduate students and enrolled master students do not have distinct information or functional requirements. Therefore there is no need to put them in different audience classes. The same applies to potential graduate students and potential master students. This decomposition of the activity “Provide Education” does not result into any new audience classes, and therefore this decomposition is stopped. In general, we continue the decomposition of an activity until no more new audience classes are found or until no decomposition is useful anymore.

The method can be summarised as follows:

1. Consider the activities of the organisation which are related to the purpose of the mission statement;
2. For each activity:
   2.1. Identify the people involved in the activity;
   2.2. Only consider those people which belong to the target audience formulated in the mission statement;
   2.3. Divide these people into audience classes based on different information or functional requirements;
   2.4. Decompose the activity if possible;
   2.5. Repeat step 2 until no new audience classes are found or no decomposition is possible.

This method is based on the fact that we are able to identify (at a high level) the requirements of people involved in an activity. For projects where we exactly know the people involved in the activities and where we can involve (a selection of) the people (the future users) in the development process, we can use the standard techniques of software engineering like questionnaires, interviews, etc. to collect the requirements. Most of the intranet projects belong in this group of projects. In the example, the part of the web site to enhance the communication between students and lecturers is an example of such a project. In most Internet projects we are usually unable to involve the target audience itself in the development process. In the example, this is for instance the case for the potential students. Therefore it looks as if we has to "guess" for their requirements. Studying the characteristics of the audience (see next section) may help to formulate their requirements. Usually, once the system is implemented and running feedback from the users will be needed to adjust and enhance the design.

Characteristics of Audience Classes

As already explained, all members of an audience class potentially have the same information and functional requirements. However the members of one audience class may diverge on how the information should be presented to them, i.e. they may have different *usability requirements*. E.g. younger people are more visually oriented than older people are. Therefore we also analyse the characteristics of the audience classes.

Some examples of user's characteristics are: level of experience with web sites in general, frequency of use, language issues, education/intellectual abilities, age, income, lifestyle… Some of the characteristics may be translated into usability requirements while others may be used later on in the implementation phase to guide the design of the "look and feel" of the web site, e.g. choice of colours, fonts, graphics, etc.

If within one audience class we can distinguish groups of members with different characteristics, we introduce *audience class variants*. Consider as an example in the university site example the audience class **Enrolled Student**s. In this class we may distinguish between local students and exchange students. Both have the same

information requirements (detail information on courses) but they have different characteristics and usability requirements:

- Local students are young (between 18 and 28), are familiar with the university jargon, the university rules and customs. They have a good level of experience with the WWW. They prefer the local language for communication, but have in general a good understanding of English.
- All communication with the exchange students is done in English. We may not presume that they are familiar with the university jargon and customs, or with the WWW.

Therefore, for the audience class **Enrolled Students** we will distinguish two variants: **Local Students** and **Exchange Students**.

CONCEPTUAL DESIGN

During the first phase, audience modelling, the information-, functional- and usability requirements and the characteristics of the potential visitors are identified and different audience classes and variants are recognised. The goal of the conceptual design is to turn these requirements into a high level, formal description which can be used later on to generate (automatically or semi -automatically) effective web sites.

During conceptual design, we concentrate on the **conceptual** "what and how" rather than on the visual "what and how". This means that like in database design we describe what kind of information will be presented (objects, object types and relationships) (the conceptual "what"), but unlike in database design we also describe how the information will be structured and how it will be possible to navigate through the information (the conceptual "how"). This is needed because navigating through the information space is an essential characteristic of web sites. If the navigation is not (well) designed or not adapted to the target audience, serious usability problems occur. The conceptual "what" is mainly covered by the information modelling step, the conceptual "how" by the navigation design.

Information Modelling

The information modelling step is intended for the so-called data intensive web sites. These are web sites dealing with a lot of structured data. E.g. an airline company web site offering the time schedule and prices of its flights is a data intensive web site. On the other hand a web site that only provides a company's profile and contact information is not a data intensive web site and does not need an information modelling step. Most web sites, like in our example, are mixed. Parts are data intensive, others not.

The purpose of the information modelling step is to model the structured data offered by the web site. The reason to do this is the same as in database design. It will offer more structure and will enhance the maintenance; e.g. the data can be maintained in a database.

Information Modelling in an Audience-driven Approach

In classical database design, information modelling results in a single conceptual schema. In WSDM, the information modelling results in several different conceptual schemes. This is because we have opted for the audience-driven approach.

