Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2000 Proceedings

Americas Conference on Information Systems (AMCIS)

2000

Towards Modeling of DataWeb Applications - A Requirement's Perspective

Werner Retschitzegger
Johannes Kepler University of Linz, werner@ifs.uni-linz.ac.at

Wieland Schwinger
Software Competence Center Hagenberg, wieland.schwinger@scch.at

Follow this and additional works at: http://aisel.aisnet.org/amcis2000

Recommended Citation

Retschitzegger, Werner and Schwinger, Wieland, "Towards Modeling of DataWeb Applications - A Requirement's Perspective" (2000). *AMCIS 2000 Proceedings*. 408.
http://aisel.aisnet.org/amcis2000/408

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Towards Modeling of DataWeb Applications - A Requirements' Perspective

Werner Retschitzegger, Department of Information Systems, University Linz, werner@ifs.uni-linz.ac.at
Wieland Schwinger, Software Competence Center Hagenberg, wieland.schwinger@scch.at

Abstract

The web is more and more used as a platform for full-fledged, increasingly complex information systems, where a huge amount of change-intensive data is managed by underlying database systems. From a software engineering point of view, the development of such so called DataWeb applications requires proper modeling methods in order to ensure architectural soundness and maintainability. The goal of this paper is twofold. First, a framework of requirements, covering the design space of DataWeb modeling methods in terms of three orthogonal dimensions is suggested. Second, on the basis of this framework, eight representative modeling methods for DataWeb applications are surveyed and general shortcomings are identified pointing the way to next-generation modeling methods.

Introduction

The Internet, and in particular the World Wide Web, have introduced a new era of computing, providing the basis for promising application areas like electronic commerce (Kappel et al., 1998). At the beginning, the web has been employed merely for simple read-only information systems, i.e., systems realized by some web server offering static web pages for browsing, only. Nowadays, the web is more and more used as a platform for full-fledged, increasingly complex information systems, where a huge amount of change-intensive data is (partly) managed by underlying database systems (Ehmayer et al., 1997). The data can be navigated through, queried, and updated by means of web browsers, whereby web pages may either be generated in advance or dynamically in response to the requests of users whose number and type is not necessarily predictable (Pröll et al., 1998; Pröll et al., 1999). This emerging kind of information systems is further on called *DataWeb applications*.

The development of such DataWeb applications is far from easy. Considering them from a software engineering point of view, as their complexity increases, so does the importance of modeling techniques. Models of a DataWeb application prior to its construction are essential for comprehension in its entirety, for communication among project teams, and to assure architectural soundness. However, the engineering of DataWeb applications has been widely neglected so far. This is not least since the unique characteristics of DataWeb applications comprising among others the usage of the hypermedia paradigm in terms of hypertext and multimedia in combination with traditional application logic make the straightforward employment of traditional modeling methods impossible (Nanard and Nanard, 1995; Powell, 1998).

The current situation can be characterized as follows. First, most current web application development practices rely on the *knowledge and experience of individual developers*. Second, *quick and dirty development* by means of various tools - if any - such as HTML editors, database publishing wizards, web site managers and web form editors, that are *driven by the underlying technology*, is the state of practice (Fraternali, 2000). Finally, and probably most important, up to now, the web has been considered simply as an information medium and consequently, web development is seen as an *authoring problem* only (Ginige et al., 1995). However, since the web evolves from a document-centric platform towards an application-centric platform, document authoring methods are no longer adequate.

In face of these problems, recently research towards modeling of DataWeb applications has been intensified. The goal of this paper is twofold. First, a framework of requirements, covering the design space of DataWeb modeling methods in terms of three orthogonal dimensions is suggested in Section 2. Second, on the basis of this framework, eight representative modeling methods for DataWeb applications are surveyed in Section 3. Finally, Section 4 concludes the paper by summarizing the key findings of our survey and points to future research.

