An Integrated Model for the authoring of Web-based Adaptive Educational Applications

Andreas Papasalouros[1], Simos Retalis[2], Paris Avgeriou[2], Manolis Skordalakis[1]

[1]Software Engineering Laboratory
National Technical University of Athens
15780 Zografou, Athens, Greece
Tel. +302 10 7722486, Fax. +302 10 7722519
{andpapas,skordala}@softlab.ntua.gr

[2]Computer Science Department
University of Cyprus
P.O. Box 20537 Nicosia, Cyprus
Tel: +357 22 892246, Fax: +357 22 339062
{retal,pavger}@cs.ucy.ac.cy

Abstract

In this paper a design model for Web-based Adaptive Educational Applications (WAEA) is presented. A model-based approach is proposed as an answering to the problem of the difficulty of authoring such applications. This approach is based on the use of object oriented modeling techniques and the specification of WAEA by means of an XML binding.

1. Introduction

The authoring of Web–based Adaptive Educational Applications (WAEA) is a complex task. People with different background are involved in this kind of development, such as software developers, web application experts, content developers, domain experts, instructional designers, etc. Furthermore, the WAEA are complex dynamic web-based applications with presentational, behavioral and architectural aspects. In order to effectively capture and specify the various aspects of a WAEA, to document the decisions concerning such applications, from the implementation of pedagogic and instructional design to subtle technical decisions of such applications and to facilitate the communication between the members of usually heterogeneous development teams there is a need for a *design model*. This model will incorporate an engineering approach [8] in the authoring of WAEA. Experience from traditional software engineering has shown that the adoption of such a model is beneficial for the quality of the software products and the efficient management of the resources, time and effort. Although this model can be used as a Reference Model, that provides common understanding and communications of the various components of a WAEA, its main purpose is to facilitate the process of development of Adaptive Educational Applications. A design model like this can be used as a framework [9] for authors of hypertext applications to develop and apply methodologies in order to create adaptive applications in a disciplined and controlled fashion. It incorporates the principle of separation of concerns in the design of hypermedia applications, dividing the design of the application in three stages: conceptual, navigational and presentational. We also claim that this separation of concerns aligns with the three types of adaptation, navigation and presentation. Beyond a design model, if the development of open, portable, maintainable WAEA is to be facilitated, there is a need for a formally specified description of the WAEA. This description must be automatically generated from the aforementioned design model, at least to an extend, and must be easily ported to specific run-time environments that will deliver the specific WAEA.

In this paper, we define a Web based Adaptive Educational Application (WAEA) as a dynamic web application, i.e. a set of dynamically generated web content, which provides a learning environment to its users. This environment comprises of electronic content for study as well as a set of tools that facilitate the study of a learner such as web – based questionnaires, glossaries, communication tools, etc. This model focuses on content, which is considered as hierarchically structured, usually dynamically created, personalized assembly of predefined learning resources, either created from scratch or reused. These resources can be available in any form such as files, database entries, etc. We propose a model for the design and the construction of this kind of learning content. This model is considered to have two equivalent views: One view consists of diagrams in the UML notation language [19], which facilitates the design of the application in an intuitive, human understandable manner. The other consists of the description of the information that describes the WAEA in a formal, machine consumable language, by means of an XML binding.

The structure of the paper is as follows: In Section 2 we provide the description of our model. In Section 3 an exemplar application is given while in Section 4 follows an analysis of existing approaches in adaptive application modeling. In Section 5 we evaluate our approach and describe feature work based on the proposed model.

2. The Model of a Web based Adaptive Educational Application

In this section the first component of our approach, the design model, is described. The model is based on the Unified Modeling Language. It is an extension of the UML, formulated using the standard extension mechanisms of the language, i.e. Stereotypes, and specified by means of a UML Profile, 'a coherent set of such extensions, defined for specific purposes' [18].