In a data-driven approach, as used in database design, the starting point of a conceptual design is the information available in the organisation: designers first model the application domain, and subsequently they associate information to each class of users (e.g. by means of views or external schemes). However, the data and the way it is organised in the application domain may not reflect the user's needs. A good example of such a mismatch could be found in the first web site of my previous university. The structure of this web site completely reflected the internal organisation structure of this university. This structure was completely irrelevant for, and unknown to most users of that site. As an example, to look for the products offered by the Computer-shop (called PC-Shop) you had to know that the PC-shop was a part of the Computer Center (actually it was listed as one of the "External Services" of the Computer Center), which itself was listed as a "Service Department" of the University. You could not find the PC-shop as a "Facility" like the Restaurant, the Copyshop or the Branch Bank.

In our audience-driven approach we start by modelling the information requirements of the different audience classes. For each audience class a conceptual schema is developed expressing the information needs of the members of that class. We call these conceptual schemes *Audience Object Models*. Like an "ordinary" conceptual schema, an Audience Object Model (AOM) is expressed in terms of the business objects of the organisation.

In conceptual modelling in general, object models describe the different object types (OTs for short), the relationships between these OTs, and rules or constraints. OO models also describe behaviour. For our purpose (modelling kiosk web sites), modelling behaviour is not (yet) needed. The traditional conceptual modelling methods like E-R (Chen, 1976), the Object-Role Model (ORM) (Halpin, 1995; Wintraecken, 1990; De Troyer, 1996), or "true" OO methods like OMT (Rumbaugh et al., 1991) are therefore suitable. Here, we have used ORM. ORM views the world as objects playing roles. The basic building blocks of ORM are Object Types (OT) (graphically represented as circles) and binary relationships composed of two roles (graphically represented as a rectangle composed into two boxes, each box connects with a line to the corresponding OT). Identifiers (represented by lines on roles) are used to indicate one-to-one, one-to-many or many-to-many relationships. A mandatory role is indicated by a dot. Object Types are divided into Lexical Object Types (graphically represented by a dashed circle) and Non Lexical Object Types (graphically represented by a solid circle). Instances of Lexical Object Types are utter-able, instances of Non Lexical Object Types not. E.g. "Course" is a Non Lexical Object Type; "CourseName" is a Lexical Object Type. If no explicit distinction between Lexical and Non Lexical is needed, the Object Type is represented by the combination of a solid and a dashed circle. See e.g. the Object Type "Telephone" in figure 6. Figure 6 shows a part of the AOM developed for the audience class **Enrolled Students** of our university department example. ORM is also explained into more depth

in another chapter of this book: "Integrating fact-oriented modeling with object-oriented modeling" by Terry Halpin.

Modelling by means of Chunks

In this section we explain how the audience object models can be constructed from the informal requirements expressed for the audience classes. The audience object models are composed in a step-like way by translating information requirements into so-called *object chunks.* As we will explain later, this allows us to support the typical incremental and evolving nature of the design process of web sites.

To build the audience object model for a particular audience class, we perform the following steps:

1. Only consider the information requirements (usually in the form of structured natural language text);
2. Elaborate the informal information requirements;
3. Divide the information requirements into so-called *elementary information requirements*;
4. For each elementary information requirement:
   Make an object model, called *object chunk* (using an information modelling technique e.g. (Chen, 1976; Halpin, 1995; Rumbaugh et al., 1991));
5. Compose the object chunks of one audience class into a single object model, a so-called *audience object model*.

As an example we again consider our university department web site. For the audience class Enrolled Students, we consider the requirement:
"*Provide detailed information on courses and on their lecturers*".

Information analysis of this requirement may result into the following more elaborated information requirement:
"*For each course the following information is needed:*
Name, code, credit points, study period in which the course is given, other courses which are a prerequisite for the course, description of the content, type of exam, literature used, and for each lecturer involved in teaching the course: the name, telephone, room number, e-mail address and contact hours."

Next, we divide this requirement into two elementary requirements:
1. *"For each course the following information is needed: name, code, credit points, study period in which the course is given, other courses which are a prerequisite for the course, description of the content, type of exam, literature used, and lecturers involved in teaching the course.*"

2. *"For each lecturer involved in teaching a course: the name, telephone, room number, e-mail address and contact hours.*"

We define an *elementary information requirement* as an information requirement that only deals with one object type. For example, requirement 1 is about "Course", requirement 2 about "Lecturer".
In the next step, for each elementary information requirement a (specific) object model describing the information needed to satisfy this requirement is built. We call these models *object chunks* because they can be seen as parts of a larger object model (i.e. the audience object model). Figure 4 and 5 give the object chunks for respectively sub-requirements 1 and 2 of the example. We have used ORM.

