Methodologies for Web Information System Design

Peter Barna, Flavius Frasincar, Geert-Jan Houben, and Richard Vdovjak
Technische Universiteit Eindhoven
PO Box 513, NL-5600 MB Eindhoven, The Netherlands
{*pbarna, flaviusf, houben, richardv*}*@win.tue.nl*

Abstract

The Web information space is rapidly growing in the size and the diversity of both its data and its audience. A consequence is that Web Information Systems (WIS) in many applications replace existing traditional (not Web based) information systems. Since the nature of WIS differs from the nature of traditional information systems there is a strong demand for design methodologies specifically oriented towards WIS design. The complexity of WIS implies the need for an effective design process and a rigorous and systematic design approach. We argue that besides the quality of the navigation that is typical for Web (hypermedia) data also the adaptation of the presented content is a desired feature of a modern WIS. In this paper we briefly describe the navigation and adaptation design in selected WIS design methodologies, RMM, OOHDM, UWE, and particulary Hera.

1. Introduction

The most evident difference between WIS and traditional (non-web) information systems is that the large amount of information is organized in a web structure that is realized via (hyper)links that are available to a large number of potentially divers end-users. As a consequence, WIS need a solid approach to conceptual structuring of the information space and its access (often referred to as authoring) and for engineering and implementing the required access services. The need for fast and effective authoring and the increasing complexity of the systems ask for a rigorous and systematic design process.

Organizing the application's data by offering an appropriate navigation structure helps end-users in finding relevant information while preventing them from getting lost in the web hyperspace. This navigation structure should be effectively materialized into a navigation model as an artefact in the design process.

Traditional web applications use a one-size-fits-all approach when it comes to structuring the content, and in a sense this holds also for traditional information systems. Due to the large and divers audience (with large scale of different platforms) of the Web, the one-size-fits-all approach does not suffice any more, and different aspects of adaptation should be considered (the different aspects and techniques of adaptation are discussed in [2, 3]). Nevertheless, most of the existing design methodologies do not take into account the notion of adaptation. The few that do, materialize the notion of adaptation into an adaptation model as an artefact in the design process.

2. WIS design

Ongoing research efforts have resulted in a number of proposed methodologies for WIS design, mostly model-driven. We discuss RMM, OOHDM, UWE and Hera (as illustrative representatives, although there are more approaches dealing with the design of personalized Web applications as for instance XAHM [3] or WebML [4]), and focus on their navigation and adaptation models.

Typically, the methodologies consider the design process in terms of process phases and their deliverables, often models. A typical WIS design methodology has the following phases:

- **Requirement Analysis**: gathering and forming the specification of the user requirements.
- **Conceptual Design**: constructing the Conceptual Model (CM) for the domain.
- **Navigation Design**: building the Navigation Model (NM) as a navigation view of the application.
- **Adaptation Design**: building the Adaptation Model (AM) and defining all associated mechanisms.
- **Presentation Design**: defining the appearance of the navigation units and their behaviour during user interaction materialized in the Presentation Model (PM).