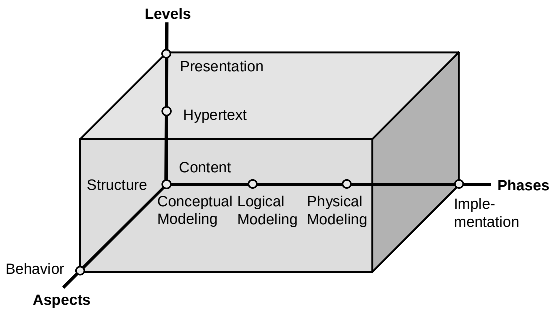
A Requirements' Framework for DataWeb Modeling Methods

In the following we want to elaborate on what is necessary when modeling DataWeb applications. The requirements discussed are partly derived from (Koch, 1999; Ceri et al., 1999a; Christodoulou et al., 1998) and (Fraternali, 2000). We have categorized these requirements by means of three orthogonal dimensions to be considered when modeling DataWeb applications, comprising *levels*, *aspects* and *phases* (cf. Figure 1). This framework of requirements allows to systematically survey DataWeb modeling methods thus indicating their strengths and shortcomings.

Levels: Content, Hypertext and Presentation

The first dimension of DataWeb application modeling comprises, similar to the Model/View/Controller (MVC) paradigm in object-oriented software development (Johnson and Foote, 1988), three different levels namely, the content level, the hypertext level, and the presentation level (Florescu et al., 1998). The *content level* refers to domain-dependent data used by the DataWeb application and is often managed by means of a database system. The *hypertext level* denotes the logical composition of web pages and the navigation structure. The *presentation level*, finally, is concerned with the representation of the hypertext level, e.g., the layout of each page and user interaction (Fernandez et al., 1997). Note that, the emphasize of each of these levels depends on the kind of DataWeb application which should be modeled as described later on.

Figure 1. Modeling Dimensions



Separation Between Levels and Explicit Mapping. A major requirement is that there should be a clear separation between these three levels, each one concerned with a distinct aspect of DataWeb applications. This can be achieved by making the interdependencies, i.e., the mapping between the levels explicit. This should facilitate model evolution and reuse, reduce complexity and enhance flexibility (Florescu et al., 1998; Rossi et al., 1999). For example, it would be possible to provide different presentations for the same hypertext level, depending on browser specifics or personalization issues.

Flexible Mapping Possibilities. In order to cope with the different goals intended when designing each of the levels, the possibilities for mapping should be as flexible as possible. For example, to make browsing more effective, documents are very redundant data-sources since the same piece of information can occur at several documents and navigated to by several different access paths. At the content level on the contrary, redundancy is eliminated by means of normalization techniques to avoid inconsistencies and update problems. Flexible mapping possibilities should ensure that despite of these differences, derivation of the levels from each other could be achieved. It would be also conceivable that the modeling method supports some kind of default mapping, which can be configured manually.

Bottom-Up and Top-Down Design. Another requirement concerning these levels is that modeling should not be limited to follow *bottom-up design*, i.e., to start with modeling the content level and then derive the other levels accordingly. Rather, it should be also allowed to adhere to *top-down design*, meaning that the content level is derived from the other levels (Fraternali and Paolini, 1998). Bottom-up design is needed when, e.g., the already existing content of a database should be brought to the web, whereas top-down design is useful in case that the content of already existing web pages should be stored within a database.

Aspects: Structure and Behavior

The second dimension comprises the aspects of *structure* and *behavior*, which are orthogonal to the three levels of the first dimension. Concerning the content level, besides structuring the domain by means of standard abstraction mechanisms such as classification, aggregation and generalization, the behavioral aspect in terms of domain-dependent application logic has to be considered too. Similarly, at the hypertext level, structure in terms of page compositions and navigational relationships in between as well as behavior like computing the endpoint of a certain link at runtime have to be modeled. At the presentation level, finally, user interface elements and their hierarchical composition have to be modeled concerning the structural aspect. The behavioral aspect comprises modeling of reactions to input events, e.g., pressing a certain button as well as interaction and synchronization between user interface elements. Note that similar to the levels discussed above, the amount of structure and behavior which has to be modeled depends on the kind of DataWeb application as described later.

