Structuring Web Sites Using Audience Class Hierarchies

Sven Casteleyn and Olga De Troyer
Department of Computer Science, Vrije Universiteit Brussel,
Pleinlaan 2, 1050 Brussel, Belgium
svcastel@vub.ac.be, Olga.DeTroyer@vub.ac.be

Abstract. WSDM is an audience driven design method for web sites. By explicitly starting from the requirements of the web sites audience (the users or visitors), WSDM avoids problems caused by poor underlying design, or by a too data or organization driven view. This paper presents how the main structure of a web site can be derived from structuring the visitors of the web site into one or more so-called Audience Class Hierarchies. Each Hierarchy represents a classification of the visitors according to one aspect. All Hierarchies have the same top, i.e. the class Visitor representing all potential visitors of the web site. Furthermore, the presented methodology forces its user to deeply reflect on the requirements of the visitors, and to resolve any semantic conflict at design time. This greatly enhances the correctness of the obtained Audience Classes. The given algorithm also allows for computer-aided support of WSDM.

1 Introduction

As primary use of the Internet is more and more evolving to commercial purposes, an exploding amount of web sites and information is being offered through the World Wide Web (WWW) today. Most of these web sites are built without any underlying systematic design, instead, web designers rather focus on the graphical presentation of the site to be as hip or flashy as competitor's sites. Adding the intrinsic evolutionary nature of web sites, and their constantly changing information, current maintenance and usability problems become obvious (see [5] for a description of some problems).

To address these problems, different design methods have been proposed: HDM and its successors HDM2 [9] and OOHDM [15] [16], RMM [11], W3DT [1] [2], SOHDM [13], WEBML [3] [4] and WSDM [5] [6] [7] [8] [10] [18]. WSDM uses an audience driven approach rather than a data driven approach: instead of letting the available data drive the design of the web site, WDSM uses the requirements of the intended users to drive the web site design process. This makes the Audience Modeling phase of WSDM a crucial phase in the design process.

The purpose of this paper is twofold. First we will describe the current Audience Modeling phase of WSDM and introduce the concepts of "Audience Subclasses" and "aspect oriented Audience Class Hierarchies". Next, we present an algorithm to automatically create the main structure of the web site starting from the different

Audience Class Hierarchies. This algorithm also has the advantage that it allows validating the Audience Class Hierarchies.

This paper is organized as follows: section 2 gives a short overview of WSDM. In section 3, Audience Classes and Subclasses are introduced. Section 4 introduces the advantage of different Audience Class Hierarchies for a single web site. Section 5 explains how the main structure of the web site can be derived from the Audience Class Hierarchy in case of a single hierarchy and in section 6 this is extended to the case of multiple Audience Class Hierarchies. Finally, section 7 gives conclusions.

2 WSDM: An Overview

The main characteristic of WSDM is the audience-driven approach. 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 create a web site based on the requirements of the intended audience(s). In this way, WSDM gives consideration to the fact that web sites usually have different types of visitors that may have different needs.

A second important characteristic of WSDM is the distinction between the conceptual design (which is free from any implementation detail) and the design of the actual presentation: 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) and the logical schema (e.g. a relational schema). It allows making web site designs that are not biased by the diversity and rapid growing obsolescence of the web technology.

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 Modeling 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 Characterization, the characteristics of the different Audience Classes are given. The result of the Audience Modeling is a set of Audience Classes together with an informal description of their requirements: information- and functional - as well as navigational- and the usability requirements, and their characteristics.

Next, we perform a Conceptual Design. The Conceptual Design phase is divided in three steps: Information Modeling, Functional Modeling and Navigational Design.
During Information Modeling, Information Chunks are created. These chunks model the information requirements of the different Audience Classes. The different Information Chunks are linked together by a single information model, called the Business Information Model. All Information Chunks are defined as views on this model. In this way, possible redundancy is described and therefore can be controlled.

During Functional Modeling, the functionality needed for the different Audience Classes is described. This is done using Functional Chunks.
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 a Navigation Track is created. Navigational requirements are taken into consideration in this step. All Navigation Tracks together form the Navigation Model of the site.
The integration of the Information Chunks and Functional Chunks in the Navigation Model is called the Conceptual Model of the web site.