Finally, the different object chunks of one audience class are combined into one model, *the audience object model*. This can be done in an automatic or semi-automatic way. Schema integration techniques may be applied to eliminate redundancy and resolve inconsistencies (which may for instance arise when different people model different chunks). However, it is important to keep track of the different object chunks in the total audience object schema. This is necessary to support the need for the continual change of web sites. By using the concept of elementary requirements, changes to information requirements can be localised to one or a number of elementary requirements. These elementary requirements correspond one-to-one to object chunks. Tracing these object chunks throughout the design process makes it possible to compute the impact of a change in the final web site (e.g. by means of a tool). Figure 6 gives a (part of) the audience object model for the audience class **Enrolled Students**. The object chunks are marked by means of dotted lines.

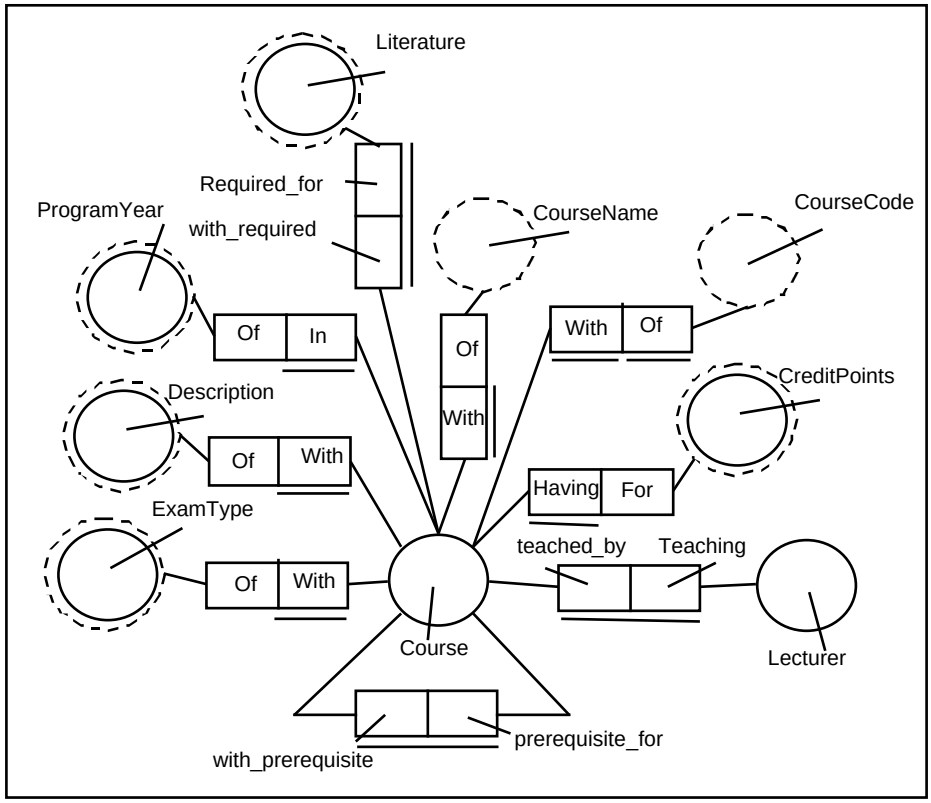

Figure 4: Object chunk for requirement 1.

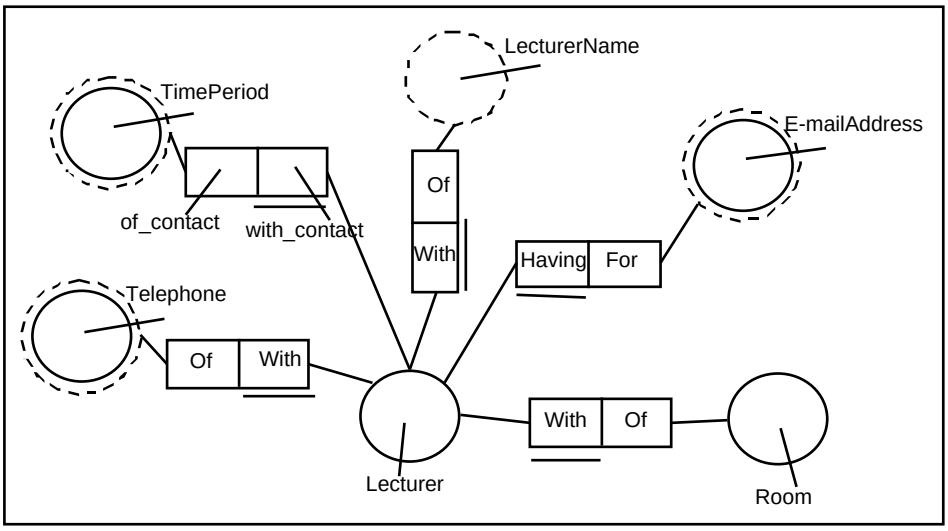

Figure 5: Object chunk for requirement 2.