Modeling Formalism for Structure and Behavior. A modeling formalism is required that takes into account both structural and behavioral particularities of the three levels. Although distinct modeling formalisms could be chosen for each of the three levels, for the purpose of a seamless mapping it would be beneficial if structure and behavior of all levels could be represented by building on a uniform basic modeling formalism. It has to be emphasized that this core modeling formalism has to be adapted in order to reflect specifics of each level. Since at all levels, both structure and behavior have to be modeled, it appears natural to build on an object-oriented modeling technique (Gellersen and Gaedke, 1999). This complies also with developments at the technology side, like the *Web Object Model* which has been suggested for the realization of DataWeb applications (Manola, 1999).

Patterns. Another requirement to facilitate reuse and abstraction of structure and behavior is that the modeling method should support the representation of design patterns at all levels. German et al. (German and Cowan, 2000) have reported on more than fifty design patterns, most of them concerning navigation at the hypertext level. Examples of navigational design patterns realizing contextual navigation, i.e., navigation from a given object to a related object in a certain semantic context, are *guided tours* which support linear navigation across pages, and *indexes*, allowing to navigate to the members of an index and vice versa (Ceri et al., 1999b).

Phases: Analysis, Logical Modeling, Physical Modeling and Implementation

The third dimension of modeling DataWeb applications comprises the different phases of a software life cycle, ranging from *analysis* to *implementation*. This dimension is orthogonal to the two previously presented ones, meaning that structure and behavior of content, navigation and presentation has to be addressed in each phase of the development process. At this time, there is no consensus on a general model for the lifecycle of DataWeb application development (Lowe and Webby, 1998). However, the influence of technological aspects tailoring the model towards the implementation environment, such as distribution, heterogeneity and database aspects, should certainly increase within the later phases of the modeling process. We therefore believe that, similar to database design, a separation between an abstract representation of the domain called *conceptual modeling*, technology independent design, i.e., *logical modeling*, and technology dependent design, i.e., *physical modeling* seems to be appropriate. Furthermore, in order to cope with the characteristics of aggressive release demands and rapid technology changes, web development should be much more incremental and iterative than development in other domains. That is, the need for prototyping and intensive testing with users is essential because user tolerance to errors in DataWeb applications is very low. A development process, which is part of an appropriate modeling method, has to take these requirements into account.

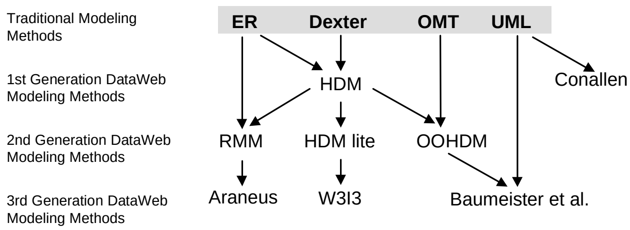
Emphasis of the Dimensions

Summarizing, modeling of any DataWeb application comprises these three dimensions, while their particular emphasis shifts for different kinds of DataWeb applications. For example, certain DataWeb applications provide a pure hypertext-oriented user interface to access large amounts of complex structured data. This might be realized as inter-linked HTML pages that are generated out of a database on a user's request by means of some server-side application logic. Examples for such kind of DataWeb applications can be found in the area of electronic commerce (cf., e.g., (Pröll et al., 1998; Pröll et al., 1999)) where the emphasis is on portability of the application across different browsers employed by Internet users, and where the underlying data changes frequently. Another kind of DataWeb application may require very complex application logic and interactivity at the client side. This could make it useful to resign the hypermedia paradigm in certain cases and instead employ Java applets for the user interface communicating directly with the database. Typical scenarios for this kind are Intranet applications, where delivering the code of the Java applet across the network does not affect performance. Of course, quite often a combination of these two kinds will be found in practice. Consequently, it is necessary that a modeling method takes into account these different peculiarities of DataWeb applications by providing appropriate concepts and modeling elements. The requirements' framework proposed is general enough to cover all these kinds of DataWeb modeling methods.