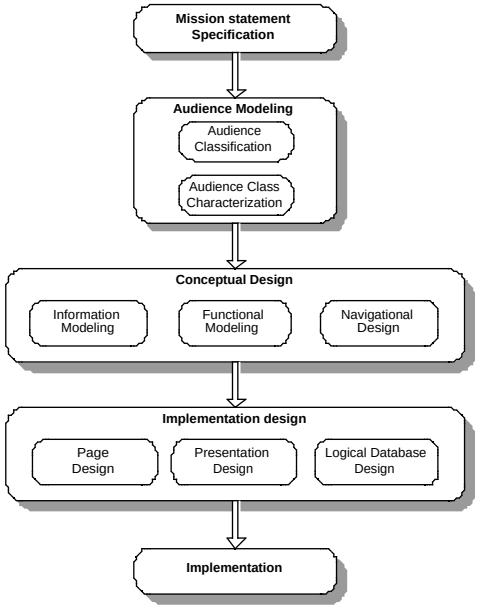

Figure 1: WSDM overview

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 by taking into consideration the usability requirements and characteristics of the Audience Classes. The design of the page structure (the grouping of information in pages) starts from the Navigation Model. 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 (which can be derived from the Business Information Model).

The last phase, Implementation, is the actual realization of the web site using the chosen implementation environment, e.g. HTML or XML. Depending on the complexity of the web site, parts of it can be automated using available tools and environments for assisting in HTML or XML implementations.

3 Audience Classes and Subclasses

We will now consider into more detail the second phase, the Audience Modeling Phase. To obtain the Audience Classes for a web site, WSDM looks at the activities of the organization that are related to the purpose and subject of the web site. Each activity involves people, which are potential users of the site if they belong to the target audience of the mission statement. If necessary, the activities are decomposed in order to refine in each decomposition step the target audience. By definition, users belonging to the same audience class have the same (information and functional) requirements. Whenever the requirements differ, a new Audience Class is made.

***Definition*:** an *Audience Class* is a group of potential visitors that belongs to the target audience of the mission statement, and has the same information and functional requirements. Graphically, we represent an Audience Class A as follows (figure 2):

Figure 2: Graphical Representation of an Audience Class

We illustrate this process with an example: the Conference Review System (for the complete specification of this case study see [14]). The mission statement for this site could be formulated as follows: *"To support the overall selection process (submission by authors, evaluation and selection by the Program Committee) of papers for a conference*". So, the purpose of the site is to support the paper selection process; the target audiences are authors and PC-program Committee; and the subject of the web site are papers for a conference. To refine this target audience into Audience Classes we look at the activities related to the purpose and subject of the web site, and examine the people who are involved in these activities. For the Conference Review System, the activities are Paper Submission, Assignment of Papers to PC Members, Assignment of Papers to Reviewers, Entering a review, Selecting Papers, Notify Authors (see figure 3). The people involved in these activities are Authors, PC Chair, PC Members, and Reviewers. To decide whether these can be one Audience Class or we need several Audience Classes, we look to their requirements. Due to lack of space, we have simplified the example to the most important requirements.

Authors

Functional requirements*:* submit paper, change submission, pre-register co-author
Information requirements*:* information about own submission

PC Chair

Functional requirements: create conference, pre-register PC Member, pre-register Reviewer, assign papers to PC Member, mark paper as accepted/rejected
Information requirements: view all available information

PC Members

Functional requirements*:* pre-register Reviewer, re-assign paper to reviewer, download papers assigned to him, submit review, advice PC Chair.