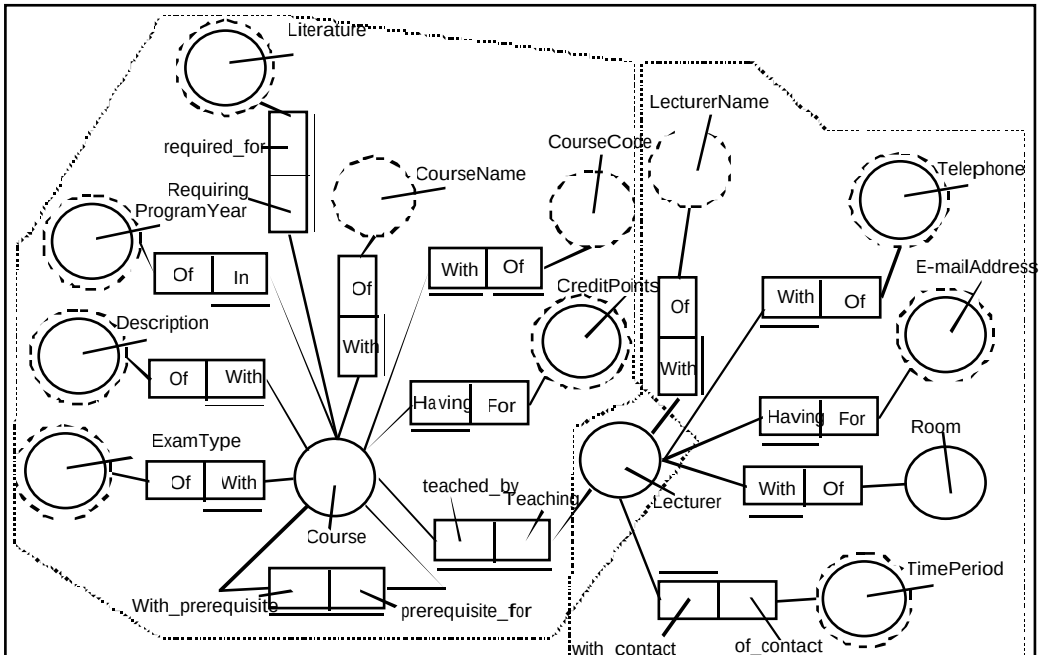

Figure 6: Audience Model (partially).

Object Type Variants

As explained earlier, an audience class may have different variants expressing different usability requirements and characteristics. It is possible that this also results in slightly different information requirements. E.g. for the audience class **Enrolled Students**, the members of the **Exchange Students** variant require the same type of information on courses as the members of the **Local Students** variant but the information should be in

English. In addition, exchange students do not follow a program like the local students, only a number of courses. In WSDM, we model this by means of *object type variants* (OT variants*)*.

An OT variant is not a subtype. An OT variant largely corresponds with the original OT but has some small differences (*variations*). Consider as an example the OT **Course** for the audience class **Enrolled Students**. About a course, enrolled students need the following information:

- the code of the course;
- the name of the course;
- a description of the content of the course;
- the prerequisites for the course;
- specification of the required literature;
- the type of exam of the course;
- the number of study points;
- the program year in which the course can be followed.

For the **Local Students** variant we will offer this information in the local language, while for the **Exchange Students** variant the information must be provided in English. Also the program year is not relevant for the **Exchange Students** variant and (the actual value of) the required literature and the exam type may differ between exchange students and local students. Indeed, local students may have required literature written in the local language while for the exchange students the required literature must be written in English. In implementation terms, this means that for most (but not all) attributes of the OT **Course** we will need to maintain two *variants*; one for the **Exchange Students** variant (in English) and one for the **Local Students** (in the local language). The recognition of these differences is essential for an audience-driven approach and therefore they should be modelled in an early phase. Some people may argue that the language is a representation issue and therefore it should not be considered in the conceptual phase but left to the implementation design. However, in this example, the language issue is an important usability requirement that also influences the actual information that will be provided. If we do not recognised this during conceptual design, all information provided for a course, except for the language in which it is given, would be the same for local students and exchange students.

To model the difference we introduce two variants for the OT **Course**: the **Course/LocalStudents** variant and the **Course/ExchangeStudents** variant. A graphical representation of the **Course/ExchangeStudents** variant is given in figure 7. The name of an OT variant is composed of the name of the original OT (also called parent OT) followed by the variant identification, e.g. **Course/ExchangeStudents.** Only the relationships that differ from the original ones are given.
CourseName/ExchangeStudents is the **Exchange Students** variant of lexical OT **CourseName**.

An OT variant can have fewer relationships than its parent OT can. Semantically, this means that those relationships are not meaningful for the variant. E.g. the relationship with ProgramYear is struck out for **Course/ExchangeStudents** because it is not

meaningful for exchange students. An OT variant cannot have relationships that are not defined for the parent OT. This is to prohibit that completely new information (relationships) is added to a variant, in which case it is not a variant anymore.