Evaluation of Existing Modeling Methods

On the basis of the requirements' framework given above, eight representative DataWeb modeling methods are surveyed in the following. Figure 2 illustrates the different origins of these methods, the arcs denoting influences between them. Accordingly, the modeling methods are categorized into different generations.

Figure 2. Origins of DataWeb Modeling Models



The modeling methods have their origins in different communities, including *database systems* being therefore mainly based on Entity-Relationship (ER) modeling (Chen, 1976) (cf. RMM (Isakowitz et al., 1998) and Araneus (Atzeni et al., 1998)), *hypermedia* using the Dexter Reference Model as basis (Halasz and Schwartz, 1994) (cf. HDM (Garzotto et al., 1995; Garzotto et al., 1997), HDM lite (Fraternali and Paolini, 1998; Fraternali, 2000) and W3I3 (Ceri et al., 1999b)) and *object-oriented modeling* in terms of the Object Modeling Technique (OMT) (Rumbaugh et al., 1991) and the Unified Modeling Language (UML) (Rumbaugh et al., 1998) (cf. OOHDM (Rossi et al., 1999), Baumeister et al. (Baumeister et al., 1999) and Conallen (Conallen, 1999)). Figure 3 summarizes the support of levels, aspects and phases concerning each of these approaches.

HDM. In modeling DataWeb applications, *HDM (Hypermedia Design Method)* (Garzotto et al., 1997) distinguishes between hypertext level and presentation level only, i.e., modeling of the application domain is intermingled with modeling the hypertext level. A reason could be that HDM originates from the hypermedia community, where the explicit modeling of the content level which is managed by databases is no issue. Although, a clear separation between hypertext level and presentation level is pursued by the authors, concepts for an explicit and flexible mapping are not described. The modeling formalism used is based on concepts borrowed from the ER Model (Chen, 1976) and from the Dexter Model (Halasz and Schwartz, 1994). Structural aspects are considered on both levels by means of various concepts. Behavioral aspects are mainly considered at the presentation level by modeling the user interaction in terms of (de)activation rules called *dynamics* in HDM. At the hypertext level, HDM further distinguishes between a so-called *hyperbase layer*, modeling the application domain and the *access layer*, defining a set of *collections* that provide users with the patterns to access the hyperbase such as index and guided tour. Finally, concerning the dimension of phases, HDM largely concentrates on two phases called *authoring in the large* and *authoring in the small*. Whereas authoring in the large comprises modeling of overall, general features, authoring in the small makes some refinements and takes the implementation technology into account.

RMM. *RMM (Relationship Management Methodology)* (Isakowitz et al., 1995; Isakowitz et al., 1998) is influenced by the ER Model and HDM. RMM recognizes all three levels. The content level is modeled separately whereas the presentation level is refined jointly with the hypertext level. A dedicated modeling formalism called *Relationship Management Data Model (RMDM)* is introduced, using the ER Model for the content level and proprietary concepts which are influenced by HDM for the hypertext level and the presentation level. The concept of so-called *m-slices* is used to map between the content level and the hypertext in that attributes from the entities of the ER-diagram and/or previously defined m-slices are grouped together. Navigational patterns in terms of index and guided tours are provided. Relationships between entities are used to capture contextual information during navigation. A so-called *Application Diagram* provides a global view of the presentation level of the DataWeb application by capturing all pages and hyperlinks in-between. Additionally, authoring tools are employed for creating page templates which in turn are assigned to every page. Only structural aspects are considered for all levels. RMM specifies a development process with initial steps for requirement analysis and content modeling in form of ER-diagrams. These are followed by iterative steps refining the Application Diagram both bottom-up, and top-down whilst m-slice design.

