Retalis, R., & Papasalouros, A. (2005). Designing and Generating Educational Adaptive Hypermedia Applications. *Educational Technology & Society*, 8 (3), 26-35.

Designing and Generating Educational Adaptive Hypermedia Applications

Symeon Retalis
University of Piraeus
Department of Technology Education and Digital Systems
80 Karaoli & Dimitriou 185 34 Piraeus, Greece
retal@unipi.gr

Andreas Papasalouros
National Technical University of Athens
Software Engineering Laboratory
9 Heroon Polytechneioy 157 80 Zografou, Greece
andpapas@softlab.ntua.gr

ABSTRACT

Educational Adaptive Hypermedia Applications (EAHA) provide personalized views on the learning content to individual learners. They also offer adaptive sequencing (navigation) over the learning content based on rules that stem from the user model requirements and the instructional strategies. EAHA are gaining the focus of the research community as a means of alleviating a number of user problems related to hypermedia. However, the difficulty and complexity of developing such applications and systems have been identified as possible reasons for the low diffusion of Adaptive Hypermedia in web-based education. Experience from traditional Software Engineering as well as Hypermedia Engineering suggests that a model-driven design approach is appropriate for developing applications where such requirements and constraints occur. This paper presents on a model-driven design process of EAHA. This process accords to the principles of hypermedia engineering and its innovation is the use of a formally specified object oriented design model.

Keywords

Hypermedia design, Adaptive Hypermedia, Application generation, Design method.

Introduction

An Educational Adaptive Hypermedia Application (EAHA) is a dynamic web-based application, which provides a tailored learning environment to its users, by adapting both the presentation and the navigation through the learning content. Such an application is comprised of learning resources that have specific learning objectives and they are interrelated in order to facilitate the learning process. The learning resources are designed based on pedagogical rules (or teaching rules) that combine the (domain) model of the content with the user model and the instructional strategies. In this paper, the term 'adaptive' pertains to the conditional change of the sequence of the presentation of learning resources during learning activity based on the previous interaction of the learner with the application, as well as with the predefined learner type.

EAHAs are currently a research topic of particular interest in the broader field of adaptive hypermedia applications and several EAHA systems have been built during the past years (De Bra et al., 1999; Brusilovsky, 1996; Brusilovsky, 1999; Fischer, 2001). The design and implementation of EAHA are complex, if not overwhelming, tasks. This is due to the fact that it involves people from diverse backgrounds, such as software developers, web application experts, content developers, domain experts, instructional designers, user modeling experts and pedagogues, to name just a few. Moreover, these systems have presentational, behavioral, pedagogical and architectural aspects that need to be taken into account. To make matters worse, most EAHA are designed and developed from scratch, without taking advantage of the experience from previously developed applications, because the latter's design is not codified or documented. As a result, development teams are forced to 're-invent the wheel'.

Therefore, systematic and disciplined approaches must be devised in order to overcome the complexity and assortment of EAHA and achieve overall product quality within specific time and budget limits. One such approach is the use of a systematic design method to support the whole design process.

Two candidate approaches exist in this direction, software engineering and hypermedia engineering design methods. Software Engineering methods fail to deal with the particular requirements of hypermedia applications, their user interface intensive nature and their complex node-and-link structure. Although the discipline of Hypermedia Engineering (Garzotto et al., 1993; Hennicker & Koch, 2001; Rossi et al., 1995; Schwabe & Rossi,

1995) emerged to address this issue, existing Hypermedia Engineering methods are not adequate for properly dealing with the design of educational hypermedia applications. Since educational applications deal with learning, the specification of such applications is a planned set of carefully designed activities and tasks, assessment procedures, selection of proper resources that will support these activities and procedures, that is, the outcome of instructional analysis. According to (Lowe & Hall, 1999), in the design phase, the specification of a hypermedia application is converted into a description of how to create the application. Existing hypermedia engineering methods do not provide adequate constructs for capturing this specification since they pertain to applications that provide views over highly structured data and front ends for transactions resulting to consistent changes of these data. Thus, modeling elements suggested by these methods for capturing the domain concepts are abstractions and representations this kind of data usually stored in databases. This is the reason why these modeling primitives typically derive from entity-relationship and object-oriented modeling approaches. However, these primitives do not integrate well with the semi-structured nature of the instruction solutions met in educational applications. Moreover, this pure integration makes it difficult to assign to modeling structures of existing methods proper semantics, which is crucial if it is desired for models to drive the generation of actual applications. Furthermore, although navigation structures proposed by existing methods are generic enough to be used in the domain of educational applications, a more specialized approach would be more eloquent and efficient to use. This is the main reason why a new approach, specialized in the educational domain is needed. Thus, designing Educational Adaptive Hypermedia Applications is an open research issue (Brusilovsky & Maybury, 2002).