Information requirements*:* list of papers, view own reviews, state of reviews of other reviewers of papers assigned to him

Reviewer

Functional requirements*:* download papers assigned to him, submit review

Information requirements*:* view own reviews

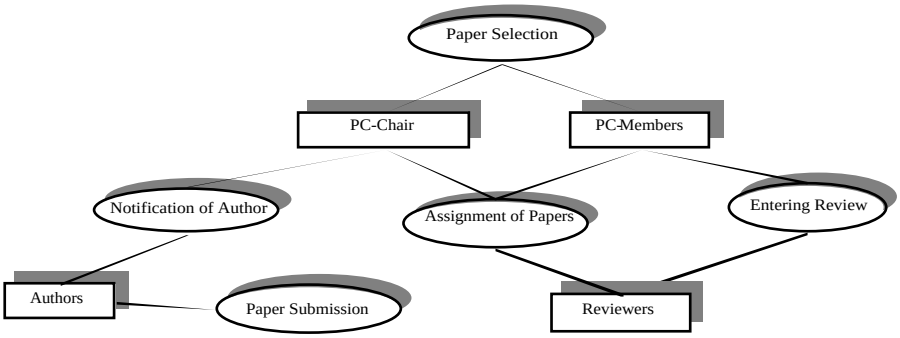

Figure 3: Activity diagram for the Conference Review System

The requirements for these different groups are sufficiently different to put them in separate Audience Classes. This results in four different Audience Classes: Authors, PC Chairs, PC Members and Reviewers.

However, notice that the set of requirements of the Audience Class Reviewer is a subset of the requirements of the audience class PC Member. Actually, a Reviewer needs to do exactly the same things as a PC Member, but in addition, PC Members have some extra needs: e.g. a PC Member has to be able to assign papers to Reviewers, indicate his preferences for some tracks/topics. Such a situation appears quite common. Therefore, in analogy to the superclass-subclass relationship in OO, we have introduced the concept of Audience Subclass.

Definition An Audience Class B is an *Audience Subclass* of an Audience Class A if B has all the requirements of A and some extra. Graphically, we represent the audience super/sub classes as follows (figure 4):

Figure 4: Audience Subclass

In this way, we can create an Audience Class Hierarchy. Similar as in some OO programming languages, the Audience Hierarchy has a single top. In WSDM this common superclass is called *Visitor*. The Audience Class Visitor represents all

potential users of the web site, including those that accidentally comes to the web site, and have no specific needs. Every Audience Class is a subclass of Visitor.

Considering the example of the Conference Review System, we have identified an audience subclass relationship between Reviewer (parent) and PC Member (child). The complete Audience Class Hierarchy (identified so far) is given in figure 5.

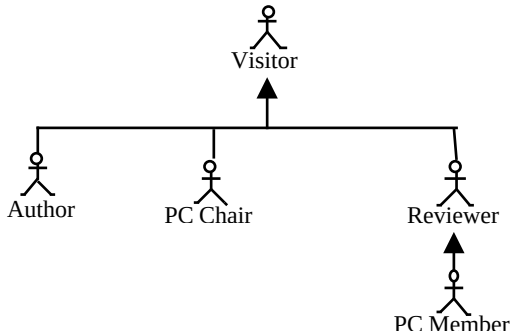

Figure 5: Audience Class Hierarchy

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) this people in the development process (like for intranets), we can use the standard techniques of software engineering like questionnaires, interviews, etc. to collect the requirements. For most public web sites we are usually unable to involve the target audience itself in the development process. Therefore it looks as if we have to "guess" for their requirements. Studying the characteristics of the audience 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.

Later on, during Navigational Design, a Navigation Track will be made for every Audience Class. Such a Navigation Track will fulfill all information- and functional requirements formulated for the Audience Class. Because for an Audience Subclass, part of its requirements is modeled at the level of its superclass, a "hierarchical" structure similar to the Audience Class Hierarchy will be present between the Navigation Tracks. This is explained into more detail in section 5.

4 Different Subclass Hierarchies

In building the Audience Class Hierarchy for the Conference Review System example, until now, we neglected the aspect of security formulated in the case study. The site needs to be protected against unauthorized use by means of a login. In addition, users can be pre-registered in which case they first need to confirm their

registration. Using the same method as before, we can derive an(other) Audience Class Hierarchy only considering the requirements concerning authorizations.

Registered Users
Authorization requirements: logging in
Pre-Registered Users
Authorization requirements: logging in, confirm registration
Not-Registered Users
Authorization requirements: register

This gives us the hierarchy of figure 6. We call this the Authorization Class Hierarchy because the focus in this hierarchy is the authorization requirements. Note that this hierarchy has also the class Visitor as its top.