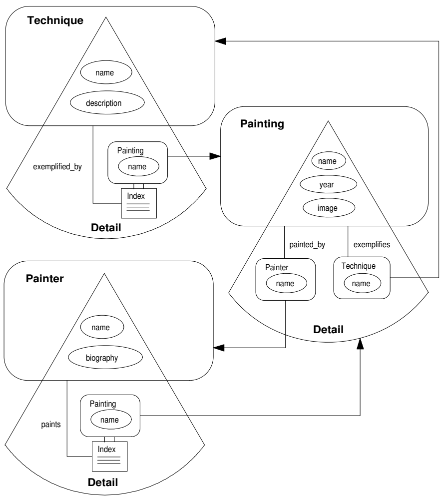

Figure 1. RMM Diagram for example slices

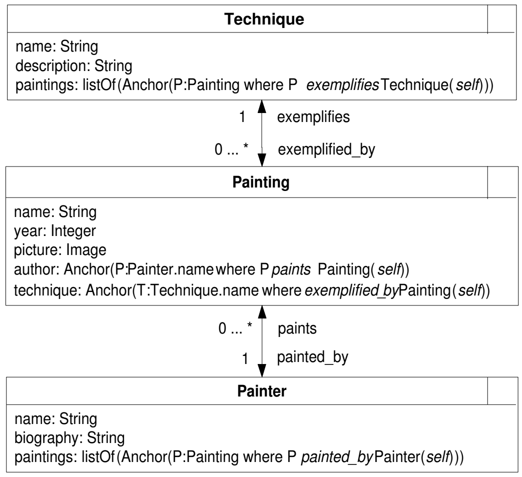

Figure 2. OOHDM Example Navigation Class Schema

- **Implementation**: implementing the application.

To illustrate these methodologies we use a small example composed of three views. The problem domain contains the concepts painter, painting, and painting technique and relationships between them.

3. RMM - Relationship Management Methodology

RMM [9] is a methodology that covers the navigation design phase and uses the Entity-Relationship (E-R) approach for the conceptual modeling.

The NM, in RMM called the Application Model (AM), consists of navigation/presentation units (slices) and relationships among them. Slices represent meaningful chunks of information that typically will be displayed within one web page or a meaningful part thereof. Slices contain sets of attributes from one or more (related) concepts in the CM. For instance, the slice `Technique.Detail` in Figure 3 has the `name` and the `description` attributes from its root concept `Technique` and an index of `name` attributes from the `Painting` concept. The relationships among the slices are of the following two types: aggregations (e.g. `exemplified_by`) used for slice nesting, and references creating (hyper)links between the slices. Slices can contain access structures of the types index (for list access to multiple instances - for instance `Painting.Name`), or guided tour (for sequential access).

Although RMM does not explicitly support adaptation, it is still possible to manually build multiple application models based on the same conceptual model suiting different users or user groups/platforms.

A strong point of RMM is that the problem domain model is clearly separated from the application (navigation) model: for one conceptual model it is possible to build different application models. The simplicity of the methodology and its precise description prove to be a solid foundation for its extensions.

4. OOHDM - Object-Oriented Hypermedia Design Methodology

The OOHDM methodology [15] represents an object-oriented approach to WIS design. The NM in OOHDM consists of the *Navigation Class Schema* and the *Navigation Context Schema*.

The *Navigation Class Schema* contains navigation classes (nodes) derived from the conceptual classes by selecting and combining attributes from (possibly) different related conceptual classes. Attributes are of the type data or of the type hyperlink anchor. Figure 4 gives the navigation class structure of our example.

The *Navigation Context Schema* represents the navigation structure of the application and consists of navigation contexts. The navigation contexts are composed from the navigation classes, hyperlinks, and access structures (e.g. indexes, guided tours, indexed guided tours, menus) and represent collections of navigation classes instances that can be explored in some way (e.g. sequentially). For example, Figure 3 shows a particular instance of a painting technique. The behaviour of the navigation contexts (the conditions for instance selection) is specified in context classes definitions (see [15]).

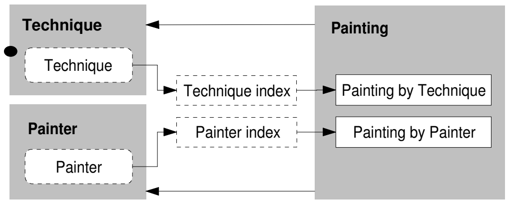

Figure 3. OOHDM Example Navigation Schema

Although a designer can achieve adaptation by realizing multiple NMs (e.g. for different users or user groups) from one CM, there is no specific support for building adaptation in OOHDM. However, there are some proposals for extending OOHDM with adaptation. The approach described in [14] (proposed by the authors of OOHDM) includes only a set of recommendations (and examples) how to achieve different kinds of personalization. The different advised techniques use the OOHDM notation. They are based on the application of object-oriented techniques, on widely used design patterns, and use parameterized navigation context specifications.