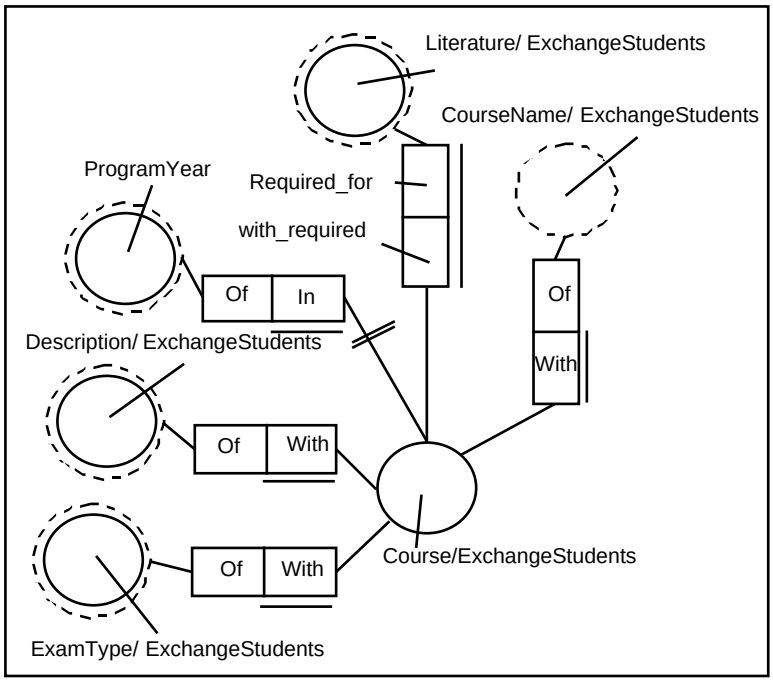

Figure 7: Course/ExchangeStudents variant.

For each OT in an audience object model of an audience class, and for each variant of this audience class, an OT variant may be defined to reflect the possible information differences. Replacing the OTs by the corresponding OT variants in the original audience object model gives the specific object model for the variant of an audience class called a *Variant Object Model.*

Linking the Object Models

Different audience classes may need the same information. For example, enrolled students as well as potential students need general information about courses, such as the name, a short description and the program year. This means that different audience object models may model the same information. This redundancy is deliberately (the same information may need different layouts or different navigation paths for different audience classes) and will not cause any problems as long as we recognise that it is indeed the same information and that it can be maintained in a single place. Therefore, we have to relate the information in the different audience object models to each other. To do this we use an overall object model or domain model, called the *Business Object Model* (BOM). This model describes the information needed by the different audience classes is an independent way. In fact it is a conceptual description of the information (business objects) available in the organisation, and may already be developed for the organisation. If not or it is not in a shape usable for our purpose it must be (re-)

developed. For this model, a data-driven approach is not a problem; on the contrary it is even preferred.

Next, the different audience object models are expressed as (possibly complex) views on this BOM. Figure 8 illustrates how the different types of conceptual schemes developed during Information Modelling related to each other.

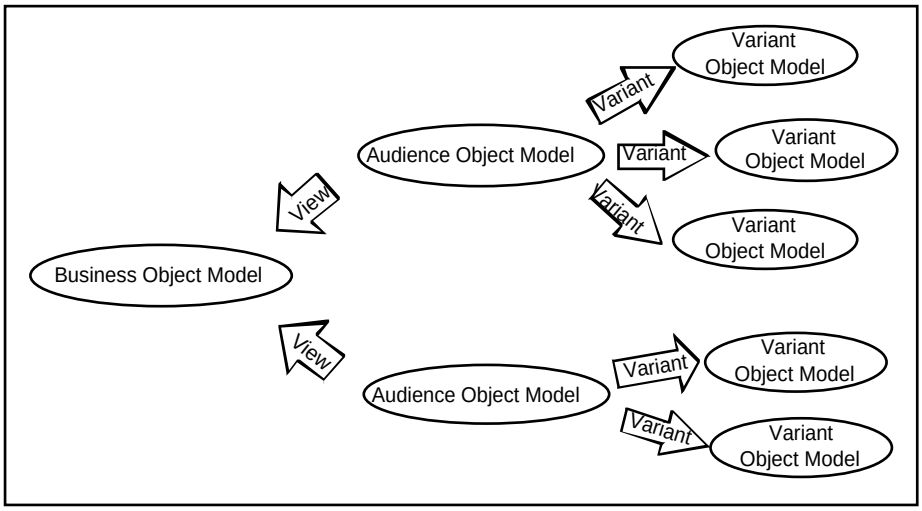

Figure 8: Relationship between the different types of object models in WSDM.