The structure of this paper is as follows: In the following section, the CADMOS-D hypermedia design method is outlined. The steps and the outcomes of each step of the design method are presented next. In the following section an approach for the automatic generation of educational applications is discussed. A review of the related literature is presented afterwards and the paper ends with some concluding remarks.

The CADMOS-D design method

In this paper, we propose the CADMOS-D design method which captures the outcomes of instructional analysis and drives the development of the whole EAHA. CADMOS, which stands for a CoursewAre Development Methodology for Open instructional Systems, proposes a sequence of phases for the development of web-based educational applications. These phases are requirements capturing, design, implementation and evaluation. CADMOS proposed a specific method, named CADMOS-D, to support the design phase. We are in the process of extending CADMOS-D in order to support EAHA design.

CADMOS-D, as a design method, provides two distinct models for educational web applications development: A process model, that pertains to the detailed definition and specification of the various design steps, their temporal relationships and sequencing and a list of the outcomes of each step, and a product model (Lowe & Hall, 1999) that refers to the detailed specification of the outcomes of each step, capturing the design decisions, the relationships and dependencies between these outcomes and the mechanisms that allow these outcomes to drive the development of the actual application. Furthermore, the product model can form the basis for the description of existing applications, provide the blueprints that depict knowledge and common understanding for particular applications, either completed or under development, much in the way that the blueprints of a building can both drive its development and depict its form, structure and function. The proposed model is defined as a UML profile (OMG, 2003). This profile is specified by the extension of basic UML elements, the definition of additional semantics for the new elements and as well as the definition of syntactic constraints for the interconnection of these elements, beyond these defined in the specification of the UML itself. The design model can be decomposed into three sub-models: conceptual, navigational and user interface (presentation) models.

The Conceptual Model

The *Conceptual Model* defines the learning activities that students will be engaged in during the instructional process of a specific subject, together with the semantic interrelationships between these activities. The learning activities are applied to the various thematic concepts-topics of the domain. The thematic topics should be considered as the Ontology of the subject domain to be learned by the students. The Conceptual Model provides a didactic solution over the objective definition of the knowledge subject. This definition is provided by the authors of the educational application who are considered as subject matter experts. They have arranged that body of knowledge hierarchically, subdividing the field into areas, which are then broken down further into units and individual topics. For each set of topics a set of learning objectives is defined. An overview of the body of knowledge for the domain of Computer Science appears in (IEEE, 2001).