OOHDM appears to follow the main lines of the object-oriented design approach and adds specifically the navigation (and presentation) design to the development process. Some modeling aspects are formally specified (e.g. temporal logic is used in the definition of contexts) in OOHDM. The conceptual model and the navigation model are clearly separated.

5. UWE - UML based Web Engineering methodology

UWE [11] represents another object-oriented approach to WIS design. UWE is based on the Unified Software Development Process and uses UML notation. The NM consists of the *Navigation Space Model* and the *Navigation Structure Model*.

The *Navigation Space Model* consists of navigation classes and navigation relationships between them. The navigation classes are derived from the conceptual classes by selecting and combining attributes from the related conceptual classes (similarly as in OOHDM).

The *Navigation Structure Model* enriches the Navigation Space Model by adding access structures (indexes, guided tours, queries, and menus). For comprehensive description of the procedures of building the Navigation Structure Model see [11]. Figure 4 gives the Navigation Space Model and Figure 5 gives the Navigation Structure Model for the sample example. The original navigation space model was

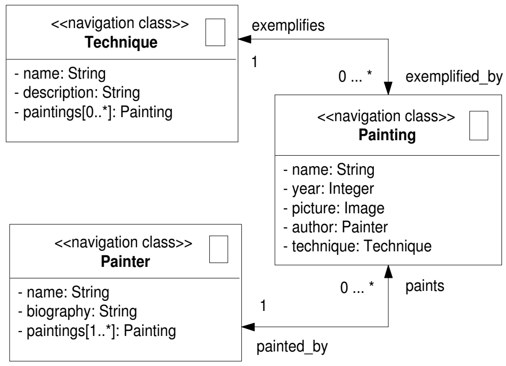

Figure 4. UWE Example Navigation Space Model

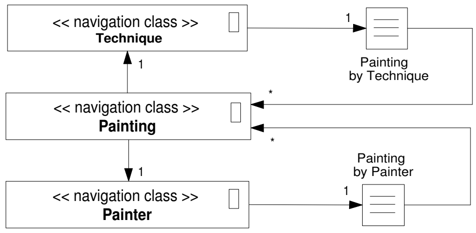

Figure 5. UWE Example Navigation Structure Model

enriched with the indexes `Painting by Technique` and `Painting by Painter`.

Adaptation modeling is covered by the UWE extension referred to as Munich Reference Model [12]. Alike AHAM [6], this model is based on the Dexter Reference Model and refines its Storage Layer into three meta-models:

- **Domain Meta-Model** describes the hypertext structure of the designed system by means of `Component`, `Concept`, and `ConceptRelationships` classes.

- **User Meta-Model** describes models of each individual user via the `User` class.

- **Adaptation Meta-Model**, which describes the mechanism of the adaptation. The adaptation is (similarly as in AHAM) rule-based (see [6] and section 6). The core element is an instance of the class `Adaptation` with attached a set of `Rule` class instances.

The instantiation of the components and the processing of the user events is provided by the *Session Manager* that is part of Dexter's Run-time Layer. The specification of all these (meta-)models is written in UML and OCL.

UWE is a methodology based on the widely used, standard UML object-oriented methodology. Its Requirements Analysis and Conceptual Design phases follow the Unified Software Development Process, while the Navigation Design and Presentation Design phases produce models with UML and OCL semantics. The methodology steps and meta-models are clearly stated and rules allowing formalization and partial automation are present. Nevertheless, there are some limitations inherited from UML as it is for instance vague interpretation of methods behaviour, here partially improved by using OCL for specification of pre- and post-conditions. The conceptual, navigation, and adaptation models are clearly separated.

6. Hera

Hera [7] is our design methodology aiming at automated generation of adaptive hypermedia presentations. Hera provides a specification framework supporting the design of a WIS that generates presentations based on the data retrieved from the data repository in response to a user query.

