Papasalouros, A., Retalis, S., & Papaspyrou, N. (2004). Semantic description of Educational Adaptive Hypermedia based on a Conceptual Model. *Educational Technology & Society*, 7 (4), xx-xx.

Semantic description of Educational Adaptive Hypermedia based on a Conceptual Model

Andreas Papasalouros[1], Symeon Retalis[2] and Nikolaos Papaspyrou[1]
[1] National Technical University of Athens
School of Electrical and Computer Engineering
Software Engineering Laboratory
Polytechnioupoli, 157 80 Zografou
Athens, Greece
Tel: +30 210 772 2487
Fax: +30 210 772 2519
Email: {andpapas,nickie}@softlab.ntua.gr

[2] University of Piraeus
Department of Technology Education
and Digital Systems
80 Karaoli & Dimitriou
185 34 Piraeus, Greece
Tel: +30 210 414 2765
Fax: +30 210 414 2753
Email: retal@unipi.gr

Abstract

The role of conceptual modeling in Educational Adaptive Hypermedia Applications (EAHA) is especially important. A conceptual model of an educational application depicts the instruc-tional solution that is implemented, containing information about concepts that must be ac-quired by learners, tasks in which learners must be involved and resources that will be used. The importance of such a model is multifold. It formally depicts the outcomes of instructional analysis, drives the development of actual applications and provides the information basis for automatic application generation in the context of Adaptive and Intelligent Tutoring Systems. In this paper we begin by describing a method for designing EAHA. Our method uses UML as the modeling language and defines the design process in three distinct steps. We propose an RDF encoding of the Conceptual Model, which is the outcome of the first step, following a specific RDF schema that is appropriate for such applications. This encoding can be obtained automatically and, in this way, a machine processable semantic description of the EAHA becomes available, as an outcome of the design process. In the resulting RDF-encoded Concep-tual Model, we define rules of adaptation using the existing language RuleML.

Keywords

Adaptive hypermedia, hypermedia design, conceptual modeling, semantic web, RDF.

Introduction

Educational Adaptive Hypermedia Applications (EAHA) provide personalized views of Educational Hypermedia to individual learners. They are gaining the focus of the research community as a means of alleviating a number of user problems related to hypermedia. However, despite the growing interest of the community and the increasing number of available systems, the actual impact of these systems in e-learning remains low. The difficulty and complexity of developing such applications and systems have been identified as a possible reason for this low diffusion of Adaptive Hypermedia in web-based education.

The development of EAHA is a complex task engaging people with different backgrounds: instructional designers, subject matter experts, content developers, multimedia developers, user interface experts, programmers, etc. Experience from traditional Software Engineering as well as Hypermedia Engineering suggests that a model-driven design approach is appropriate in developing applications where such requirements and constraints occur. This approach has a number of benefits:

- It facilitates the communication of the various stakeholders involved into the development process.
- It captures and depicts high-level design decisions and solutions at various levels of abstraction. These decisions and solutions are not only related to implementation issues but also to higher-level concerns.

- It establishes a disciplined development process.
- It automates the development of the final product by using specialized tools.
- It provides an intuitive, easy to comprehend and understand view of the applications under development through applying visual modeling techniques. A design model can be derived from existing applications so as to describe their architecture, structure and functionality in a process known as reverse engineering.

Existing 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 (Lowe & Hall, 1997) emerged to address this issue, existing Hypermedia Engineering methods are not adequate for properly dealing with the design of educational hypermedia applications. While such models and methods have been successful in modeling the navigation and presentation issues, they fail to capture the abstract, conceptual issues of such applications. This results from the fact that traditional hypermedia applications are "information-oriented", that is, they consider the hypermedia-hypertext structure as a front-end for accessing structured information, usually stored in databases or other information systems. Thus, conceptual models of such applications provide models of highly structured information and a description of the business logic underlying these applications. Conversely, educational hypermedia applications — either adaptive or not — are not information- but learning-oriented. That is, the added value for a user of such applications is not access to information, but learning, as an outcome of planned instruction. Learning is the result 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 (Smith & Ragan, 1999). Furthermore, existing Hypermedia Engineering methods cannot either capture the issues specific to adaptivity or describe the semi-structured nature of such applications and systems. Thus, designing Educational Adaptive Hypermedia is an open research issue (Brusilovsky & Maybury, 2002).

In (Retalis & Skordalakis, 2001) a method has been proposed, named CADMOS (CoursewAre Development Methodology for Open instructional Systems), that proposes a sequence of phases for the development of web-based educational applications. These phases are requirements capturing, design, implementation and evaluation. CADMOS has a specific component, 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* (Lowe & Hall, 1999; Sommervile, 1997), 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.