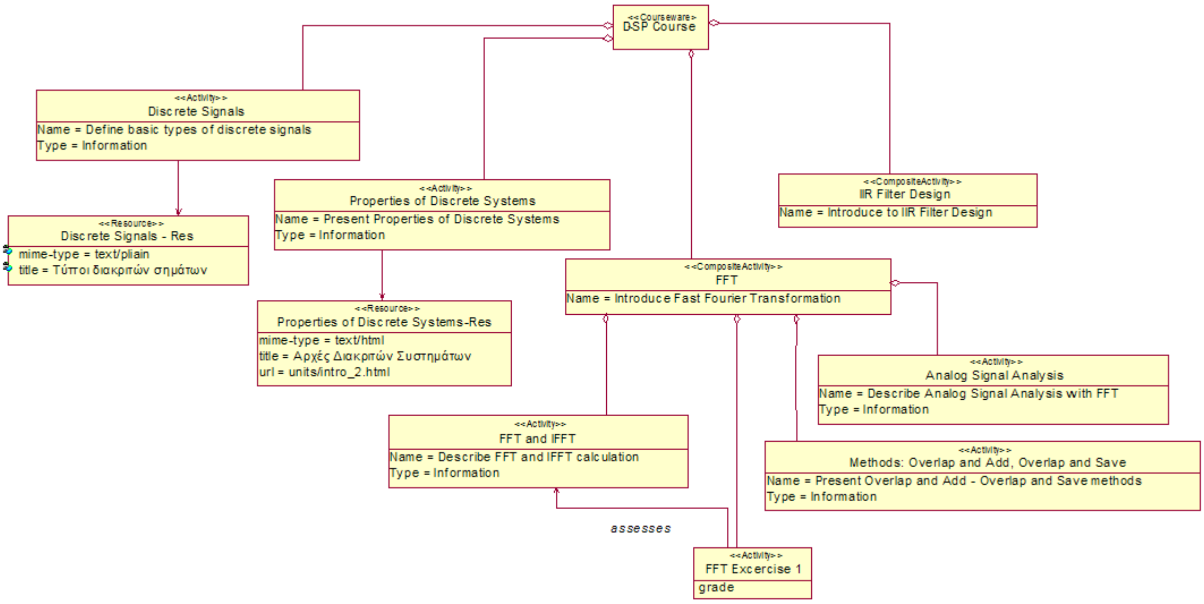

Figure 1. Extract of the Conceptual Model of a course on Digital Signal Processing

Each learning activity is related to particular learning objectives, notions and terms to be taught, etc, according to the syllabus. The activities are hierarchically structured, since composite activities can encapsulate simpler ones. The hierarchy of activities corresponds to the hierarchy of learning objectives, that the learner has to meet via her/his interaction with an educational application under design. Different types of activities exist: Information activities, where the learner access new information, interactive activities, where the learner is dynamically interacts with the educational content, and assessment activities during which the assessment or self-assessment of the learner's knowledge or achievement of the learning objectives is evaluated. Apart from their hierarchical organization, activities can be associated with each other with specific interrelationships thus forming a semantic network that provides an abstract representation of the solution of the problem of instruction of a specific topic. This particular view can be reused per se, thus promoting the reusability of educational applications at an abstract level, apart from navigation and presentation issues. In this way, the proposed method incorporates the principle of separation of concerns and promotes reusability. The activities are associated with specific learning resources. The resources align with the notion of Learning Object. These resources are physical, reusable, binary entities, either static fragments of digital content, e.g. text, image, video, simulations etc, or dynamic content generated 'on the fly' from proper scripts in the context of a web-based application environment or Learning Management System. An example of a Conceptual model is illustrated in Figure 1, which concerns a hypermedia course on digital signal processing.

The hierarchical structure of the Conceptual Model diagrams defines an implicit ordering of activities. The children of an activity are visited after their parent, from left to right. Applying this scheme iteratively, the 'in-ordered' traversal of the activity tree defines the default sequence of activities.

The elements of this sub-model are expressed as stereotyped UML classes and they are actually attribute-value pairs connected with proper association relationships. The concepts are mapped to specific learning resources. The modeling elements of this submodel are:

- *Courseware*. This is the top-level element in the hierarchy of activities that compose the conceptual view of the application.
- *Activity*. This defines a simple activity which is an atomic one. This activity may contain specific attributes. Predefined attributes are the title and the type of the activity (information, assessment, etc).
- *CompositeActivity*. This element defines a composite activity, which contains others, either atomic or concept, thus forming a hierarchy of activities into the educational application.
- *Relationship*. This refers to the association between two activities, atomic or composite.
- *Resource*. This element defines the resources associated to specific activities.

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, The Navigation Structure Model and the Navigation Behavior Model. These two sub-models are presented in the following subsections.

The Navigation Structure Model

This model defines the structure of the EAHA and defines the actual web pages and the resources contained into these pages. An example of this model is shown in Figure 2. This structure is composed of the following elements:

- *Content*, which is the top-level container in the hierarchy of an electronic content organization.
- *Composite* 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 structure* elements, namely indexes and guided tours, which are related to Content or Composite components. An access structure contains structural links, that facilitate the navigation into the hypertext space of the application.
- *Nodes*, which are the actual pages of the learning content. Content, Composite and Nodes are associated with Concept elements, or directly with Resources, in the Conceptual Model.
- *Fragments* that are contained into the Nodes. Fragments correspond to Resource elements in the Conceptual Model.
- *Links* between Nodes as well as between Fragments. Note that these links are associative links (Garzotto et al., 1993; Rossi et al., 1995) 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.