The architecture of a WIS accordingly to Hera is illustrated by the software suite depicted in Figure 6 and is divided into the three layers:

- The **Semantic Layer** defines the semantics of the data repository in terms of the CM. Since the data repository is virtual, the data instances are retrieved from external (and possibly heterogenous) data sources on demand. The Integration Model links the semantics of the external sources to the semantics of CM.

- The **Application Layer** defines the abstract hypermedia structure of the data in terms of the AM. In addition, the Application Layer defines the adaptation in the generated presentation based on the User Model (capturing dynamic features of the user including browsing history) and the User Profile (capturing static features of the user and his platform).

- The **Presentation Layer** defines the presentation details in terms of the PM (described in [8]) that is needed together with the AM for the generation of presentations (possibly for different presentation platforms as for instance HTML or WML).

All models in Hera are represented using RDF(S) [1, 13], and queries are in RQL [10].

The CM is built from concepts and concept properties, and both are organized in hierarchies based on specialization.

The AM specifies the navigation structure on top of the CM and is based on the Application Model defined in RMM [9].

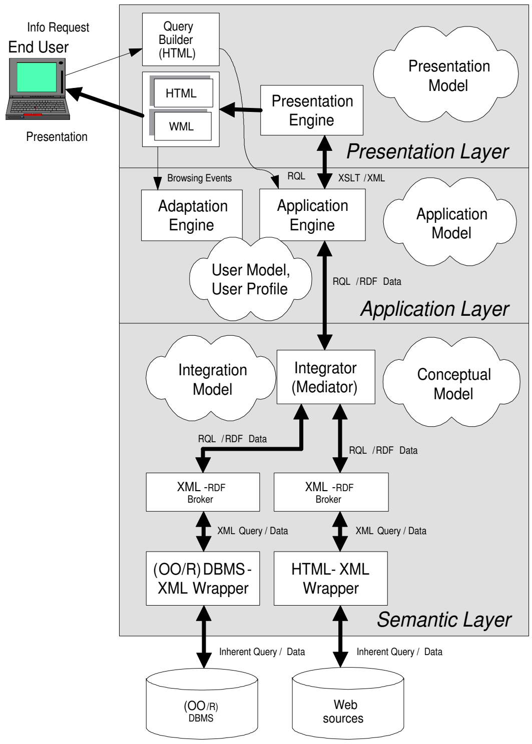

Figure 6. The layers of Hera suite

In Hera the static adaptation (adaptability) is based on the static properties such as the user profile and platform properties (see the condition `prf:ImageCapable=Yes` in Figure 7), while the dynamic adaptation (adaptivity) is based on the user model that changes during the browsing the presentation.

Adaptivity in Hera is considered in terms of AHAM (Adaptive Hypermedia Application Model) [6]. The implementation is based on AHA! [5].

AHAM defines in the Storage layer the following three models:

- The **Domain Model** defines the concepts used for adaptivity. In Hera it consists of concepts/slices from the CM and AM.

- The **User Model** defines user interests by means of a table of concept-value pairs. The concept in the pair corresponds to a Domain Model concept and the value is an integer (from 0-100) defining the relevancy of the concept to the user.

- The **Adaptation Model** consists of *Generate Rules* and *Requirement Rules*, all of the event-condition-action type. *Generate rules* are triggered when the top-slices (pages) are visited. The rules update the values of the attributes in the user model (corresponding

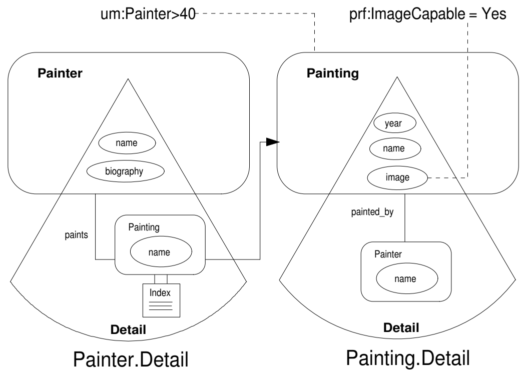

Figure 7. Hera graphical representation of adaptation in slices

to slices). *Requirement rules* are adaptivity conditions (based on the concepts from UM) attached to slices (`um:Painter>40` in Figure 7). The validity of the conditions influence the appearance of the slices and their incoming links.