While the process model of CADMOS-D has been left in its original form, the product model was updated in order to meet the requirements of adaptivity. The Unified Modeling Language (Object Management Group, 2003) has been used as a visual notation and modeling language for the design model of CADMOS-D. As it will be discussed later, the product model of CADMOS-D is separated into three sub-models: A Conceptual Model, a Navigation Model and an Interface Model. This is a widely adopted distinction in Hypermedia Engineering. In this way, the domain specific issues of the application are captured in the Conceptual Model. The connection between conceptual primitives and navigation elements such as pages and hyperlinks are depicted in the Navigation Model. Last, presentation issues such as styles, layouts, page templates, are depicted in the Interface Model.

This paper focuses on the Conceptual Model of EAHA, which is of particular importance for designing such applications. A conceptual model of an educational application depicts the instructional solution that is implemented, containing information about concepts that must be acquired by learners, tasks and activities in which learners must be involved and resources that will be used. The importance of such a model is multifold. It formally depicts the outcomes of instructional analysis, drives the development of actual applications and provides the information basis for automatic application generation in the context of Adaptive and Intelligent Tutoring Systems. It also reveals the internal structure of an educational application regardless of the technology used in any particular system hosting the application.

We provide an RDF encoding for the Conceptual Model of EAHA that are designed using CADMOS-D. More specifically, we first propose a UML meta-model that abstracts the conceptual models of all EAHA and we encode this meta-model in an RDF schema. Next, we propose an encoding for conceptual models of specific EAHA, i.e. instances of the previous meta-model, that can be obtained automatically given the corresponding UML diagram. Last but not least, we propose an encoding of the rules that implement adaptivity in EAHA using the existing language RuleML (http://www.ruleml.org). This encoding can also be obtained automatically, provided that there has been some formal specification of the adaptation rules in the original model, through rules in an appropriate language, e.g. OCL.

Through the encoding of the Conceptual Model of EAHA using Semantic Web technologies, we obtain a machine processable semantic description of EAHA, that can be produced automatically as an outcome of the design process. Whereas the other outcomes of the design process, e.g. the UML models, are only valuable for the developers of the EAHA, this semantic description is valuable in its own right, in parallel to the application itself.

The Semantic Web provides globally meaningful representations of concepts and their interrelationships that go beyond the scope of a particular system, application or organization. Furthermore, the information described using Semantic Web technologies, such as RDF or OWL, is organized in a form that permits automatic logic reasoning based on this information, involving Artificial Intelligence techniques. Although semantic web technologies are not the best candidates for conceptual modeling during the design process (e.g. as compared to a modeling language like UML), their adoption has a definite added value in the development of EAHA for the following reasons:

Design reuse

The Conceptual Model captures the decisions concerning the organization of the activities that the learners are involved in during their interaction with the application and the association of these activities to particular web-based resources in a particular application. These design decisions concerning a particular educational domain have a value of their own and can be reused. For example, a particular design of a tutorial on the subject of programming languages, i.e. the organization of the topics of study, the examples, exercises and tests taken by the learners compose a set of decisions that can be used by some author of another tutorial on the subject, regardless of the actual content that will be used for the development of the tutorial.

Formalization of the description.

The Unified Modeling Language is a visual, standard notation that was extended so as to describe the Conceptual Model of an EAHA. Apart from its certain advantages, namely intuitive presentation of the design decisions, ease of use, ease of communication, etc, UML inherently lacks a strict formalism. RDF, on the other hand, is a formal, machine processable notation. Furthermore, the basis for RDF all the Semantic Web descriptions is XML. This facilitates the integration of these descriptions with both UML (through XML Metadata Interchange – XMI) (OMG, 2003) and the existing learning technology standards which all have XML bindings.

Establishment of common understanding

In the conceptual model a specific distinction is made between high-level, abstract entities –*concepts* and *relationships*– and specific real-life objects such as *resources*. RDF semantics aligns well with this distinction, as it provides a basis for the creation of common language for expressing metadata with specific syntax and commonly agreed semantics.

Design sharing and retrieval

An RDF representation of the conceptual design of an AEHA provides a meaningful semantic representation of the subject domain as well as the strategies employed for the teaching of this domain. Furthermore, this representation is globally unique, has specific meaning and is machine processable. As a consequence, the resulting representations can be used by proper design tools that will contain proper search agents in order to provide the appropriate resources available on the Web for the designer. This search may be conducted to specific learning resource brokers or peer to peer networks such as Educanext (http://www.educanext.org/) and Edutella (http://edutella.jxta.org/), or more generic platforms for the management of RDF data, such as RDFSuite (Alexaki et al., 2001). Thus, both the *provision* and the integration of learning resources into an application are made possible.

As an example application, throughout the paper, we use a tutorial on Fire Safety. This tutorial was initially developed by the Emergesmart company (http://www.emergesmart.com) using the LRN Toolkit (Microsoft, 2003), a tool for creating and structuring on-line tutorials, that did not have any adaptive features. It was extended in order to sequence the activities offered to the learners according to their performance in specific tests. The IMS Simple Sequencing Learning Technology Standard (IMS, 2003) was used so as to formally specify the structure of the learning material and the conditions under which part of the material is selected and delivered to learners during their interaction with the tutorial.