The Navigation Structure Model defines, also an implied sequence of nodes. Again, this model has a hierarchical structure. This sequence is defined by the 'in-ordered' visit of the hierarchy of nodes. However, this default behavior can be altered by alternative sequencing which is specified in the Navigation Behavior Model, described later.

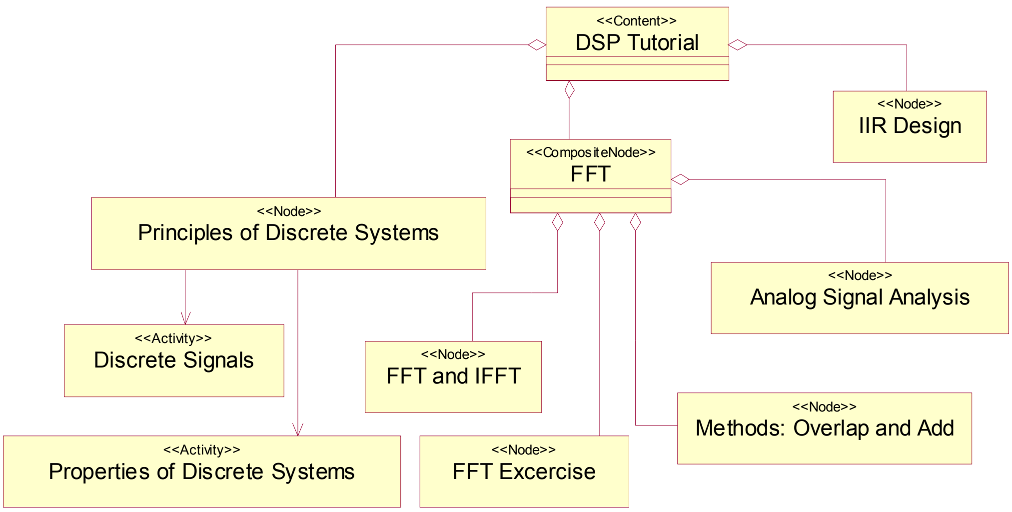

Figure 2. An extract of the Navigational Structure Model

Figure 2 illustrates an extract of a Navigation Structure Model, though not all elements of this model are used in it. In this figure the hierarchy of composite and simple nodes are displayed. As shown a node is associated to one or more learning activities from the Conceptual Model. For example, a designer could use a hypermedia node to incorporate both the presentation of a theoretical part of the subject domain along with an assessment task. Another designer might want to separate those two learning activities. It is obvious that the learning activity model remains intact in two different aforementioned cases. This fact allows the reusability of design models and the separation of concerns. CADMOS-D advocates that we should not think of "nodes" from the beginning. "Nodes" are the realisation in the hypermedia space of learning activities which should be designed first and which entail the decisions of the instructional design.

The Navigation Behavior Model

The Navigation Behavior Model defines the run-time behavior of the EAHA in terms of navigation. This behavior overrides the default run-time behavior, implicit in the previous Navigation Structure Model, which is elicited by the structuring of the hypermedia nodes. In this sense, this model supports adaptive navigation, which is the method of adaptation that is currently supported by CADMOS-D. Earlier research attempts, such as (De Oliveira et al., 2001), have 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 EAHA. Every containing element of the Navigation Structure Model (Content, or Composite) is associated to a composite state in the Navigation Behavior Model, while every Node 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 Nodes: next, previous, up level, etc. In addition, guard conditions in these transitions can define alternative navigational transitions, which correspond to conditional behavior of the EAHA, thus implementing content sequencing and adaptive navigation. These conditions have the form 'event[condition]' and are displayed as labels in the transitions in statechart diagrams. An example of such a design model is shown in Figure 3. In this example, the event 'continue' corresponds to a navigation request by the learner after she or he has solved a test (FFT Exercise). Depending on the performance in this test, the target page of this request can be the 'Analog Signal Analysis' or 'IIR Design' pages. Such rules can be defined by the designer implementing specific Instructional Design strategies.