The decomposition of this model into sub-models is based upon and extends the AHAM reference model for Adaptive Hypermedia Applications [2]. The following sub-models comprise an extended set of the models defined in AHAM:

2.1 The Conceptual Model

The *Conceptual Model* defines the concepts of the subject that is going to be taught with their semantic interrelationships. It can be considered as the Ontology [22] of the subject to be learned by the students. The Conceptual Model provides an objective definition of the knowledge subject. This definition is provided by the author of the educational application who is considered as a subject matter expert. The actual hypertextual content delivered to learners comprises a personalized, dynamic view over this conceptual realm. The main entity of the conceptual model is the C*oncept*, which depicts a main idea or topic of interest into the educational application. Concepts are abstract entities that do not carry actual content by themselves. They can contain meta-data or other descriptions, but the actual content is defined in the associated Resources. The *Resources* are the actual fragments of content that compose the WAEA, text, images, sounds, videos, simulations, forms, etc, which are static, reusable components or dynamic components such as multiple choice questions or glossary terms, dynamically created and delivered by appropriate web-based tools whose operation is not specified by our model. These tools can sometimes be considered as resources themselves.

Note that the granularity of a resource can vary from the content of a whole chapter to a single picture or paragraph of text. Two (or more) concepts can be associated with *Relationships*, which capture the semantic links between these concepts. Both concepts and relationships in the Conceptual Model are described as attribute-value pairs. The elements of the Conceptual Model are unchangeable during in the whole life time of the WAEA.

2.2 The Navigation Model

The Navigation Model captures the decisions about how Concepts, Relationships and Resources of the Conceptual Model are mapped to actual hypertext elements Pages and Links, and how the conceptual relationships defined in the Conceptual Model are driving the structuring of the learning content. The *Navigation Model* is composed by two sub-models:

2.2.1 The Navigation Structure Model

This model defines the structure of the WAEA and defines the actual web pages and the resources contained in these pages.

This structure is composed of the following elements:

- *Content,* which is the top-level container in the hierarchy of an electronic content organization.
- C*omposite* entities that are used as containers, thus composing the hierarchical structure of learning content. The chapters and subtopics in which an electronic tutorial or book are organized are examples of composite entities.
- *Access structures* elements, namely *indexes* and *guided tours,* which are related to Content or Composite components
- *ContentNodes*, which are the actual pages of the learning content. Content, Composite and ContentNodes are associated with Concept elements, or directly with Resources, in the Conceptual Model.
- *Fragments* that are contained into the ContentNodes. Fragments correspond to Resource elements in the Conceptual Model.
- *Links* between ContentNodes as well as between Fragments. Note that these links are associative links [9, 17] implementing domain specific relationships of the conceptual model. They are not structural links denoting, for example, the transition from a page in the learning content to the next one.
- Composite, ContentNodes, Fragments and Links have a predefined attribute of Boolean type named *included*. This denotes whether or not a specific element (and all its descendants in the hierarchy) is included in the created hypertext or not, as a result of adaptation.

2.2.2 The Navigation Behavior Model

The *Navigation Behavior Model* defines the run-time behavior of the WAEA. Earlier research [6, 14, 24] has proposed the use of statecharts for the modeling of hypertext and web based applications. The Navigation Behavior model uses statecharts, as they are incorporated in the UML in order to specify the dynamic transitions of the hypertext structures as the user interacts with the WAEA. Every containing element of the Navigation Structure Model (Content, or Composite) is associated to a composite state in the Navigation Behavior Model, while every ContentNode corresponds to a simple state. Thus, the hierarchy of the navigational elements defined in the Navigation Structure Model corresponds to the hierarchy of nested states in the Navigation Behavior Model. The events that fire the transitions in the Navigation Behavior Model correspond to structure links into the ContentNodes: next, previous, up level, etc. In addition, guard conditions in these transitions can define alternative navigational transitions, which correspond to conditional behavior of the WAEA, thus implementing content sequencing and adaptive navigation.

2.3 The Presentation Model

The *Presentation Model* deals with the presentation aspects of the elements defined in the Navigation Model.

The presentation model is by itself separated in two additional sub-models: *Presentation Structure* Model, which defines the allocation of the navigational elements to actual user interface web elements: Web pages, frames, framesets, etc. Elements of this model, which is a variation of the synonymous model proposed in [14], are the following: *frameset*, *frame*, *window*. The aforementioned elements are associated with one or more elements of the Navigation Model.