The structure of this paper is as follows: In the following section, the CADMOS-D hypermedia design method is outlined. The product model for EAHA used by CADMOS-D is presented next. It is followed by a description of its encoding, using Semantic Web technologies, and a discussion on rules of adaptation and their possible encoding. The paper ends with some concluding remarks.

CADMOS-D: A Hypermedia Design Method

CADMOS-Design (CADMOS-D) is a method for the creation of the detailed design of web-based educational applications, which includes structural details of the learning resources, the navigational schema and templates for describing abstractly the graphical user interfaces. This method follows the principles of the Object Oriented Hypermedia Design Method (OOHDM) (Schwabe & Rossi, 1995; Schwabe & Rossi, 1998), which has provided a systematic way to design (generic) hypermedia applications.

The CADMOS-D method belongs to a web-based educational application (WbEA) development methodology named CADMOS. It proposes a stepwise design process model, as shown in **Figure 1**: Conceptual Design, Navigational Design and Interface Design. The intermediate products of each step are validated according to guidelines for formative evaluation of the instructional design (checking structural, navigational, aesthetics and functional issues). The whole design process is considered to be iterative, where in each iteration loop the design artifacts are evaluated and the feedback from the evaluation is used for their improvement, until they reach the desirable level of detail so that they can drive the actual implementation.

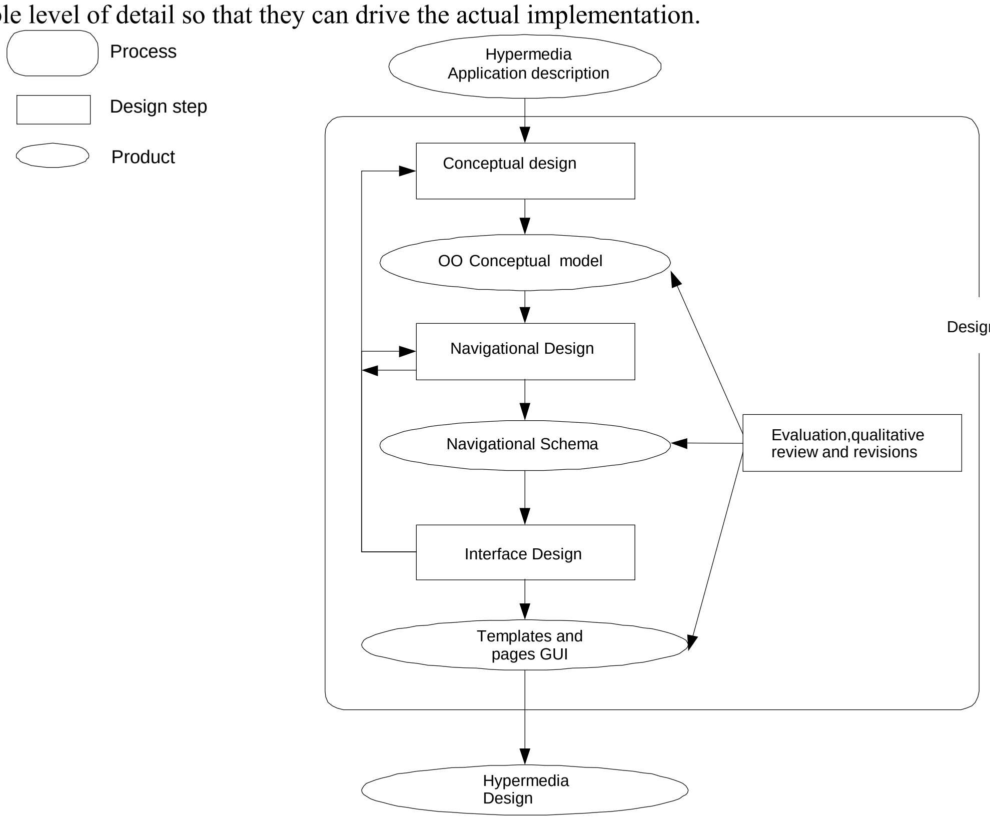

Figure 1. The three design steps in CADMOS-D.