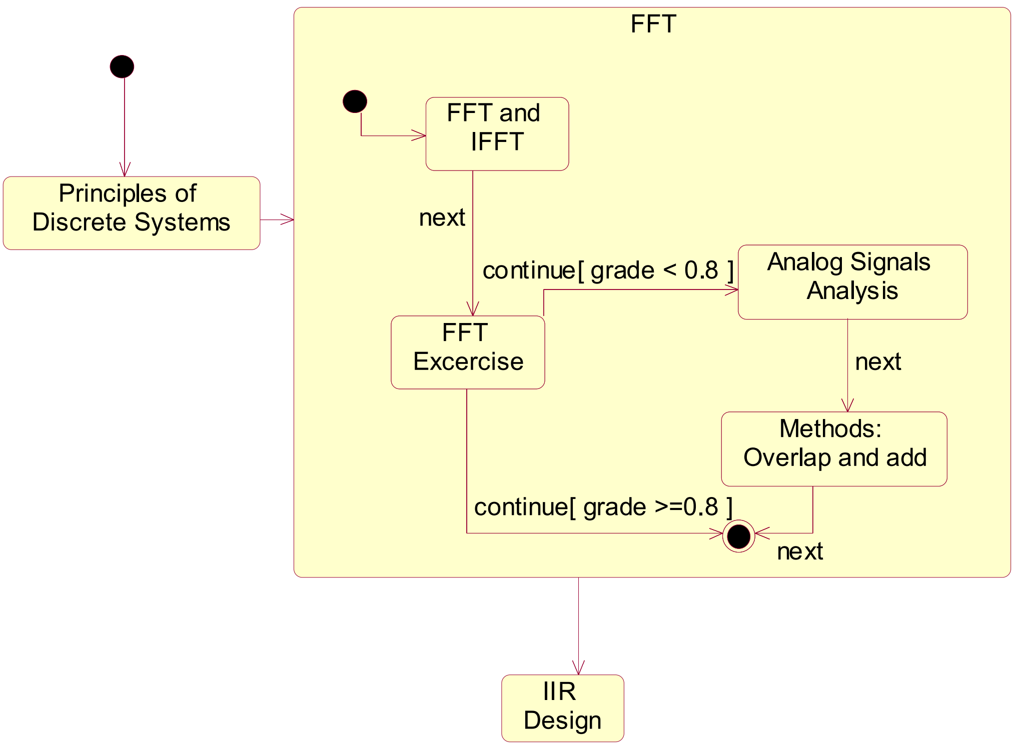

Figure 3. Example of Navigation Behavior Model

The User Interface (or Presentation) Model

The *User Interface Model* deals with the presentation aspects of the elements defined in the Navigation Model. In particular, each Node in the Navigation Model and its resources are associated with a presentation model element. Note that a multitude of navigation elements can be associated with the same presentation specification, thus promoting uniformity and ease of maintenance of the user interface. The Presentation Model elements have their counterparts in corresponding elements of web technology specifications, namely (X)HTML and CSS (W3C, 2004) elements. More specifically, the Presentation Model contains the following stereotyped UML classes: "html", that represents HTML elements or aggregations of HTML elements and "css" that actually represent Cascading Style Sheet classes.

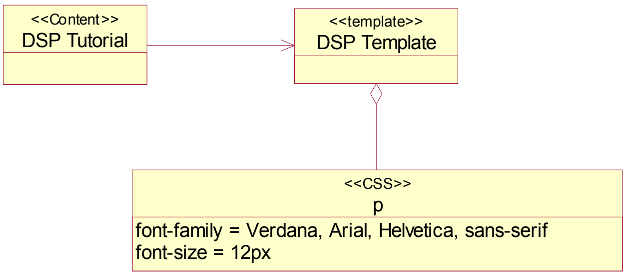

Figure 4. Example of the basic templates that have been used in the DSP courseware

Elements of the User Interface model are associated to particular nodes of the Navigation Model thus assigning specific presentation attributes to these nodes, as well to their children in the navigation structure hierarchy. In Figure 4 an example of the Presentation Specification of the DSP courseware is illustrated. The template 'DSP Template' contains, among others that are not displayed for the sake of clarity, a definition of css class 'p' for the formatting of paragraphs of text. The template is associated with the top-level element 'DSP Tutorial' of the Navigation Structure Model. This means that the formatting defined in this template is applied to all pages of the DSP tutorial.