Araneus. *Araneus* (Atzeni et al., 1998), like RMM, embrace content level, hypertext level, and presentation level but emphasizes the content and hypertext level, only. A unique characteristic of Araneus is that content and hypertext level are refined independently from each other. Regarding the content level, based upon the *Conceptual Design*, the *Logical Design* and, if necessary, the *Physical Design* can be derived. Similar to RMM, the content level of Araneus relies on the ER Model. Considering the hypertext level, the *Hypertext Conceptual Design* formulated by the *Navigation Conceptual Model (NCM)* is refined by the *Hypertext Logical Design*, using *Araneus Data Model (ADM*) as formalism, tailoring the design towards the web. Likewise RMM, just structural aspects are considered for the content level as well as for the hypertext level. The presentation level is considered during *Presentation Design* relying on HTML-page templates created by an authoring tool. Patterns are not supported for any of the three levels. Araneus defines a process comprising initially the *Database Conceptual Design* from which in turn the *Hypertext Conceptual Design* is derived. After that, the refinement into the logical models is conducted in parallel. In the final step, after Presentation Design, the hypertext level is explicitly mapped onto the content level using a declarative formalism called *PENELOPE* building the basis for automatic page generation.

HDM lite. *HDM lite* (Fraternali, 2000) is a web-specific evolution of HDM condensing the concepts of HDM. Similar to HDM, the content level is not modeled separately but rather together with the hypertext level by means of the so-called *Structure Schema*. HDM lite uses a formalism descending from the ER Model and HDM. Additionally, at the hypertext level, the *Navigation Schema* specifies the access paths applying standard navigation patterns along with contextual navigation. Unlike HDM, the presentation level is modeled by means of the so-called *Presentation Schema* using a SGML like syntax as formalism. More than one presentation schema can be mapped to a Structure/Navigation Schema pair but no mapping constructs are supplied. Behavioral aspects are neglected for all three levels. Concerning the dimension of phases HDM lite proposes a transformation to convert the HDM lite conceptual schemata into a logical representation and further into a physical representation. For the former, well-known techniques for translating ER schemata into logical schemata augmented to treat also navigational and presentational issues are used. For the later non-standard transformation techniques are introduced. These transformations are implemented by the so-called *Autoweb system* thus automatically generating an *Application Data Schema*, a *Navigation Schema*, and a *Presentation Schema* covering content, hypertext, and presentation level, respectively. These logical schemata are further on utilized for automatic page generation by the Autoweb system.

W3I3. The main research objective of the EU Esprit project *W3I3 (Web-based Intelligent Information Infrastructure)* (Ceri et al., 1999b) is to rise the level of abstraction of the specification of a DataWeb application by enriching and refocusing the classical methods for database and hypertext design. W3I3 is an evolution of HDM lite and distinguishes five different models called *Structural Model*, *Derivation Model*, *Composition Model, Navigation Model,* and *Presentation Model.* The

Structural Model and the Derivation Model describe the content level by simply using an ER Model and derivation rules, respectively. The Composition Model describes, by means of *site views,* how the concepts of the Structural Level are mapped to web pages for a certain group of users and provides a default mapping for the case that there is only one simple site view needed. The Navigation Model describes the way in which associations within the Structural Model should be used for navigation thus capturing contextual navigation. Additionally, predefined navigational patterns are given for the hypertext level. The Presentation Model corresponding to the presentation level uses style sheets in order to define the layout of pages, whereby a default style sheet is provided for each page. Behavioral aspects are not considered at any of the three levels. Finally, W3I3 does not propose a particular process or specifies phases.

Figure 3. Comparison of Modeling Methods

	Levels														
	Content					Hypertext					Presentation				
	Aspects		*Phases*			*Aspects*		*Phases*			*Aspects*		*Phases*		
	S	*B*	*CM*	*LM*	*PM*	*S*	*B*	*CM*	*LM*	*PM*	*S*	*B*	*CM*	*LM*	*PM*
HDM	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✓	✗	✓	✓
RMM	✓	✗	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗	✓	✗	✓
Araneus	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✓
HDM lite	✓	✗	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗	✓	✓
W3I3	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✓
OOHDM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓
Baumeister	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Conallen	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✓

Legend: *Aspects*: S ....Structural Aspects; B ....Behavioral Aspects. *Phases*: CM ...... Conceptual Model; LM ...... Logical Model; PM ...... Physical Model. ✓ ......supported; ✗ ......not supported