Conceptual Design

Although well known and widely used in hypermedia design, this step has a somewhat different purpose in the current approach. The Conceptual Design step aims at describing the abstract solution of the individual learning problem, which has been identified by instructional designers, subject matter experts or pedagogists into a systematic form that will guide the development of the actual educational application. This abstract solution is preferably defined as a set of learning objectives, instructional events associated with these objectives, the syllabus, and the assessment procedures (Retalis & Skordalakis, 2001; Ford et al., 1996). It is intended to facilitate the transformation of an abstract description of an educational application into a hierarchy of concepts to be taught. Each concept is related to particular learning objectives, notions and terms to be taught, etc. The hierarchy of organized concepts corresponds to the hierarchy of the terms, notions, intellectual skills, that the learner is to acquire via her/his interaction with an educational application under design. *Concepts* are notions to be taught using the specific application, as well as learner tasks or activities, described in an abstract manner, that facilitate the learning of these concepts or the achievement of specific learning objectives. Apart from their hierarchical organization, concepts can be associated with each other forming a semantic network that provides a conceptual view of the domain to be taught and the adopted instructional strategies. This particular view can be reused per se, thus promoting the reusability of educational applications at the level of the conceptual design, apart from navigation and presentation issues. In this way, the proposed method incorporates the principle of separation of concerns and promotes reusability. The designer defines this structure and specifies the particular *resources* associated to each concept. The concepts together with the associated resources align with the notion of Learning Object Metadata (http://ltsc.ieee.org/wg12/) associated to particular learning objects. These resources are considered as either static fragments of digital content, e.g. text, image, video, simulations etc, either as dynamic content generated 'on the fly' in the context of a web-based application environment (e.g. CGI, PHP, ASP, JSP, ColdFusion, etc) or Learning Management System. The concepts can be composite, if they contain other concepts, or atomic. The granularity of the resources is not specified.

In this way, an anarchic aggregate of available, reusable, learning resources is structured into an organized mosaic according to the instructional design of the course. Note that the use of existing, reusable resources is not compulsory. New resources can be authored or implemented while developing a new application and can be integrated into the application during the conceptual design phase. The Conceptual Model is a set of UML class diagrams. For facilitating the construction of these diagrams CADMOS-D has proposed an abstract object oriented meta-model (Papasalouros & Retalis, 2002) which is independent of the underlying subject domain of the application (i.e. mathematics, geography, etc.) but provides a suitable platform to describe structural and navigational issues of the learning resources. This model constitutes the input for the next step, Navigation Design.

Navigational Design

In this step the navigational schema of the Educational Application is analytically designed, so that it is clearly specified how the previous structure of resources is transformed into web pages and how these pages are inter-connected with hyperlinks. More importantly it facilitates the maintenance of the web site, especially when web pages are added or deleted and hyperlinks to and from them have to be updated. In this way, the well-known problem of 'dangling' links can be avoided. The navigational structures proposed for this kind of design are well accepted by many hypermedia design approaches, such as HDM (Garzotto et al., 1993), RMM (Isakowitz et al., 1995; Isakowitz et al., 1997) and OOHDM (Schwabe & Rossi, 1995; Schwabe & Rossi, 1998). More specifically they are: a) indices that provide direct access to every indexed node, b) guided tours which are linear paths across a number of nodes and c) indexed guided tours which combine the two previous structures.

Interface Design

In this step, the Graphical User Interface (GUI) of the hypermedia application is designed, that is the content, layout and 'look and feel' of the web pages. Interface design is ruled by the principles of the *page metaphor*, a practice taken from multimedia engineering where it has been extensively adopted and used. The page metaphor is used to specify the page components with graphic symbols and deploy them on the screen showing their layout. Therefore, with the use of graphical semantics, the design depicts the page form just as it will be implemented. The data model for the interface design contains six kinds of page components: plain text, multimedia elements, active elements, hyperlinks, frames and form elements. The designs made are actually re-usable page templates. For instance, if we design the page template of one paragraph of an on-line book in a hypermedia application, then all the other paragraphs of the book might have the same look, using the same components with the same layout, have the same frames etc. During the interface design, except for designing page components and their layout, we define certain metadata on them.

Description of the design Model

The outcomes of the CADMOS-D process steps depicted in **Figure 1** constitute a product model for Educational Applications. We have extended this model towards the direction of adaptivity. The new product model follows the traditional approach in hypermedia engineering of separation of an application model into a Conceptual, a Navigation and a User Interface view or sub-model. These views describe the products of the corresponding design steps (see **Figure 1**). Furthermore, a Learner Model and a Teaching Model are provided in order to adequately describe the adaptive features of Educational Applications. An overview of the model components and their dependencies is depicted in **Figure 2**. Note that the Conceptual, Navigational and User Interface Models comprise the Hypertext Model and correspond to the outcomes (artifacts) of the three design steps of CADMOS-D, namely Conceptual Design, Navigational Design and User Interface Design.

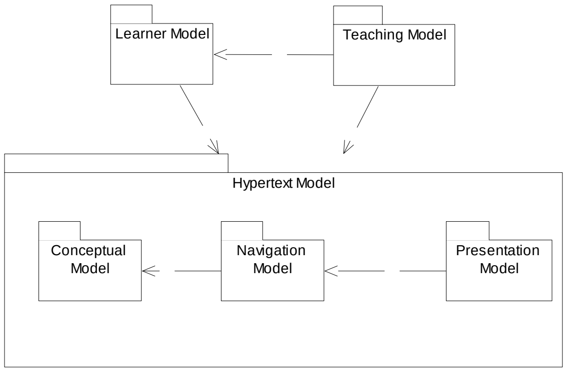

Figure 2. Overview of the Design Model.