Automatic application composition

The design model of an EAHA can be created using CASE tools like the IBM Rational Rose. CADMOS-D suggests the utilization of such tools, since UML models can be stored in XMI, the OMG standard XML metadata interchange format (OMG, 2003). In that way, it is possible to process and manipulate XMI data by standard XML processing tools (e.g. XML parsers). With the use of a specially developed tool, called CGA (Courseware Generation Application), the XMI description of CADMOS-D models are transformed into structured hypermedia educational applications. More specifically, the CGA tool accepts as entry the XMI description with the relevant learning resources (HTML pages, pictures, files of sound and video, active objects as Applets, ActiveX, Flash, etc.) and produces as output the real EAHA. This process is illustrated in Figure 5. This is a UML activity diagram, where the various artifacts are represented as rectangles (objects) and the activities or software tools in this case, are represented as activities (rounded rectangles). Dashed lines with arrows represent 'object flows' in UML terminology, that is connections of objects to processes that have as a result the creation or state change of these objects. The element 'Content Packaging' in the same figure is a note, used for commenting in UML diagrams.

The produced web pages are a composition of the provided resources wrapped with the proper XHTML code. These pages are accompanied by a description of their structure in the form of a XML manifest file. The XML manifest file conforms to the IMS Content Packaging learning technology specification for interchanging educational content (IMS, 2003a). In Figure 6 a part of the manifest file for the DSP courseware is depicted.

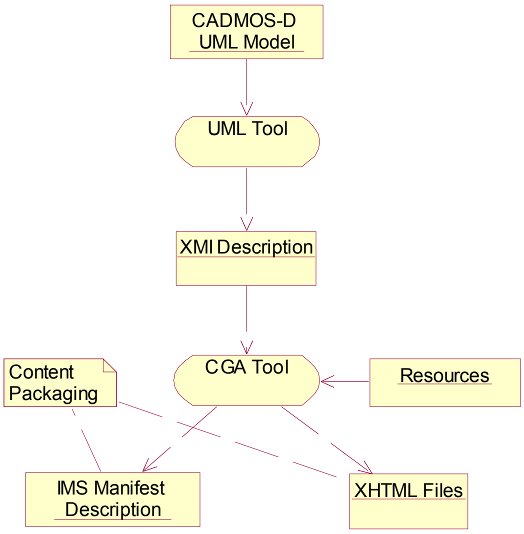

Figure 5. The process of creation of Educational Hypermedia Applications out of UML models

```
<imsmanifest version="1.3" identifier="TEST">
      <organizations default="TOC1">
            <organization identifier="TOC1">
                  <title>DSP Courseware</title>
                  <item identifier="S.10269" identifierref="S.10269_RES">
                        <title>Elements of Discrete Systems</title>
                  </item>
                  <!-- ...-->
            </organization>
      </organizations>
      <resources>
            <resource identifier="S.10269_RES" href="units/intro_1.html">
                  <title>Elements of Discrete Systems</title>
            </resource>
            <!-- ... -->
            <resource identifier="S.10319_RES" href="S.10319.html">
                  <title>IIR Design</title>
            </resource>
      </resources>
</imsmanifest>
```

Figure 6. An extract of a Content Packaging Manifest file

This XML manifest file accompanied by the learning resources can be deployed to any Learning Content Management Systems that support the IMS Content Packaging specification in order for the educational application to be delivered to its users. We experimented with the SCORM Sample Run-Time Environment (SCORM, 2004) for the course on Digital Signal Processing. A screenshot of this course is depicted in Figure 7.

At the moment, the CGA tool cannot make use of the information about the dynamic navigational behaviour, i.e. the state-transition diagrams incorporated into the models. The designer creates distinct design models per user type (users with different stereotype). Thus, we create educational hypermedia applications that provides a variety of personalised views of the domain per user type focusing on composition and structural relationships between the learning activities and the respective nodes. For each view, the designer can associate templates in order to specify the look and feel of the nodes. In this way the generated applications are not adaptive but, in a way, adaptable.