OOHDM. *OOHDM (Object-oriented Hypermedia Design Method)* (Rossi et al., 1999) strictly separates the three levels of a DataWeb application. At the content level OOHDM uses the object-oriented modeling technique OMT (Rumbaugh et al., 1991) as a modeling formalism to capture structure and behavior. The hypertext level is modeled by means of three different concepts. First, the so-called *Navigational Class Schema* is used to define structural aspects by specifying the navigable classes of the application by means of an OMT class diagram. It can be seen as a view over the content level, whereby the mapping between these two levels can be done explicitly by means of a query language. Second, the *Navigational Context Schema* models the access structures to the navigable classes in terms of six different kinds of *contexts* by means of a proprietary notation thus capturing contextual navigation and providing an index navigation pattern. Third, *Navigational Transformation* specifications refer to the behavioral aspect of the hypertext level by modeling the activation/deactivation of navigational objects during navigation. There is no specific formalism employed but it is referred to statecharts only (Rumbaugh et al., 1991). At the presentation level a formalism called *Abstract Data View (ADV)* is used to describe the layout structure of navigational objects and other interface objects such as menu bars and buttons by means of traditional abstraction mechanisms. The behavioral aspect comprising the reactions to external events is described by using *ADV-Charts*, a derivative of statecharts. In order to express the mapping to navigational objects of the hypertext level in terms of static relationships so-called *Configuration Diagrams* are used. Finally, OOHDM does not suggest a dedicated process but distinguishes three different phases which partly correspond to the level's dimension, namely *conceptual modeling, navigational design* and *abstract interface design*.

Baumeister et al. The approach proposed by *Baumeister et al.* (Baumeister et al., 1999) is based on OOHDM but instead of using a mix of different formalisms throughout the levels, UML is used as the basic modeling technique. As far as necessary, UML is enhanced on the basis of two of UML's extension mechanisms namely *stereotypes*[1] and *constraints*[2]. It is separated between all three levels, comprising a *Conceptual Model* in terms of pure UML diagrams, a *Navigational Model* and a *Presentational Model.* At the hypertext level, the *Navigational Class Model* (cf. Navigational Class Schema in OOHDM) specifies which classes and associations of the content level are available for navigation. It is represented by means of a UML class diagram, denoting the navigational classes by means of stereotypes, navigable associations by means of directions and specifying the mapping by means of constraints. The *Navigational Structure Model* (cf. Navigational Context Schema in OOHDM) is based on the Navigational Class Model and defines (interestingly by means of an UML object diagram) how each navigational class is accessed during navigation. Stereotypes are again used to represent

[1] A *stereotype* represents an adornment that allows to define a new semantic meaning for a modeling element (Rumbaugh et al., 1998).

[2] *Constraints* are rules that define the well-formedness of a model and can be expressed as free-form text or with the more formal Object constraint language (OCL) (Rumbaugh et al., 1998).

navigational contexts and thus provide navigational patters as index and guided tour. Behavior modeling is only mentioned with respect to defining the sequence of navigation by means of constraints. At the presentational level, a *Static Presentation Model*, using the possibility of UML to represent compositions by means of graphical nesting describes the layout of the user interface, and a *Dynamic Presentation Model* employs UML statecharts for describing the activation of navigation and user interface transformations. Stereotypes representing the most frequently used interface objects such as text, image, audio and button are provided. Note that the mapping between hypertext level and presentation level is not discussed by the approach. Concerning the dimension of phases, the same holds as for OOHDM.