All the above models are expressed using the Unified Modeling Language. The language has been properly extended defining new modeling elements (stereotypes) and defining their allowed relationships for the particular domain of educational applications. The extensibility mechanisms of this language, namely stereotypes and constraints have been employed in order to define this extension by means of a UML profile (OMG, 2003).

The Conceptual Model

The Conceptual Model contains the basic concepts presented in the specific educational application together with their semantic interrelationships. These concepts are expressed as attribute-value pairs connected with specific associations. The elements of the conceptual model are associated according to the meta-model illustrated in **Figure 3**. **Figure 4** depicts the conceptual model of the Fire Safety Tutorial example application, which is an instance of the meta-model depicted in **Figure 3**. The concepts are mapped to specific learning resources. The modeling elements of this meta-model are:

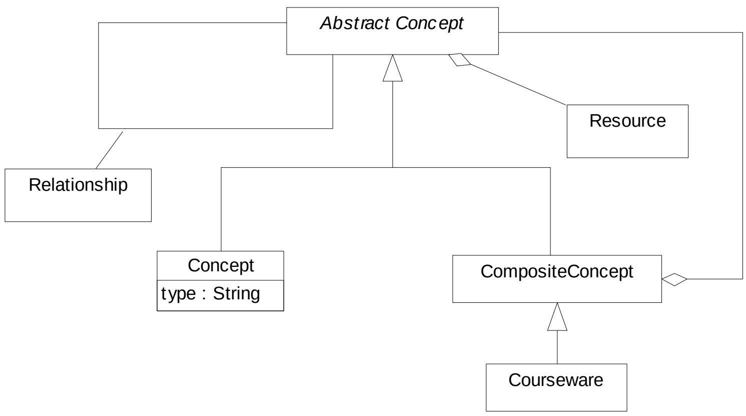

Figure 3. UML description of the Conceptual meta-model.

Courseware

This is the top-level element in the hierarchy of concepts that compose the conceptual view of the application. In the example illustrated in Figure 4 the top-level element is “Fire Safety Tutorial”.

Concept

This defines a simple, atomic concept. This concept contains specific attributes. A concept can be a specific topic for study, an activity for assessment or dynamic interaction, simulation, etc. Note that this element refers to an abstract aspect of the specific entity as the result of instructional design. It does not correspond to any concrete media element that is contained into the specific application. A concept has a predefined attribute named *type,* which denotes the kind of learning activity it engages in, that is, gaining information, assessment, interaction with active elements, etc. An example of Concept shown in **Figure 4** is “Introduction”, which is a concept of type “Information”. This represents the instructional activity of introducing the learner to the subject of fire safety and presents the topics that will be discussed in the tutorial. Note that the instructional activity represented by this concept is independent of the learning resource (see corresponding element later in this section) that will actually implement this concept, e.g. text, video or other. Another example of Concept in the same figure is “Pretest”. This is a concept of type “Assessment” and its purpose is to assess the knowledge of the learner.

Composite Concept

This element defines a composite concept, which contains other concepts, atomic or composite, thus forming a hierarchy of concepts into the educational application. It can be a chapter or topic containing other sub-topics. Using composite makes possible the organization of the learning activities into units of instruction such as, for example, sections, lessons, etc. “Types of Fire Extinguishers” is an example of a composite concept for the Fire Safety tutorial depicted in **Figure 4**.

Resource.

A resource describes the actual media element that has been developed to provide a concrete, binary entity for a concept, atomic or composite. A resource is close to the notion of a reusable learning object, a chunk of information of arbitrary granularity that can be used in the development of an educational application or courseware. It corresponds to the notion of Learning Object Metadata widely used in the field of Educational Technology Standards. In the Fire Safety Tutorial the video file “Intro_video” is a resource that actually implements the “Introduction” informative concept.

Relationship

This refers to the association between two elements, either concepts or resources. For example, two concepts can be connected with a relationship denoting that the one is an assessment over the other. In **Figure 4**, there is an association between concept “Pretest” and “Water Fire Extinguishers” meaning that “Pretest” assesses the knowledge on “Water Fire Extinguishers”.

The top-level element in the hierarchy of concepts is “Fire Safety Tutorial” which is the name of the actual educational application. This is represented in the same figure with the stereotype *Courseware*. Based on the solution of the instructional problem, this is decomposed into the following concepts, looking from left to right in the figure: “Introduction”, which presents an introduction to the tutorial in the form of a video. This video, namely, “intro_video.wmv” is described as a *Resource* element with the same name. Correspondingly, the concept “Pretest, is associated with the “Pretest_html” *Resource*. The composite concept “Types of Fire Extinguishers” is decomposed into three simple concepts, namely “Water”, “CO2” and “Dry Chemicals” Fire Extinguishers. These simple concepts are associated with the “APW”, “BC” and “ABC” resources, respectively, which are actually shockwave applications. The rest of the UML elements are self-explanatory.