Navigation Design

During navigation design, the second step of the conceptual design, we describe the (conceptual) structure of the web site and model how the members from different audience classes will be able to navigate through the site. For each audience class (variant) a different *navigation track* is created. All navigation tracks together form the *navigation model*.

A navigation track is described in terms of *components* and *links*. Components represent units of information. Such a unit can contain text, pictures and some other kind of multi media information as well as structured information modelled by means of the audience object models. Components are connected by means of links. We use links to model the structure of the web site but also to indicate the need for navigation links. We recognise that our concept of link is overloaded. However, there is (currently) no need to distinguish (explicitly) between these different types of links. Both types of links (as well as other type of links) will be translated into navigation links in the current web technology. There is no added value for the modeller to bother about the difference, also not from a modelling point of view.

The structure described by the components and links in the navigation model is a conceptual one. The translation (in the implementation design) into implementation pages

and links need not to be one-to-one. As explained later, a single component may be split into different pages or different components may be grouped into a single page.

To arrive at the navigational model, we construct a navigation track for each audience class (variant). The navigation track for an audience class starts with a single component that will be used to identify the track. This component is the *top* of the navigation track. See figure 9 for an example. The rest of the navigation track is built taking into consideration the requirements formulated for the audience class. The process to do this is similar to the process of building the audience object model: relevant requirements are selected and elaborated, and are decomposed into *elementary requirements*. For each such elementary requirement we build a so-called *navigation chunk*, next these navigation chunks are combined into a single navigation track. Clearly, the purpose for using navigation chunks is the same as for object chunks, i.e. to easily track changes to requirements in the design.

Information requirements are translated into components. Navigation requirements are translated into links. If a component deals with structured data it is associated with the object chunk(s) that model this data. If all information requirements of an audience class are taken into consideration all information modelled in the audience object model is covered by components.

To be able to access all information represented by the components each component should be within reach (directly or indirectly) from the top of the navigation track. Further on, the components and links form a directed graph. This means that the same component may be accessible in different ways.

Figure 9 and figure 10 show the navigation tracks for respectively the audience class variant **Exchange Students** and **Potential Students**.

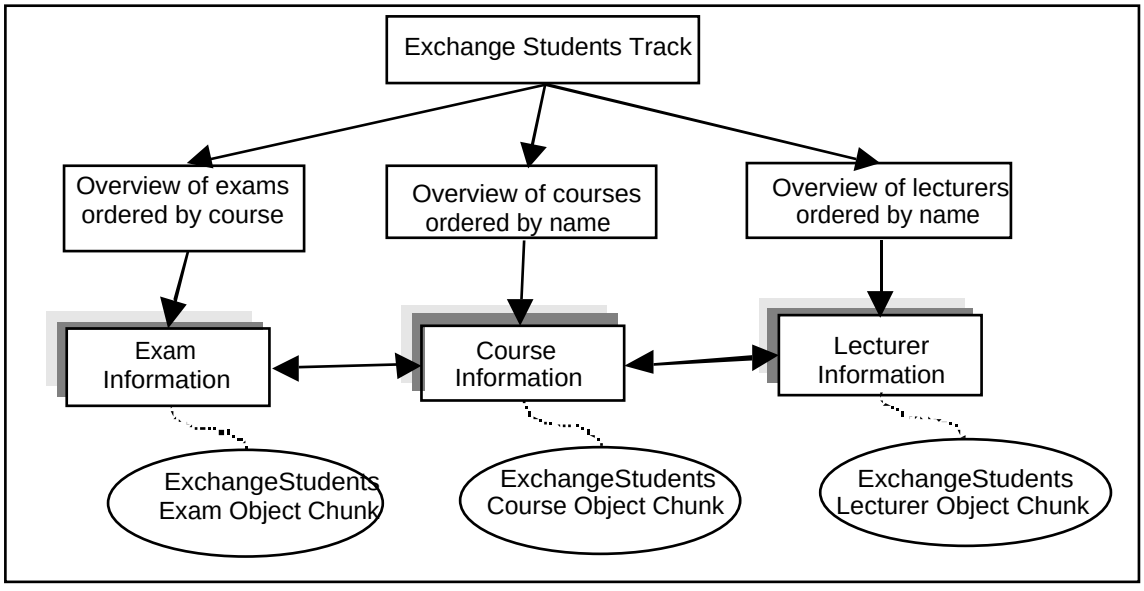

Figure 9: Navigation track for Exchange Students.