Conallen. The approach suggested by *Conallen* (Conallen, 1999) is completely different from the other ones, since it is to a great extent technology driven. UML is employed as the basic formalism and extended by means of stereotypes and *tagged values*[3]. Instead of distinguishing between content, hypertext and presentation level, Conallen models *web pages* at the *server side* and at the *client side* by stereotyping UML classes. Stereotyped associations are used to represent hyperlinks and to model the mapping between client pages and server pages, since every dynamic client web page, i.e., a page whose content is determined at runtime is constructed with a server page. Data entry forms which can be part of client pages together with their submit relationship to server pages are modeled by another class and association stereotype, respectively. Finally, there are also class stereotypes for Java Applets, Java Scripts, ActiveX controls and frames. Conallen does not discuss any behavior modeling apart from operations which can be defined together with the stereotyped classes and does not suggest any modeling phases.

Concluding Remarks

This paper has proposed a requirements' framework for DataWeb modeling methods. The requirements have been categorized by means of three dimensions which are orthogonal to each other. This requirements' framework was used to survey eight DataWeb modeling methods. The main shortcomings of these methods encountered during the evaluation can be summarized as follows:

- *Lack of Explicit and Flexible Mapping*. The definition of explicit and flexible mapping knowledge between the three levels is often not discussed by the approaches.
- *Top-Down and Bottom-Up Design is not Distinguished*. Most of the methods, except RMM, assume that modeling is done by starting either at the content level or at the hypertext level.
- *Behavioral Modeling is Widely Neglected*. Modeling the behavioral aspect of DataWeb applications at all levels is widely neglected by existing methods. If behavior is considered then mainly at the presentation level. Only those methods being based on object-oriented modeling formalisms partly deal with behavior modeling at all levels.
- *No Uniform Modeling Formalism*. Except the approaches of Baumeister et al. and Conallen which fully rely on UML, all modeling methods are based on a mix of mainly proprietary modeling formalisms.
- *Patterns are Supported at the Hypertext Level only*. There are no concepts provided to support the modeling of patterns at all three levels.
- *Presentation Level not Captured by Conceptual and Logical Modeling Concepts*. Most of the modeling methods do not support the presentation level with appropriate conceptual and logical modeling concepts. Rather, authoring tools are often suggested for capturing the presentation level, thus loosing the benefit of technology-independence.
- *No Process Support*. Most modeling methods do not follow a process for guiding the activities throughout the development of a DataWeb application.

In face of these various shortcomings, it can be argued that those modeling methods being based on the object-oriented paradigm and in particular on UML, seem to have the largest potential to cover all requirements of DataWeb application modeling.

Currently, we are working on an extension of UML towards DataWeb application modeling particularly addressing shortcomings identified in this paper.

References

Atzeni, P., Mecca, G. and Merialdo, P. "Design and Implementation of Data-Intensive Web Sites", *Proc. of the Conf. On Extended Database Technology (EDBT'98)*, Valencia, Spain, March 1998.

Baumeister, H., Koch N. and Mandel, L. "Towards a UML extension for hypermedia design", UML'99 The Unified Modeling Language - Beyond the Standard, LNCS 1723, Fort Collins, USA, Springer, October 1999.

Ceri, S., Fraternali, P. and Paraboschi, S. "Design Principles for Data-Intensive Web Sites", *SIGMOD Record*, 24(1), March 1999a.

Ceri, S., Fraternali, P. and Paraboschi, S. "Data-Driven One-To-One Web Site Generation for Data-Intensive Applications", *Proc. VLDB '99*, Edinburgh, Sept. 1999b.

[3] *Tagged values* are the third UML extension mechanism that allows to associate key value pairs with a modeling element (Rumbaugh et al., 1998).

Chen P. "The Entity-Relationship Model - Toward a Unified View of Data", *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, pp. 9-36.

Christodoulou, S., Styliaras, G. and Papatheodourou, T. "Evaluation of Hypermedia Application Development and Management Systems", *ACM Hypertext'98 Conference*, Pittsburgh, 1998, pp.1-10.

Conallen, J. "Modeling Web Application Architectures with UML", *CACM*, Vol. 42, No. 10, October 1999.

Ehmayer, G., Kappel, G. and Reich, S. "Connecting Databases to the Web - A Taxonomy of Gateways", *Proc. of the 8th Int. Conf. on Database and Expert Systems Applications (DEXA 97)*, France, Springer LNCS 1308, September 1997.