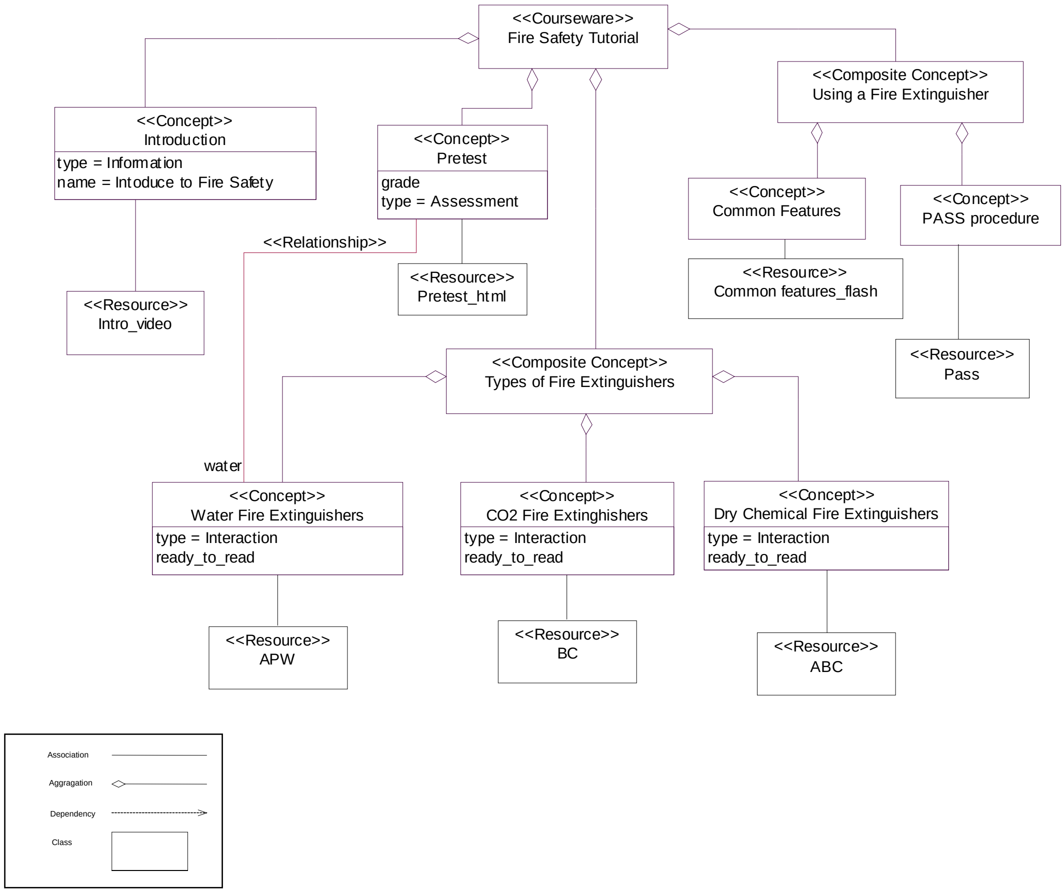

Figure 4. Conceptual Model of the Fire Safety Tutorial (Detail)

The RDF description of the Conceptual Model

The Conceptual Model described above is expressed using the Resource Description Framework – RDF (Manola & Miller, 2003). The mapping between UML and RDF was based on Chang (1998). The RDF is a language for providing information about resources on the web. This information is provided as properly structured metadata about resources pertaining to specific application domains. For each domain, there is a need to establish a common vocabulary for the domain-related information of the resources. This is achieved using a language for defining specific RDF vocabularies, RDF Schema (Brickley & Guha, 2003). In this sense, RDF Schema is a meta-language, also expressed in RDF, for the definition of specific RDF bindings.

Our approach for the definition of RDF descriptions of specific educational hypermedia designs follows two steps: First, a Schema of the Conceptual Meta-model itself is provided in Figure 5. This is actually a slightly modified version of the meta-model described in (Papasalouros & Retalis, 2002). In this schema the basic elements of the conceptual model, i.e. the elements Concept, Relationship, Resource, etc, are defined as RDF Schema classes, while their predefined attributes are defined as properties of these classes. Note the *related Resource* property that relates *Concepts* to actual learning resources.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xml:base="http://www.softlab.ntua.gr/andpapas/schemas/conceptual">

    <rdfs:Class rdf:ID="Courseware"/>
    <rdfs:Class rdf:ID="Concept"/>
    <rdfs:Class rdf:ID="CompositeConcept"/>
    <rdfs:Class rdf:ID="Resource"/>