User Interface Model, that captures the layout, colors, styles, etc of the entire web pages as well of atomic elements of the pages. This model consists of *Presentation* elements, which define the layout and style of associated elements of the navigation model.

2.4 The User Model

The User Model consists of two different parts, each one containing two types of elements: The *Overlay Model*, which is the domain specific part of the user model and defines the status of the learner's knowledge of the specific concepts covered by the learning material. The state of this model is frequently updated as a result of the learner's interaction with the learning content, for example the reading of learning material, the taking of an on-line test, the interaction with simulations, etc. The knowledge is defined as a structure of concepts (schema) and this structure is built during the user's learning activities. The Overlay Model depicts the system's awareness of the current status of the user's knowledge about the domain of the specific application as it is stated in the Conceptual Model. The elements of this sub-model are called *UserScheme* [2], and there can be one *UserScheme* element for each class of the Conceptual Model.

The second part of the User Model defines elements that are used to represent the usually predefined user knowledge profile either concerning the knowledge of the particular domain (novice, intermediate, expert, etc) or corresponding to the user's preferences or learning style. According to [4] this constitutes the Stereotyped User Model. The elements of this submodel are called *User*.

2.5 The Rules Model

The adaptive behavior of the application is specified with appropriate rules. The rules are applied in two ways:

- As Object Constraint Language Expressions (OCL) [23]. OCL is a formal language for applying constraints to UML models. Constraints are conditions that must hold for a specific model they are applied. The rules defined in the Rules Model are applied as two types of constraints:

- *Invariants*, that is conditions that must *always* be true in the context they are applied (concept components, concept relationships).
- *Postconditions*, that is conditions that must be met *after* the execution of a method or operation of a specific class.

The constraints are applied to specific model elements, defined by the keyword **context,** as will be shown in the following examples.

- As *guard-conditions* in the transitions defined in the Navigation Behavior Model.

OCL rules can be applied to elements of every one of the aforementioned models. For example, a rule in the User Model provides a mechanism for updating the knowledge of the user on a particular concept as a result to his/her navigation.

2.6 The XML definition

For this purpose we developed an XML binding for the model described above. The XML bindings are defined as a Document Type Declaration (DTD). The DTD definition was preferred to XML Schema [http://www.w3c.org/XML/Schema], as the former is supported by a much wider range of current XML tools such as validators, XSL processors, etc. In the following the listing of the whole DTD is omitted due to space restrictions.

Each of the elements in the previous sections is defined as an element in the DTD. For example, the root element, i.e. *waea*, is defined as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT waea (ConceptualModel, NavigationModel,
   PresentationModel, UserModel, RulesModel)>
```

Each element has a unique identifier through which it can be referred by its associated other elements. For example

```
<!ELEMENT Resource ANY>
<!ATTLIST Resource
   id ID #REQUIRED
   mime-type CDATA #IMPLIED
   uri CDATA #IMPLIED
>
```

Certain elements have arbitrary sets of attributes. The attributes are defined as follows:

```
<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute
   name CDATA #REQUIRED
   value CDATA #REQUIRED
   type CDATA #REQUIRED
>
```

3. An Example Application

We provide an example application in order to demonstrate the use of our model in a simple WAEA This example is not aiming to depict how a whole educational application can be modeled and authored according to the proposed approach, but to illustrate the elements and their use in constructing a model for an WAEA.

Figure 1 shows the conceptual model of an application for the teaching of Java Swing Basics. Java Swing is a framework for the development of user interfaces in the Java programming language. It shows the basic concepts that are going to be taught together with their semantic interrelationships. In order to keep the diagram simple, not all the resources associated with corresponding Concepts are depicted. Note that the model primitive elements (Concept, Resource, etc) are shown as stereotypes of UML classes.

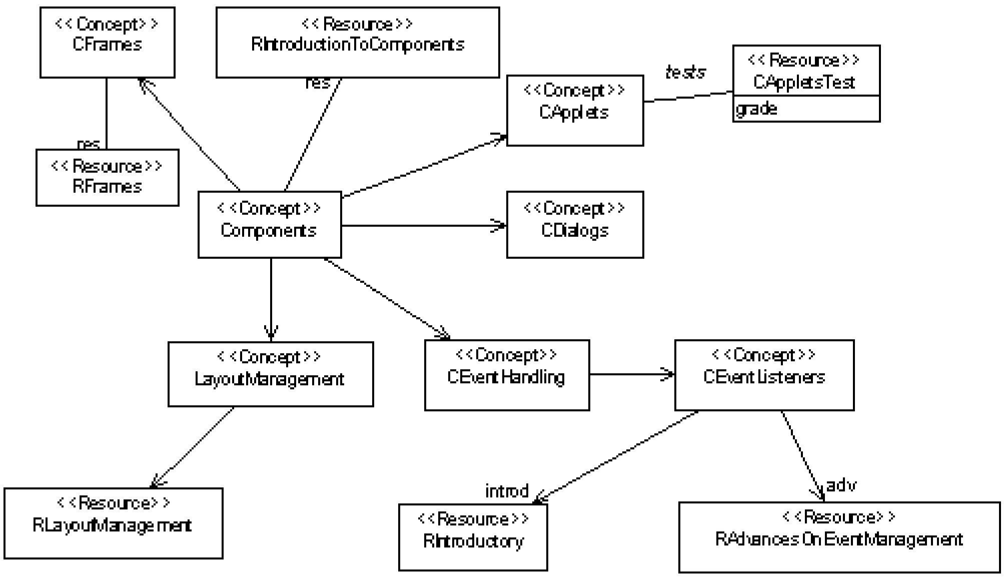

Fig. 1. Example of Conceptual Model

Figure2 displays the Navigation Model. Each element of this model is associated with corresponding elements of the Conceptual Model, though these UML associations are not displayed for the sake of clarity.

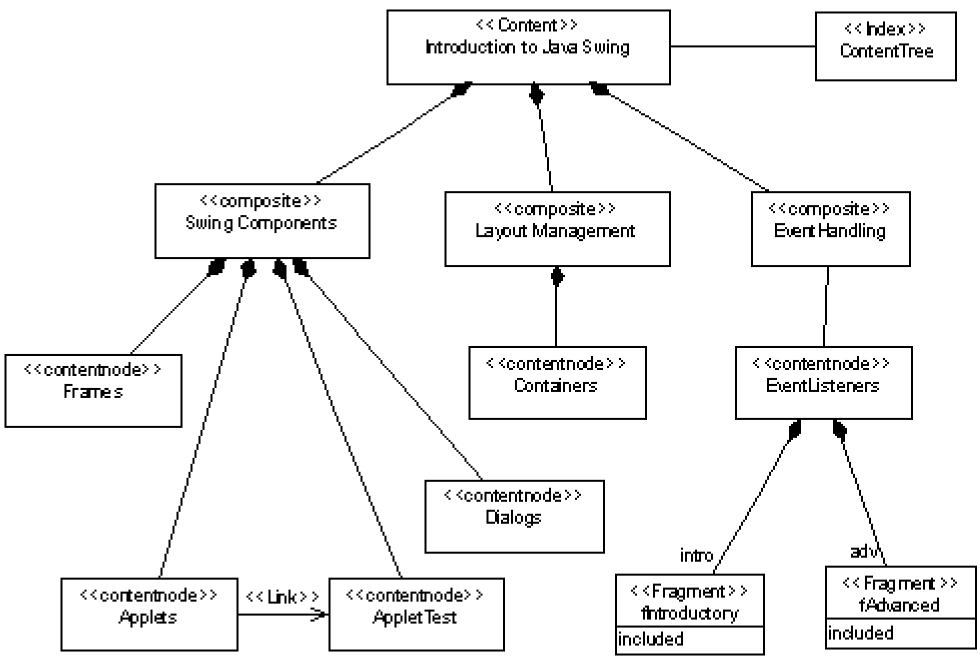

Fig. 2. Example of Navigational Structure Model

In Figure 3 the Navigation Behavior model for the application is shown. Note that the hierarchy of nested states in this example corresponds to the hierarchy of Composite and ContentNode elements in Figure 2.