Fernandez, M., Florescu, D., Kang, J., Levy, A. and Suciu, D. "Strudel: A web-site management system", *Proc. ACM SIGMOD Int. Conference on Management of data*, Tucson, Arizona, May 1997.

Florescu, D., Levy, A. and Mendelzon, A. "Database Techniques for the World Wide Web: A Survey", *ACM SIGMOD Record*, Vol. 27, No. 3, Sept. 1998.

Fraternali, P. and Paolini, P. "A conceptual model and a tool environment for developing more scalable and dynamic Web applications", *Proc. of the Conference On Extended Database Technology (EDBT'98)*, Valencia, Spain, March 1998.

Fraternali, P. "Tools and approaches for data-intensive Web applications: A survey", to appear in *ACM Computing Surveys*, 2000.

Garzotto, F., Paolini, P. and Schwabe, D. "HDM - A Model-Based Approach to Hypertext Application Design", *ACM Transactions on Information Systems*, Vol. 11, No. 1, January 1995.

Garzotto, F., Mainetti, L. and Paolini, P. "Designing Modal Hypermedia Applications", *Proc. of the 8th ACM Conference on Hypertext*, University of Southampton, UK, April, 1997.

Gellersen, H.-W. and Gaedke, M. "Object-Oriented Web Application Development", *IEEE Internet Computing*, 3(1), Jan.-Feb. 1999.

German, D. and Cowan, D. "Towards a Unified Catalog of Hypermedia Design Patterns", *33rd Hawaii International Conference on System Sciences (HICSS 2000)*, Maui, Hawaii, January, 2000.

Ginige, A., Lowe, D.B. and Robertson, J. "Hypermedia Authoring", *IEEE Multimedia*, Vol.2, No.4, 1995.

Isakowitz, T., Stohr, E.A. and Balasubramanian, P. "RMM: A Methodology for Structured Hypermedia Design", *CACM*, Vol. 38, No. 8, Aug 1995, pp. 34-44.

Johnson, R.E. and Foote, B. "Designing Reusable Classes", *Journal of Object-Oriented Programming (JOOP)*, 1(2), June/July 1988.

Kappel, G., Retschitzegger, W. and Schröder, B. "Enabling Technologies for Electronic Commerce", *Proc. of the XV. IFIP World Computer Congress*, Vienna/Austria and Budapest/Hungary, Aug./Sept. 1998.

Koch, N. "A Comparative Study of Methods for Hypermedia Development", Technical Report 9905, Ludwig-Maximilians-Universität München, Nov. 1999.

Lowe, D. and Webby R. "Web Development Process Modelling and Project Scoping: Work in Progress", *Workshop on Web Engineering, WWW7 Conference*, Brisbane, 1998.

Manola, F. "Technologies for a Web Object Model", *IEEE Internet Computing*, 3(1), Jan.-Feb. 1999.

Nanard, J. and Nanard, M. "Hypertext Design Environments and Hypertext Design Process". *CACM*, Vol. 38, No. 8, August 1995.

Powell, T. "Web Site Engineering", Prentice Hall, 1998.

Pröll, B., Retschitzegger, W., Wagner, R.R. and Ebner, A. "Beyond Traditional Tourism Information Systems - TIScover", *Journal of Information Technology and Tourism*, Vol. 1, Inaugural Volume, 1998, pp. 15-31.

Pröll, B., Retschitzegger, W., Sighart, H. and Starck, H. "Ready for Prime Time – Pre-Generation of Web-Pages in TIScover", *Proc. of the ACM Conf. On Information and Knowledge Management (CIKM'99)*, Kansas City, Missouri, November 1999.

Rossi G., Schwabe, D. and Garrido, A. "Designing Computational Hypermedia Applications", *Journal of Digital Information (JODI)*, 1(4), February 1999.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F and Lorenson, W. "Object-Oriented Modeling and Design", Prentice-Hall, 1991.

Rumbaugh, J., Jacobson I. and Booch, G. "The Unified Modeling Language Reference Manual", Addison-Wesley, 1998.