    <rdfs:Property rdf:ID="name">
```

```
        <rdfs:domain rdf:resource="#Concept"/>
        <rdfs:domain rdf:resource="#CompositeConcept"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </rdfs:Property>
    <rdfs:Property rdf:ID="type">
        <rdfs:domain rdf:resource="#Concept"/>
        <rdfs:domain rdf:resource="#CompositeConcept"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </rdfs:Property>

    <rdfs:Property rdf:ID="relatedResource">
        <rdfs:domain rdf:resource="#Concept"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    </rdfs:Property>

    <rdfs:Property rdf:ID="Relationship">
        <rdfs:domain rdf:resource="#Concept"/>
        <rdfs:domain rdf:resource="#CompositeConcept"/>
        <rdfs:range rdf:resource="#Concept"/>
        <rdfs:range rdf:resource="#CompositeConcept"/>
    </rdfs:Property>

</rdf:RDF>
```

Figure 5. RDF schema of the Conceptual Meta-model.

The RDF Schema in Figure 5 does not describe the concrete design of any particular web-based application. It provides the basis for the formalization of conceptual model descriptions by providing the appropriate elements and their interconnections. Particular designs of educational applications constitute schemata that are based on the aforementioned one, being its refinements. In these designs new classes can be defined, based on the primitive classes of the generic Conceptual Model schema and new properties and associations can be assigned to these classes. This is slightly different from the UML approach, where specific designs are considered as *instances* of the classes of the meta-model and not its *subclasses* (refinements). Additionally, a design of a newer hypermedia application can be based on a previous one after being able to locate and reuse its ontology-based description. Furthermore, the same or different designer can provide different versions of an RDF-described conceptual model. Although not demonstrated in this article, these capabilities are provided by the inherent support for distributed knowledge representation provided by the semantic web technologies and by the evolvability of Web itself (Lee, 1998). In this way, the involvement of semantic web in representing specific designs leads to evolution rather than mere design reuse.

In Figure 6 the RDF Schema of the Conceptual Design for the Fire Safety Tutorial described in the previous section is illustrated. In this schema, the generic classes *Courseware*, *Concept* and *CompositeConcept* are refined into application-specific classes such as *FireSafetyCourseware*, *PretestConcept*, *WaterConcept*, etc. For some of these application-specific classes, new properties are defined. For example, for the *PretestConcept* class a property named “grade” is defined, which expresses the grade that a student is assigned after taking a preliminary test for the Fire Safety tutorial.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY xsd "http://www.w3.org/2000/10/XMLSchema#">
    <!ENTITY conc "http://www.softlab.ntua.gr/andpapas/cadmos/conceptual#">]>
<rdf:RDF

    xml:base  ="http://www.softlab.ntua.gr/andpapas/cadmos/FireSafety"
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:conc ="&conc;"
    xmlns:xsd ="&xsd;">

    <rdfs:Class rdf:ID="FireSafetyCourseware">
        <rdfs:subClassOf rdf:resource="&conc;Courseware" />
    </rdfs:Class>
    <rdfs:Class rdf:ID="IntroductionConcept">
        <rdfs:subClassOf rdf:resource="&conc;Concept" />
    </rdfs:Class>
    <rdfs:Class rdf:ID="PretestConcept">
        <rdfs:subClassOf rdf:resource="&conc;Concept" />
    </rdfs:Class>
    <rdfs:Class rdf:ID="TypesofExtinguishersCConcept">
        <rdfs:subClassOf rdf:resource="&conc;CompositeConcept"/>
    </rdfs:Class>
```

```
    <rdfs:Class rdf:ID="WaterConcept">
        <rdfs:subClassOf rdf:resource="&conc;Concept" />
    </rdfs:Class>
    <rdfs:Class rdf:ID="CO2Concept">
        <rdfs:subClassOf rdf:resource="&conc;Concept" />
    </rdfs:Class>
    <rdfs:Class rdf:ID="DryChemicalConcept">
        <rdfs:subClassOf rdf:resource="&conc;Concept" />
    </rdfs:Class>

    <rdf:Property rdf:ID="grade">
        <rdfs:domain rdf:resource="#PretestConcept" />
        <rdfs:range  rdf:resource="&xsd;integer" />
    </rdf:Property>
    <rdf:Property rdf:ID="ready_to_read">
        <rdfs:domain rdf:resource="#WaterConcept" />
        <rdfs:domain rdf:resource="#CO2Concept" />
        <rdfs:domain rdf:resource="#DryChemicalConcept" />
        <rdfs:range  rdf:resource="&xsd;boolean" />
    </rdf:Property>
    <rdf:Property rdf:ID="isTested">
        <rdfs:domain rdf:resource="#WaterConcept" />
        <rdfs:domain rdf:resource="#PretestConcept" />
    </rdf:Property>

</rdf:RDF>
```

Figure 6. RDF Schema of the Fire Safety Tutorial.