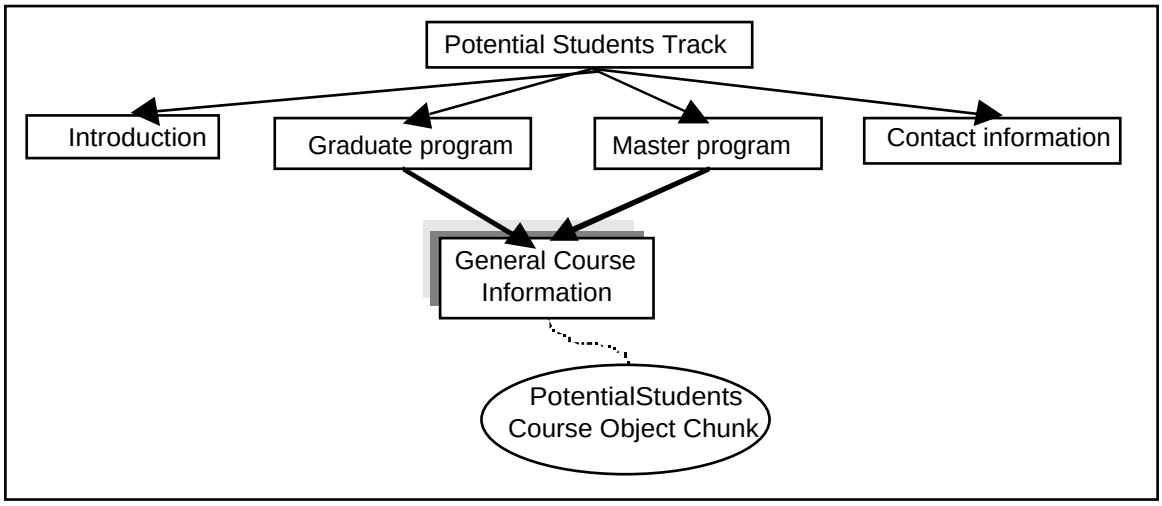

Figure 10: Navigation track for Potential Students.

Finally, if all navigation tracks are constructed, linking the tops of the different navigation tracks to a new 'top' component derives the navigational model. This component will provide the information that allows a user of the web site to identify the navigation track that is most appropriated for him in a given situation. See figure 11 for an illustration. Also note that the different navigation tracks are disjoint.

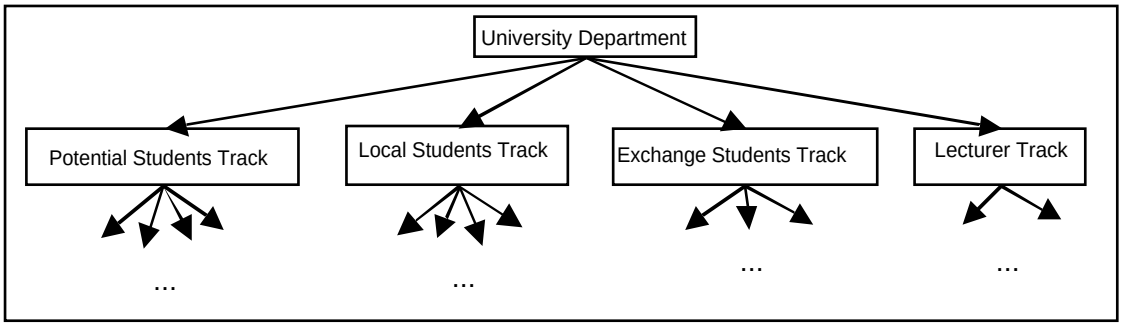

Figure 11: Composition of navigation tracks into a navigation model.

CONCLUSIONS

WSDM is a web site design method based on a new approach, called audience-driven. This means that not the data available in the organisation or the available technology but the requirements of the target audience is the starting point of the modelling process. This approach must prevent so-called "jumble sites" and "ego sites". Jumble sites are sites that contain all information the developer could collect and that are organised in a way only obvious for the developer. Ego sites (Powell, 1998) are those sites that are built to satisfy the ego of the developer, not to fulfil user needs.

The method is based on the principle that the web site should be designed for and adapted to its target audience. We have also explained the need for a conceptual design phase in web site design similar to the conceptual design phase in database systems. Therefore, the emphasis in this method is on audience modelling and conceptual modelling rather than on layout and presentation aspects. In principle, an implementation can be generated once the conceptual design is completed.

As a consequence of our audience-oriented approach, the conceptual information schema of a web site cannot be seen as a single schema but as a collection of schemes; each audience class has its own conceptual information schema. To relate the different schemes and to control the redundancy possibly introduced in this way, a Business Object Model is used. To capture small variations between audience class members, OT variants are introduced. Because navigation is an essential characteristic of web sites, the conceptual design also include a navigation model that is a collection of navigation tracks, one for each audience class (variant). A navigation track describes the (conceptual) structure of the web site and shows how the members of the audience class will be able to navigate through the information.