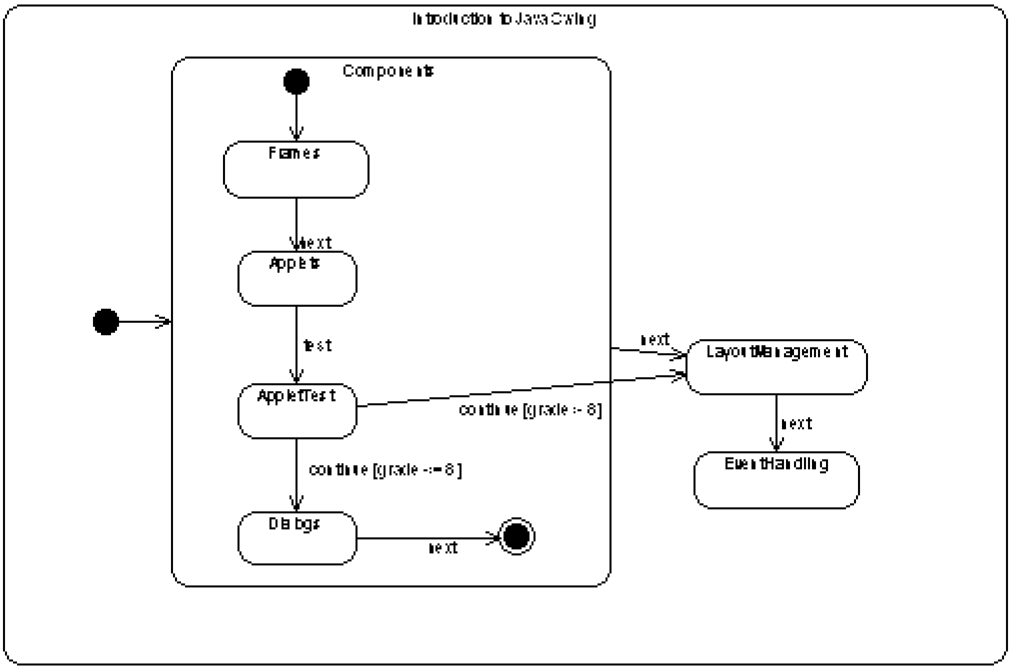

Fig. 3. Example of Navigation Behavior Model

The arcs in Figure 3 denote state transitions that correspond to link traverse in the navigational space of the applications. As shown in the same figure, when the user follows the *next* link while he or she was in the Applets page, the AppletTest page is presented to the user. The user takes the test and if the grade is high enough, the Dialogs section is omitted and the system presents the user the LayoutManagement section, as he follows the *continue* link. Note that in guard condition

[grade > 8]

grade is an attribute of the AppletTest class of the Navigation Model (see Figure 2).

As an example of content adaptation, according to the User Profile, we define a very simple user model as a User stereotyped class. This user class has an attribute named 'type'. We assume that there are two types of users, namely 'Novice' and 'Advanced'. We want to include different content (resources) related to the EventListeners concept, dependent on the type of the user. In the first case the RIntroductory resource is included, while in the second case the RadvancesInEventManagement is included (see Figure 2). The fragments correspond to the two available resources, namely fIntroductory and fAdvanced. The control over which of the two resources is finally displayed is through the *included* attribute applying an appropriate OCL expression:

```
context EventListeners
inv:    (user.type = 'Novice' implies (intro.included = true and
                adv.included = false))
        and
        (user.type = 'Advanced' implies (intro.included = false and
                adv.included = true))
```

where ‘user’ is the user type associated with the ‘EventListeners’ navigational class. These associations, which constitute the UML diagrams syntactically correct, are omitted for the sake of clarity. Note that the keyword **implies** means that when the expression is evaluated then if the condition left to the **implies** keyword is true then the condition to the right must also be true, in order for the whole expression to be true.

4. Related work

Up to our knowledge, the following approaches exist in modeling of Adaptive Hypermedia Applications, in general. In these approaches we have included the information models proposed by some learning technology standards specifications, namely the IMS Learning Design and Simple Sequencing Standards.

The meta–model defined in [7] provides a four layer architecture for the definition of hypertext applications, with provision for user modeling. This approach for modeling of hypermedia applications utilizes adaptive presentation and engages a variety of formal mechanisms and visualization techniques. It cannot be easily applied in different contexts, due to its tight relationship with specific systems, like the ConceptBase system, while the diversity of applied formalisms is an obstacle in applying it in general. Although it is domain independent it cannot be used for educational applications, because it does not address issues like adaptive sequencing, adaptive structuring of the learning content, etc which are important in this kind of applications.

AHAM [2] is a reference model for Adaptive Hypermedia Applications. Our modeling approach is based on primitives of the AHAM using them in the direction of a Design Model for a specific category of Adaptive Applications, i.e. Educational ones. AHA [1] is a system based on AHAM. The data model for the description of the Adaptive application is much similar to the one we propose. The main difference in our approach is the separation of concepts from resources, navigation and presentation, which gives the ability for conditional text, or other type of resource, inclusion, conditional page creation and presentation.

In [15] a software engineering approach based on UML (UWE) is proposed in order to facilitate the developing Adaptive Web based applications. In this approach specific views of an Adaptive Web Applications are defined, expressed as different models. This approach is very similar to ours in the separation of concerns in the hypertext design defining different models, the use of the UML and the use of OCL for the applications of rules. However, it is not appropriate for the domain of educational applications since it focuses on the development of web based systems with adaptive features which present highly structured content. On the contrary, educational applications are semi-structured, in the sense that they lay between highly structured systems, or applications, like hypermedia front ends for library systems or electronic market places, and unstructured applications that derive as a result of a purely creative artistic task, like electronic presentation of novels and literary work [11].

In [21] a meta-model for courseware design is defined. According to this abstract meta-model, specific models can be derived for certain domains of teaching, or disciplines. It resembles our approach in that it has both a UML design and an XML part. However it cannot be considered as an adaptive educational hypermedia model, since it does not provide a specific user model neither proposes a specific formalism for the definition of rules for adaptation.

The IMS Learning Design (IMS LD) Standard defined in [13] is not a hypermedia design model, but rather a framework for formally specifying educational activities in the context of a learning system that incorporates traditional learning methods with learning technologies. Not being a hypermedia model, IMS LD does not cope with the details of structuring of the hypertext, hyperlinks, presentation, etc, but rather focuses on the dynamic aspects of electronic content delivery viewing them as the implementation of pedagogic practices and specific learning design. It proposes an XML schema and does not cope with facilitating the design of Educational Applications.

IMS Simple Sequencing (IMS SS) [12] is a standard proposed by the IMS consortium that provides the basis for sequencing of Learning Activities in the context of a Learning Technology System according to specific rules. IMS SS deals with the sequencing issues of predefined, pre-structured learning content. It does not deal with adaptive content or adaptive presentation of the hypermedia content, but only with adaptive navigation, in the sense of the automatic selection of next resource to be presented to a learner according to her/his history of interaction with the learning material.