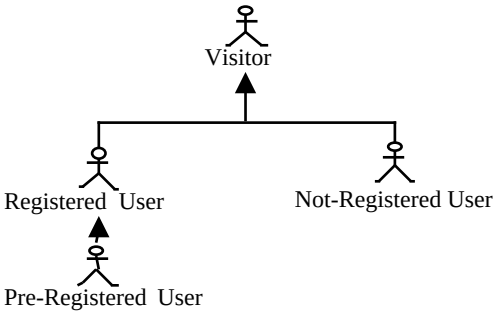

Figure 6: Authorization Class Hierarchy

Even though we also could consider the authorization requirements together with the other requirements, we find it easier to separate them from the others and to concentrate on one aspect at a time. In this way several Audience Class Hierarchies can be constructed, representing different aspects of the web site. Off course, for whatever web site, we always have at least one Audience Class Hierarchy, the general Audience Class Hierarchy. This kind of abstraction mechanism, used during modeling, can be compared in a sense to the concept of aspect oriented programming [12]. This is a very powerful concept and used in web design it allows simplifying modeling and results in easier specifications.

For some Audience Class Hierarchies, Audience Class Transitions can be specified. In the example of the Authorization Class Hierarchy, an Author may become a Registered User simply by registering; other users need to be Pre-Registered. The Audience Class Transitions for the example are given in figure 7.

The introduction of different Audience Class Hierarchies has some impact on the construction of the Navigation Tracks during the Conceptual Design. First, we describe the general principles for constructing the Navigation Schema in section 5.

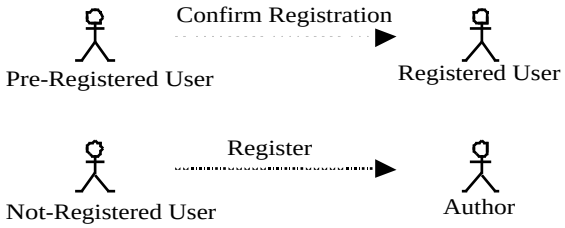

Figure 7: Transition Diagrams for the Authorization Class Hierarchy

5 Navigational Design

As already indicated in section 2, during Navigational 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 a different *Navigation Track* is created. All Navigation Tracks together form the *Navigation Model.*

A Navigation Model is described in terms of *tracks*, *components* and *links*. Components represent units of information or functionality. They are connected by means of links. Links are used to model the structure of the web site as well as to indicate the need for navigation. We can put *conditions* on links to indicate that the availability of the link is dependent on the truth-value of a condition. Figure 8 gives the graphical notation for tracks, components and links. A multiple link is used to indicate that one component is linked to several instances of the other component.

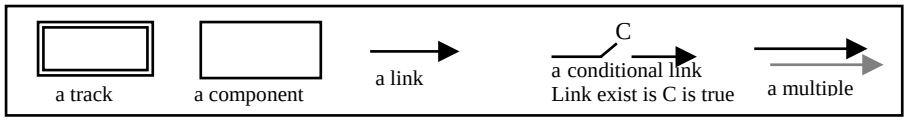

Figure 8: Graphical Representation of Tracks, Components and Links

The main structure of the Navigation Model can be derived from the different Audience Class Hierarchies. As already indicated, with each Audience Class will correspond a Navigation Track. To link the different tracks, the main idea is to follow the sublink structure of the different Audience Class Hierarchies. If there is only one audience class hierarchy, this is rather easy. The sublink structure of the Audience Class Hierarchy can be mapped in a one-to-one way into a track structure. Figure 9 shows the main track structure as derived from the general Audience Class Hierarchy of our Conference Review Site (given in figure 5). Note that for the sake of simplicity all links are represented as non-conditional links. The next step in the Navigational Design is to elaborate each Navigation Track into more detail. This is outside the scope of this paper; we refer to [7] for more details.

If there is more than one Audience Class Hierarchy, the translation into a Navigational Schema becomes more complicated. Actually, the different hierarchies need to be integrated ('weaved") into a single hierarchy before we can map them into a Navigation Track structure. This will be explained in the next section.