Separating the conceptual and the implementation design has the same advantages as in database design. As for database design it is possible to deploy the conceptual schema technology in the future automation of web site creation, and of their upkeep. CASE-type tools generating well-structured web sites from audience requirements and business domain models are the next logical step.

References

Bichler, M. and S. Nusser (1996), "W3DT - The Structural Way of Developing WWW-sites," In *Proceedings of ECIS'96.*

British HCI Group (1999) "Proceedings of the Active Web Conference," http://www.hiraeth.com/conf/ActiveWeb/

Chen, P.P. (1976), "The Entity-Relationship Model: Towards a Unified View of Data,"*ACM Transactions on Database Systems* 1, 1, 471-522.

Date, C.J. (1990), *An Introduction to Database Systems,* Volume I, fifth edition, Addison-Wesley.

December, J. and M. Ginsberg (1995), *HTML and CGI unleashed,* Sams.net Publishing.

Decruyenare, T. (1999), "The use of XML in modeling Conceptual Schemas in web site design," Master Thesis, Department of Computer Science, Vrije Universiteit Brussel.

De Troyer, O. M.F. (1996), "A formalization of the Binary Object-Role Model based on Logic," *Data & Knowledge Engineering* 19, 1-37.

De Troyer, O.M.F. (1998), "Designing Well-Structured Web Site: Lessons to be Learned from Database Schema Methodology," In *Proceedings of the ER'98 Conference*, Lecture Notes in Computer Science (LNCS), Springer-Verlag.

De Troyer, O.M.F. and T. Decruyenare (2000), "Conceptual Modelling of Web Sites for End-Users," *WWW Journal* 3, 1, Baltzer Science Publisher.

De Troyer, O. and C. Leune (1998), "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.

Goedefroy, W., R. Meersman and O. De Troyer (1998), "UR-WSDM: Adding User Requirement Granularity to Model Web Based Information Systems," In *Proceedings of 1st Workshop on Hypermedia Development*, Pittsburgh, USA.

Garzotto, F., P. Paolini and L. Mainetti (1993), "Navigation patterns in hypermedia databases," In *Proceedings of the 26th Hawaii International Conference on System Science*, IEEE Computer Society Press, pp. 370-379.

Garzotto, F., P. Paolini and D. Schwabe (1993), "HDM - A Model-Based Approach to Hypertext Application Design," *ACM Transactions on Information Systems* 11, 1, pp. 1-26.

Halpin, T. (1995), *Conceptual Schema and Relational Database Design*, Second Edition Prentice Hall Australia.

Isakowitz, T., E. A. Stohr and P. Balasubramanian (1995), "RMM: A Methodology for Structured Hypermedia Design," *Communications of the ACM* 38, 8, pp. 34-43.

Jacobson, I. et al. (1993), *Object-Oriented Software Engineering, A Use Case Driven Approach*, Addison-Wesley Publishing Company.

Lee, H., C Lee and C. Yoo (1998), "A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems," In *Proceedings of HICSS '98*.

Maurer, H. (1996), *Hyper-G - The Next Generation Web Solution*, Addison-Wesley.

Nielsen, J. (1995), *Multimedia and Hypertext the Internet and Beyond*, Academic Press.

Powell, T.A. (1998), *Web Site Engineering, Beyond Web Page Design*, Prentice Hall PTR.

Redmond-Pyle, D. and A. Moore (1995), *Graphical User Interface Design and Evaluation, A practial Process*, Prentice Hall.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen (1991), *Object Oriented Modeling and Design*, Prentice Hall Inc..

Sano, D. (1996), *Designing Large-scale Web Sites*, Wiley Computer Publishing.

Schwabe, D. and G. Rossi (1995), "The Object-Oriented Hypermedia Design Model," *Communications of the ACM* 38, 8.

Schwabe, D. and G. Rossi (1998), "An Object-oriented approach to web-based application design," Theory and Practice of Object Systems (TAPOS).

Schwabe, D., G. Rossi and S.D.J. Barbosa (1996), "Systematic Hypermedia Application Design with OOHDM," In *Proceedings of the ACM International Conference on Hypertext (Hypertext '96*, http://www.cs.unc.edu/barman/HT96/P52/section1.html

Takahashi, K. and E. Liang (1997), "Analysis and Design of Web-based Information Systems," In *Proceedings of Sixth International World Wide Web Conference*, http://www6.nttlabs.com/papers/PAPER245/Paper245.html

J.J. Wintraecken, J.J. (1990), *The NIAM Information Analysis Method - Theory and Practice*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.

[1] We used to call it *user-centered*; however we renamed it to avoid confusion with the use of this term in the field of HCI, where is has a somewhat different meaning. The difference will be explained later on.