7. Conclusions and future development

Thanks to its simplicity and clarity RMM can be (and is) easily extended. OOHDM and UWE are object-oriented state-of-the-art methodologies that can be used by WIS designers. They both have extensions for adaptivity support. Object orientation make them strong in design of WIS with complex business logic.

Nevertheless, we argue that the vagueness of specification of class methods (even with pre- and post-conditions the algorithms need to be designed separately) in object-oriented approach makes the automated generation of hypertext presentation difficult. The focus of Hera is in automating the presentation generation. Therefore, Hera exploits the simplicity of RMM for the application modeling. The adaptation is based on AHAM that is one of the most matured adaptive hypertext models. Hera supports both static and dynamic adaptation and belongs to modern approaches to automated hypermedia generation.

However, there is a number of issues that offer space for future development. Incorporation of more complex functionality (business logic) including good modeling of user-system interaction, (semi)automated building of AM (based on CM, and strategy models), and the problems of automated search of the data sources and their semantic integration with the CM represent a list of open issues.

References

[1] D. Brickley and R. V. Guha. Rdf vocabulary description language 1.0: Rdf schema. *W3C Working Draft 30 April 2002*.

[2] P. Brusilovski. Adaptive hypermedia. *User Modelling and User-Adapted Interaction*, 11(1-2):87–110, 2001.

[3] M. Cannataro and A. Pugliese. A flexible architecture for for adaptive hypermedia systems. In *IJCAI's Workshop on Intelligent Techniques for Web Personalization*, 2001.

[4] S. Ceri, P. Fraternalli, and A. Bongio. Web modeling language (webml): a modeling language for designing web sites. In *Computer Networks, Ninth International World Wide Web Conference*, volume 33, pages 137–157, 2000.

[5] P. De Bra, A. Aerts, and G. Houben. Making general-purpose adaptive hypermedia work. In *Proceedings of the WebNet World Conference on the WWW and Internet*, pages 117–123, AACE, 2000.

[6] P. De Bra, G. J. Houben, and H. Wu. Aham: A dexter-based reference model for adaprtive hypermedia. In *Proceedings of The Tenth ACM Conference on Hypertext and Hypermedia*, pages 147–156, 1999.

[7] F. Frasincar, G. J. Houben, and R. Vdovjak. Specification framework for engineering adaptive web applications. In *Proceedings of The Eleventh International World Wide Web Conference, Web Engineering Track*, 2002.

[8] F. Frasincar, G. J. Houben, and R. Vdovjak. An rmm-based methodology for hypermedia design. In *Proceedings of Advances in Databases and Information Systems*, volume 2151, pages 323–337. Lecture Notes in Computer Science, Springer, 2001.

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

[10] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. Rql: A declarative query language for rdf. In *Proceedings of The Eleventh International World Wide Web Conference*, ACM, 2002.

[11] N. Koch, A. Kraus, and R. Hennicker. The authoring process of the uml-based web engineering approach. In *Proceedings of The First International Workshop of Web-Oriented Software Technology*, 2001.

[12] N. Koch and M. Wirsing. The munich reference model for adaptive hypermedia applications. In *Proceedings of The Second International Conference on Adaptive Hypermedia Adaptive Web Based Systems*, volume 2347 of Lecture Notes in Computer Science, page 213, Springer, 2002.

[13] R. R. Lassila O., Swick. Resource description framework (rdf) model and syntax specification. *W3C Recommandation 22 February*, 1999.

[14] G. Rossi, D. Schwabe, and R. M. Guimares. Designing personalized web applications. In *Proceedings of The Tenth International World Wide Web conference*, pages 275–284, ACM, 2001.

[15] D. Schwabe, G. Rossi, and S. D. J. Barbosa. Systematic hypermedia application design with oohdm. In *Proceedings of The Seventh ACM Conference on Hypertext*, pages 116–128, 1996.