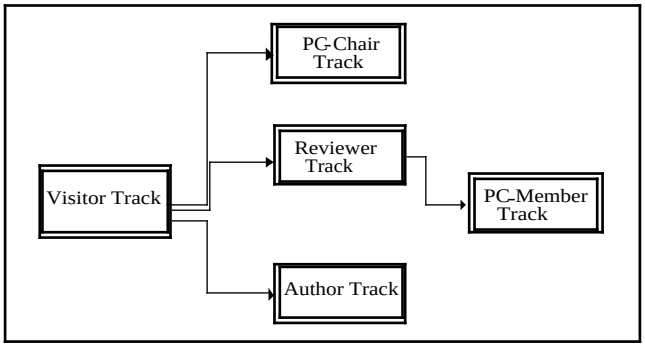

Figure 9: Main Track Structure based on the General Audience Class Hierarchy

6 Mapping Audience Class Hierarchies into Navigation Tracks

The one-to-one mapping of an Audience Hierarchy into Navigation Tracks described in the previous section works fine as long as there is only one Audience Hierarchy. However, when other aspects come in, we need a way to merge ("weave") the different hierarchies into a single one before we can make a mapping to the Navigation Tracks. We will present an algorithm to do this[1]. This algorithm is inspired by an algorithm given in [17], used to generate the object type structure for a conceptual schema. Before we can give the algorithm, we need some definitions.

Definition: An *Audience Class Matrix* M for a web site is an n by n matrix where each row i (with $0 < i \leq n$) of M is associated with some user requirement UR[i], and each column i is associated with the same user requirement UR[i]. n is the total number of user requirements. The entries for the Audience Class Matrix M are 'Y' (yes) or 'N' (no), having the following meaning:

> The entry on row i, column j is the answer to the question "Does every user who has the requirement UR[i], also have the requirement UR[j]?"

Figure 10 presents the Audience Class Matrix for the example web site. For simplicity and lack of space, we have only taken into account the functional requirements and omitted the information requirements. As an example, the entry on row 1 and column 2 tells us that every user who can submit a paper can also change a submission. The entry on row 1 and column 4 tells us that not every user who can submit a paper can

[1] Actually, the algorithm does more than just merging Audience Classes. We will go into further detail at the end of the section.

create a conference (i.e. the answer on the question "Does every user who may submit a paper, may also create a conference" is No).

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	Submit paper	Y	Y	Y	N	N	N	N	N	N	N	N	N	Y	N	N
2	Change submission	Y	Y	Y	N	N	N	N	N	N	N	N	N	Y	N	N
3	Pre-register co - author	Y	Y	Y	N	N	N	N	N	N	N	N	N	Y	N	N
4	Create conference	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
5	Pre-register PCMember	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
6	Pre-register Reviewer	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
7	Assign paper to PCMember	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
8	Mark Paper as accepted/rejected	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
9	Re-assign paper to Reviewer	N	N	N	N	N	Y	N	N	Y	Y	Y	Y	Y	N	N
10	Download paper(s) assigned to review	N	N	N	N	N	N	N	N	N	Y	Y	N	Y	N	N
11	Submit review	N	N	N	N	N	N	N	N	N	Y	Y	N	Y	N	N
12	Advice PCChair	N	N	N	N	N	Y	N	N	Y	Y	Y	Y	Y	N	N
13	Login	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N
14	Confirm registration	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	N
15	Register	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y

Figure 10: Audience Class Matrix

Evidently, the diagonal of any Audience Class Matrix will contain all 'Y' values. Also not all entries in the matrix are unrelated: if the question to UR's (x,y) and (y,z) was yes, then also (x,z) must have an affirmative answer. In other words, the Audience Class Matrix is transitively closed. This is a useful proposition. After the designer has filled in the Audience Class Matrix we can check this property. If the matrix is not transitively closed it is usually due to some semantic errors.

Informally, every column *i* of an Audience Class Matrix represents which users can do UR[*i*]. E.g. column 1 represents all the users who are able to submit a paper. In this case the answer to the question "who may submit a paper?" is “evidently the users that can submit a paper, the users that can change a submission and the users that can pre-register a co-author”. Column 13 represents the users who can login ("who may login?"). If two or more columns are exactly alike, it means that the users that are represented by these columns are actually the same, and thus they belong to the same Audience Class. However, if the ‘Y’ entries of a column i is a subset of the ‘Y’ entries of another column j, this means that the set of users represented by column i will be a subset of the set of users represented by column j. Indeed, in the example, the users who may submit a paper (column 1) are a subset of the users who may login (column 13). Therefore we can define a subclass relationship between the Audience Classes

represented by the columns *i* and *j*. In our example, the set of users that can submit a paper is a subclass of the set of users that can login.