In [5] a layer approach for the modeling of Adaptive Educational Applications is provided, together with a method for the design of such applications. This approach is similar to ours in the distinction of three views of Adaptive Educational Application depicted as layers: A conceptual Layer, a lesson layer and a student adaptation and presentation layers, which resemble our separation in three sub-models, i.e. conceptual, navigational and presentational. A second main similarity is that both approaches recognize that the authoring of WAEA is driven by an initial mapping of the available resources in a high level conceptual model. The main differences from this approach are in the way of mapping of the initially defined concepts into specific navigation and presentation elements, as well as the specific formalism used in our approach, namely UML.

In [20] we define an Object-oriented Model for Adaptive Hypermedia Educational Applications. This model is an extension of AHAM for the modeling of adaptive educational hypermedia using UML and object-oriented principles.

5. Conclusions –Future Work

A number of solutions exist for modeling and representation of Web-based Adaptive Hypermedia for Educational purposes. The one presented here combines many of the features from the above in the aim of providing the means for an adequate description of WAEA and the facilitation of the author of such applications for a disciplined and effective application authoring. This approach is adequate in that it can describe various types of adaptation, useful in educational applications: Conditional content inclusion, sequencing of content according to the user’s interaction, adaptation of the user interface. It provides a design model for WAEA that facilitates the process of authoring and maintaining of such applications through a consistent visual formalism facilitated with the use of UML which is a widely adopted modeling language.

Future work according to this model includes the development of a case tool that will facilitate the process of model authoring and the automatic production of XML-based descriptions of WAEA, the specification of the run-time system that complies with the proposed model and the implementation of an instance of such a system for further testing of the model. Also the compatibility of this model with the existing learning standards described above is to be investigated.

References

1. P. De Bra and L. Calvi. AHA! An open Adaptive Hypermedia Architecture, *The New Review of Hypermedia and Multimedia*, vol. 4, pp 106-112 (2000).
2. P. De Bra, G.J. Houben and H. Wu. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and Hypermedia*, Darmstadt, Germany, February 1999. ACM Press.
3. Peter Brusilovsky. Adaptive educational systems on the world-wide-web: A review of available technologies. In *Proceedings of Workshop WWW-Based Tutoring,* 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, 1998.
4. Brusilovsky. Methods and Techniqes of adaptive hypermedia. *User Modeling and User Adapted Interaction*, vol. 6, pp 87-129, 1996.
5. A. Cristea and L. Aroyo. Adaptive Authoring of Adaptive Educational Hypermedia. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web System*, Malaga, Spain, May 2002.
6. M. C. Ferreira de Oliveira , M. A. Santos Turine and P. C. Masiero. A Statechart-Based Model for Hypermedia Applications, *ACM Transactions on Information Systems*, Vol. 19, Issue 1, 2001.
7. P. Fröhlich, N. Henze, and W. Nejdl. Meta Modeling for Hypermedia Design. In *Proceedings of IEEE Metadata Conference,* Maryland, USA, Sept. 1997.
8. D. Garlan and M. Shaw. Software Architecture: Perspectives of an Emerging Discipline. Prentice Hall, New Jersey, 1996.
9. F. Garzotto, D. Schwabe and P. Paolini. HDM- A Model Based Approach to Hypermedia Application Design. *ACM Transactions on Information Systems*, vol. 11, n. 1, pp. 1-26, 1993.
10. F. Halasz and M. Schwartz. The Dexter Reference Model. *Communications of the ACM*, vol. 37, n. 2, pp. 30-39, 1994.
11. T. Isakowitz, E. Stohr and P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, vol. 38, n. 4, 1995.
12. IMS Global Learning Consortium Inc. *IMS Simple Sequencing Information and Behavior Model*. Available at http://www.imsproject.org/simplesequencing/
13. IMS Global Learning Consortium Inc. *IMS Learning Design Information Model*. Available at http://www.imsproject.org/learningdesign.
14. N. Koch and A. Kraus. The Expressive Power of UML-based Web Engineering. In *Proceedings of the 2nd International Conference Workshop on Web Oriented Software Technologies*, Malaga, Spain, June 2002.
15. N. P. de Koch. Software Engineering for Adaptive Hypermedia Systems, PhD Thesis, Verlag Uni-Druck, Munich 2001.
16. Rod Koper. Modeling units of study from a Pedagogical Perspective: the pedagogical meta-model behind EML. Available at http://eml.ou.nl.
17. D. Lowe and W. Hall. *Hypermedia & the Web, an Engineering Approach*, John Wiley & Sons, 1999.
18. OMG. UML Profile for CORBA Specification. Available at http://www.omg.org/uml.
19. OMG. Unified Modeling Language Specification. Available at http://www.omg.org/uml. September 2001.
20. A. Papasalouros and S. Retalis. Ob-AHEM: A UML-enabled model for Adaptive Educational Hypermedia Applications. *Interactive educational Multimedia*, vol. 4, 2002.

21. C. Süß, B. Freitag and P. Brössler. Meta-Modeling for Web-Based Teachware Management. In *Advances in Conceptual Modeling*, ER'99 Workshop on the World-Wide Web and Conceptual Modeling , Paris, France 1999. Springer.
22. M. Uschold and M. Grunninger. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, vol.1, n. 2, pp. 93-155, 1996.
23. J. Warmer and A. Kleppe. *The Object Constraint Language: Precise Modeling with UML.* Addison Wesley, Reading Mass., 1999.
24. Y. Zheng and M. Pong. Using statecharts to model hypertext. In *Proceedings of the ACM conference on Hypertext,* Milan, Italy, 1992, ACM Press.