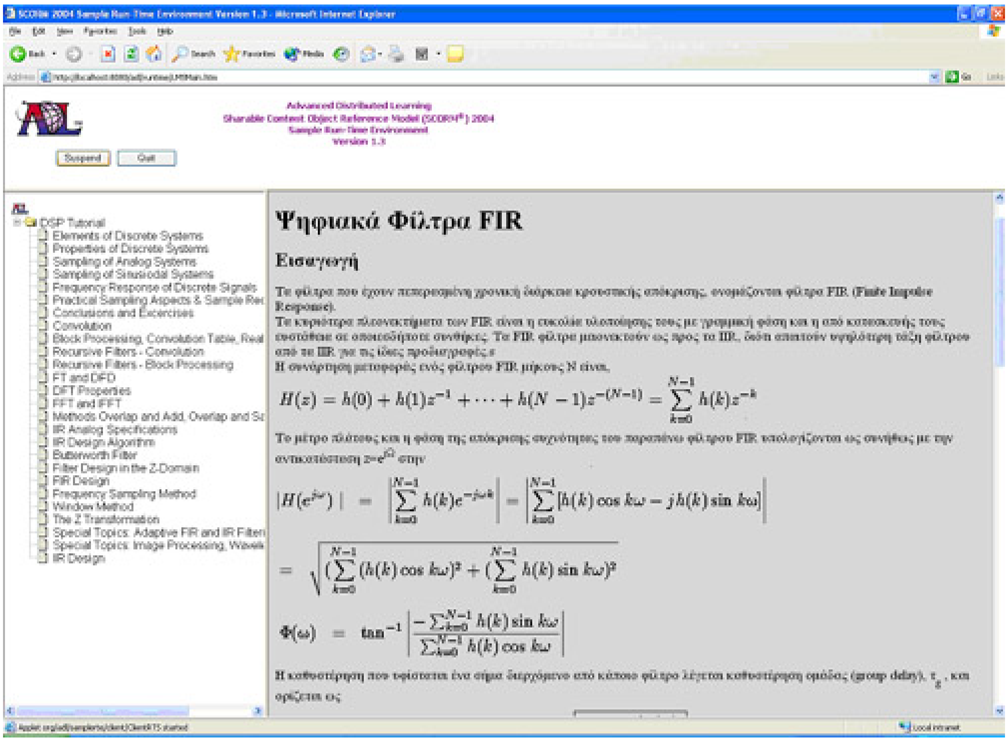

Figure 7. A screenshot of the DSP tutorial

Related Work

This work aspires the bridging of the gap between the conceptual description and the implementation of educational web applications as it is also suggested in (Aroyo et al., 2002). The depiction of the instructional design solution as a set of interrelated learning activities and associated resources is also found in the ACCT toolkit methodology (Dagger et al., 2004). This scheme of activities and resources is also Like other approaches for the design of generic web applications such as WebML (Ceri et al., 2000), WCML (Gomez et al., 2001; UWE (Hennicker & Koch, 2001), etc. it maintains the classical, in hypermedia engineering, discrimination of the design of web applications into structure, navigation and presentation design, and uses XML as the product model for the implementation of actual applications. The use of XMI and the focus of the CADMOS-D model on the specific domain of education, which sets certain constraints in the structure of applications makes it different from the aforementioned methods. The current work has also close similarities to (Dolog & Nejdl, 2003), which also uses the same model representation, XMI, and the same method for adaptive educational application generation. The main difference with this method is the provision for navigation and presentation issues, which is not covered in (Dolog & Nejdl, 2003), and the support for Learning Technology Standards.

Conclusions

A design method like CADMOS-D can be used as a framework (Garzotto et al., 1993) 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 the design model, the development of open, portable, and maintainable EAHA can be facilitated with the adoption of learning technology standards. In this paper we are proposing the CADMOS-D design method that produces models that accord to the IMS content packaging. With the use of the CGA tool that has been developed, the EAHA is automatically generated using the XML manifest file and the learning resources. The lack of dynamic navigational structure is a limitation of our approach, but not an unsolved problem. In a new release of our application generation tool, which is not yet available, the UML activity diagrams will be transformed into IMS Simple Sequencing schema (IMS, 2003b) that will supplement the content packaging description. This schema defines rules for personalised sequencing of navigation into the educational content.