Formally, we define the above notions as follows:
***Definition*:** For every Audience Class Matrix we can define a partial order relation '<' on the columns of the matrix as follows:

column i < column j
if and only if
for every row r, if column i contains a 'Y' on row r, then also column j contains a 'Y' on row r.

Informally, this means that for every row where column i has a Y, column j *must* also have a Y. In addition, column j may have Y's on rows where column i has N's.

Definition: Two columns i and j of an Audience Class Matrix are *equivalent*, if and only if

column i < column j and column j < column i.

Equivalent columns are called *equivalent classes*, they form a potential Audience Class for the system.

Obviously, when column *i* < column *j*, then the < relation remains valid when we substitute column *i* by any of it's equivalent columns, or column *j* by any of it's equivalent columns. As a shortcut notation, we might use the < relation between equivalent classes.

From the definition of Audience Subclasses, the Audience Class Matrix and the '<' relation, we notice that the '<' relation actually defines a subclass relation between the potential Audience Classes. Column *i* < column *j* implies that the Audience Class defined by the equivalence class of column *i* is a subclass of the Audience Class defined by the equivalence class of column *j*.

We can now give the algorithm to compute the integrated Audience Class Hierarchy:
Algorithm:
Step 1. Compute the Audience Class Matrix based on all the requirements formulated for the web site to be built.
Step 2. Compute the equivalence classes, and give them a meaningful name (these are the potential Audience Classes). The user requirements for an Audience Class are the requirements associated with the columns in the corresponding equivalence class
Step 3. Identify the '<' relation between the different equivalence classes.
Step 4. Compute the integrated Audience Class Hierarchy (using the Audience Classes and the '<' relations between them).

We illustrate the algorithm for the (simplified) Conference Review System example.
Step 1: As already indicated, the Audience Class Matrix is given in figure 10.
Step 2: The following equivalence classes are identified and given meaningful names. Note that the names usually correspond with the names of the Audience Classes

obtained during Audience Classification. However, new classes may appear (e.g. for column 6):

- Columns 1, 2 and 3: Author
- Columns 4, 5, 7, 8: PC Chair
- Columns 9, 12: PC Member
- Columns 10, 11: Reviewer
- Column 13: Registered User
- Column 14: Pre-Registered User
- Column 15: Non Registered User
- Columns 6: User with Pre-register Reviewer right

Step 3: We identify the following < relations:

PC Member < Reviewer
Author < Registered User
PC Chair < Registered User
Reviewer < Registered User
PC Member < Registered User
Pre-Registered User < Registered User
User with Pre-register Reviewer right < Registered User
PC Member < User with Pre-register Reviewer right
PC Chair < User with Pre-register Reviewer right

Step 4: With these relations, and the Audience Classes obtained in step 2, we find the following integrated Audience Class Hierarchy (figure 11).

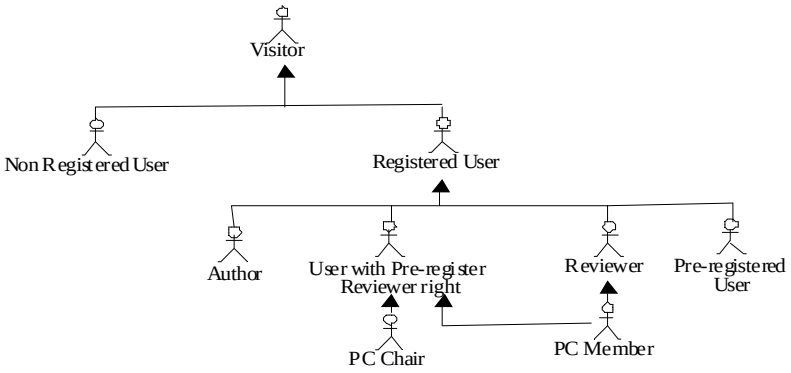

Figure 11: Integrated Audience Class Hierarchy

We notice that one of the potential Audience Classes, "User with Pre-register Reviewer right", was not yet identified before (in section 3 or 4). When we look to the requirements, indeed, we see that PC Chairs and PC Members actually have in common that they can both pre-register a Reviewer. However, as this is their only common requirement, this audience superclass is a rather artificial one. Therefore we decide not to take this class into consideration and to leave the pre-register

requirement in both classes PC Chair and PC Member. The final integrated Audience Class Hierarchy is given in figure 12.

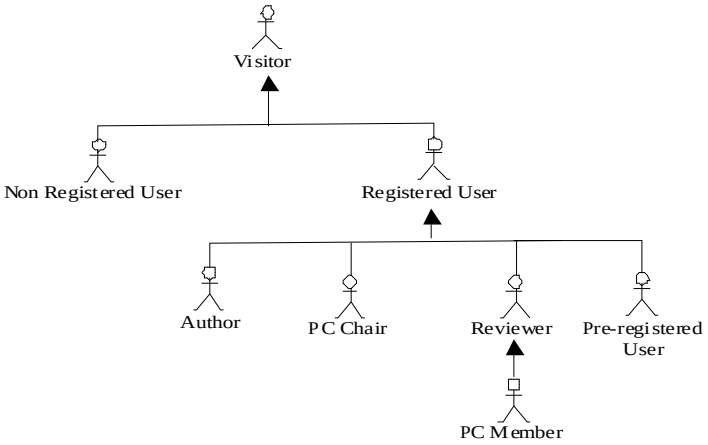

Figure 12: Final Integrated Audience Class Hierarchy

Having obtained the final integrated Audience Class Hierarchy, we can now, as described in section 5, use a simple one-to-one mapping to create the Navigation Track structure. For the example, this structure is given in figure 13.

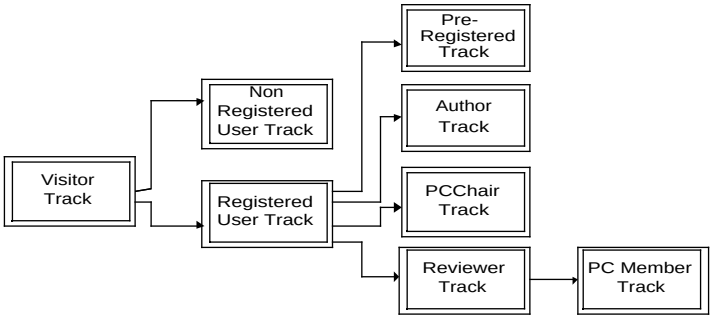

Figure 13: Navigation Track Structure for the Conference Review System

However, so far we did not yet consider the Audience Class transition diagrams (given in section 4). These diagrams model dynamic transitions between Audience Classes. These can be mapped by defining a conditional link between the Navigation Tracks derived for the Audience classes involved. Figure 14 shows this mapping for the Transition Diagrams given in figure 7. Finally, figure 15 shows the complete Navigation Track Structure.

As can be seen from the example, the Audience Class Hierarchy Weaving algorithm does more than only merging different hierarchies. By using the requirements

gathered for the different users as described in section 3, and by using these requirements to compute the Audience Class Matrix, the designer is forced to deeply reflect on these requirements. In the mean time, his work is validated, and can be corrected if necessary. In addition, the designer gets the integrated Audience Class Hierarchy, from which the Navigation Tracks can be derived easily.