Acknowledgements

Many acknowledgements to Mrs. V. Siskos who produced the learning resources for the DSP courses.

References

Aroyo, L., Dicheva, D., & Cristea, A. (2002). Ontological Support for Web Courseware Authoring. In Ceri, S., Guardères, G., Paraguaçu, F. (Eds.), *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, Lecture Notes in Computer Science, 2363, Berlin Heidelberg New York: Springer-Verlag, 270-280.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language User Guide*, Reading, MA: Addison Wesley.

De Bra, P., Houben. G., & Wu, H. (1996). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In Westbomke, J., Wiil, U. K., Leggett, J. J., Tochtermann K., & Haake J. M. (Eds.), *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots,* New York: ACM Press, 147-156.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction, 6*, 87-129.

Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. *Kunstliche Intelligenz, 4*, 19-25.

Brusilovsky, P., & Maybury, M. (2002). From adaptive hypermedia to the adaptive web. *Communications of the ACM, 45* (5), 30–33.

Ceri, S., Fraternali, P., Bongio, A. (2000). Web Modeling Language (WebML): a modeling language for designing web sites. *Computer Networks and ISDN Systems, 33* (1-6), 13–157.

Dagger, D., Wade, V., & Conlan, O. (2004). Developing Active Learning Experiences for Adaptive Personalised eLearning. *Paper presented at the 2nd International Workshop on Authoring Adaptive and Adaptable Educational Hypermedia*, retrieved July 18, 2005 from https://www.cs.tcd.ie/Owen.Conlan/publications/ah2004_daggerd_v1.pdf.

De Oliveira, M., Turine, M., & Masiero, P. A. (2001). Statechart-Based Model for Hypermedia Applications. *ACM Transactions on Information Systems, 19* (1), 28–52.

Dolog, P., & Nejdl, W. (2003). Using UML and XMI for Generating Adaptive Navigation Sequences in Web-Based Systems. In Stevens, P., Whittle, J. & Booch, G. (Eds.), *Lecture Notes in Computer Science*, 2863. Berlin Heidelberg New York: Springer-Verlag, 205-219.

Fischer, S. (2001). Course and Exercise Sequencing Using Metadata in Adaptive Hypermedia Learning Systems. *ACM Journal of Educational Resources in Computing, 1* (1).

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

Gomez, J., Cachero, C., & Pastor, O. (2001). Conceptual modeling of device-independent Web applications. *IEEE Multimedia, 8* (2), 26–39.

Hennicker, R., & Koch, N. (2001). Systematic Design of Web Applications with UML. In Siau, K. & Halpin, T. (Eds.), *Unified Modeling Language: Systems Analysis, Design and Development Issues*, Hershey, PA: Idea Group Publishing, 1-20.

IEEE (2001). *Overview of the CS Body of Knowledge*, retrieved July 18, 2005, from http://www.sigcse.org/cc2001/cs-overview-bok.html.

IMS (2003a). *Content Packaging Specification*, retrieved July 18, 2005, from http://www.imsglobal.org/content/packaging/.

IMS (2003b). *Simple Sequencing Specification*, retrieved July 18, 2005, from http://www.imsglobal.org/simplesequencing/.

Lowe, D., & Hall, W. (1999). *Hypermedia & the Web, an Engineering Approach*, West Sussex, England: John Wiley & Sons.

OMG (2003). *UML Version 1.5 Specification*, retrieved July 18, 2005, from http://www.omg.org/technology/documents/formal/uml.htm.

Rossi, G., Schwabe, D., Lucena, C., & Cowan, D. (1995). An Object-Oriented Model for Designing the Human-Computer Interface of Hypermedia Applications, In Fraisse S. (Ed.), *Proceedings of the International Workshop on Hypermedia Design (IWHD'95)*, New York: Springer-Verlag, 123-143.

Schwabe, D., & Rossi, G. (1995). The Object-Oriented Hypermedia Design Model. *Communications of the ACM, 38* (8), 45-46.

SCORM (2004). *SCORM 2004 Sample Run-Time Environment Version 1.3.3*, retrieved July 18, 2005, from http://www.adlnet.org/downloads/197.cfm.

W3C (2004). *Cascading Style Sheets Home Page*, retrieved July 18, 2005, from http://www.w3.org/Style/CSS/.