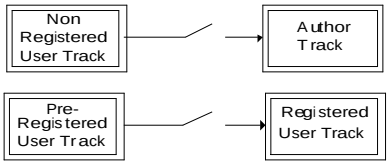

Figure 14: mapping Transition Diagrams into the Navigation Track

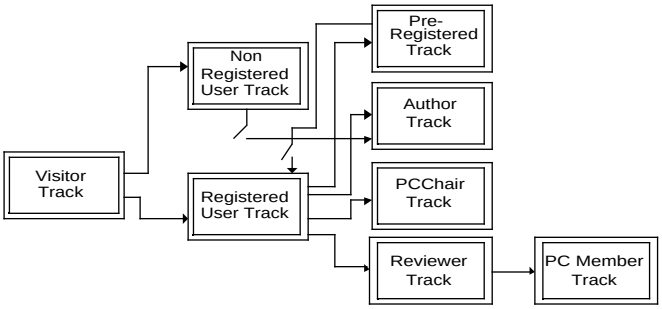

Figure 15: Final Navigation Track Structure for the Conference Review System

7 Conclusions

In this paper we have explained and demonstrated a method that, based on the requirements of the different (potential) users (audiences), systematizes the design of the main structure of a web site. The method is defined in the scope of an audience driven web site design method, WSDM. The resulting structure of the web site is therefore based on the different Audience Classes identified for the web site. The Audience Classes can be identified in an aspect-oriented approach. This allows designers to concentrate on the user requirements of one aspect at a time. The different aspect oriented Audience Class Hierarchies derived in this way are weaved together using the Audience Class Weaving Algorithm. The method also allows some checks on consistencies and may detect ambiguities. In addition the designer is gently forced to precisely formulate the requirements of the different potential users of the web site. Implementation of the method seems feasible and would be very useful in the context of a Computer Aided Web Engineering (CAWE) tool for WSDM.

References

1. Bichler, M., Nusser, S.: "W3DT - The Structured Way of Developing WWW-Sites", Proceedings of the ECIS 1996, Lisbon, Portugal (1996)
2. Bieber, M., Isakowitz T.: "Designing Hypermedia Application", In Communications of the ACM, Volume 38, S. 26 - 29.
3. Bongio, A., Ceri, S., Fraternali, P., Maurino, A.: "Modeling Data Entry and Operations in WebML, Proceedings WebDB, p. 201-214, Dallas, USA (2000)
4. Ceri, S., Fraternali, P., A. Bongio: "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites", Proceedings of the WWW9 Conference, Amsterdam (2000)
5. De Troyer, O.M.F.: “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 (1998)
6. De Troyer, O.: “Audience-driven web design,” In Information modelling in the new millennium, IDEA GroupPublishing, ISBN 1-878289-77-2 (2001)
7. De Troyer, O., Casteleyn, S.: "The Conference Review System with WSDM", IWWOST 2001, http://www.dsic.upv.es/~west2001/iwwost01/, Spain (2001)
8. 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 (1998), p. 85 - 94
9. Garzotto, F., Paolini, P., Baresi, L.: "Supporting Reusable Web Design with HDM-Edit", In Proceedings of the 34th Hawaii International Conference on System Sciences, Maui USA (2001)
10. Goedefroy, W., Meersman, R., De Troyer, O.: “UR-WSDM: Adding User Requirement Granularity to Model Web Based Information Systems,” In Proc. of 1st Workshop on Hypermedia Development, Pittsburgh USA (1998)
11. Isakowitz, T., Stohr, E.A., Balasubramanian: "RMM: A methodology for structured hypermedia design", Communications of the ACM 38, 8 (Aug 1995), p. 34-44
12. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J., Irwin, J.: "Aspect Oriented Programming", Proc. Of ECOOP 1997, Springer-Verslag LNCS 1241, Finland (1997)
13. Lee, H., Lee, C., Yoo, C.: "A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems, Proc. Of HICSS (1998)
14. Schwabe, D.: "A Conference Review System" IWWOST 2001, http://www.dsic.upv.es/~west2001/iwwost01/, Valencia, Spain (2001)
15. Schwabe, D., Guell, N., Vilain, P.: "Modeling Interactions and Navigation in Web Applications", Lecture Notes in Computer Science 1921, Proceedings of the World Wild Web and Conceptual Modeling 2000 Workshop, ER Conference, Springer, Salt Lake City (2000)
16. Schwabe, D., Rossi G.: "An Object Oriented Approach to Web-Based Application Design", Theory and Practice of Object Systems 4(4). Wiley and Sons, New York (1998)
17. Vermeir, D., Nijssen, G.M.: "A Procedure to Define the Object Type Structure of a Conceptual Schema", Inform. Systems Vol. 7 No. 4 p. 329-336 (1982)
18. WSDM website: "http://wsdm.vub.ac.be/"