The RDF Schema of the Conceptual Design of a particular application, like the one illustrated in **Figure 6**, can be automatically generated from UML models. These models, created using appropriate CASE tools and described in XMI format, can be easily transformed to RDF Schema descriptions using XSL Transformations (Clark, 1999).

The schema for the Fire Safety Tutorial provides the basic concepts and relationships for the educational application under consideration. Thus, it specifies the instance space for this application. An instance of it contains elements defined in the schema having appropriate values in their properties. An excerpt of such an instance is depicted in Figure 7. Note that while the namespace prefix for the generic conceptual schema is "conc", the namespace prefix for the elements of the Fire Safety Tutorial is "fireEx".

```
    <!-- ... -->
    <!DOCTYPE rdf:RDF [
    <!ENTITY xsd "http://www.w3.org/2000/10/XMLSchema#">
    <!ENTITY fireEx "http://www.softlab.ntua.gr/andpapas/cadmos/FireSafety#">]>
    <!-- ... -->

    <fireEx:FireSafetyCourseware rdf:ID="#FireSafety">
        <conc:name>Fire Safety Tutorial</conc:name>
        <dcterms:hasPart>
            <rdf:Bag>
                <rdf:li rdf:resource="#Intro"/>
                <rdf:li rdf:resource="#Pretest"/>
                <rdf:li rdf:resource="#types"/>
            </rdf:Bag>
        </dcterms:hasPart>
    </fireEx:FireSafetyCourseware>
    <!-- ... -->
    <fireEx:PretestConcept rdf:ID="#Pretest">
    <conc:name>Pretest Knowledge on Fire Safety</conc:name>
    <conc:type>Assessment</conc:type>
        <fireEx:grade rdf:datatype="&xsd;integer">0</fireEx:grade>
        <conc:relatedResource rdf:resource="#pretest_html"/>
    </fireEx:PretestConcept>

    <fireEx:TypesofExtinguishersCConcept rdf:ID="#types">
        <conc:name>Present the Types of Fire Extinguishers</conc:name>
        <dcterms:hasPart>
            <rdf:Bag>
                <rdf:li rdf:resource="#Water"/>
                <rdf:li rdf:resource="#CO2"/>
                <rdf:li rdf:resource="#Dry"/>
            </rdf:Bag>
        </dcterms:hasPart>
    </fireEx:TypesofExtinguishersCConcept>
```

```
    <fireEx:WaterConcept rdf:ID="#Water">
        <conc:name>Water Fire Extinguishers</conc:name>
        <fireEx:ready_to_read rdf:datatype="&xsd;boolean">true</fireEx:ready_to_read>
        <fireEx:isTested rdf:resource="#Pretest">
        <conc:relatedResource resource="#APW"/>
    </fireEx:WaterConcept>

    <!-- ... -->

    <rdf:Description rdf:about="#APW">
        <dc:title xml:lang="en">
             Water Fire Extinguishers
        </dc:title>
        <dc:format>
            <dcterms:IMT>
                <rdf:value>application/x-shockwave-flash</rdf:value>
            </dcterms:IMT>
        </dc:format>
    <lom-tech:location
resource="http://www.softlab.ntua.gr/andpapas/cadmos/FireSafetyDemo#APW.dcr"/>
        <lom-tech:otherPlatformRequirements xml:lang="en">Shockwave</lom-
tech:otherPlatformRequirements>

<!-- ... -->
```

Figure 7 The RDF description of the Fire Safety Tutorial Conceptual Model (excerpt).

The aggregation of concepts, that is, the containment of concepts by others, is achieved by the use of the `dcterms:hasPart` element adopted by both the Dublin Core and IEEE metadata RDF descriptions. These descriptions were adopted in attempt to reuse existing standards in the proposed conceptual model.

In the RDF description of the Fire Safety Tutorial exemplified above, the actual resources are expressed using an IEEE LOM metadata description as proposed in (Nilsson, 2002). This facilitates the easy storage, search and retrieval of specific resources incorporated into particular applications. In this way, a specific RDF description of an application contains not only information about the structure of the application (design reuse), but also information about the actual learning objects that it has reused.

Rules of adaptation

The Conceptual Model presented so far, in both its UML representation and its RDF encoding, corresponds to the Domain Model component of the Adaptive Hypermedia Applications Model (De Bra et al., 1999). This, among others, also identifies a Teaching Model as the component that defines the actual rules of adaptation in the form of if-then-else rules applied to specific properties of the concepts of the Domain Model. In addition to the UML description of the Conceptual Model, we define rules of adaptation that correspond to the Teaching Model, which are expressed using the Object Constraint Language (Warmer & Kleppe, 1999). More specifically, certain rules are expressed as *invariant* conditions, that is, logical expressions that must always evaluate to true in order for the model to be in a valid state. An example of an invariant condition is given below. This is applied to the “Pretest” concept in **Figure 4**, as it is determined by the keyword `context`.

```
context Pretest inv:
        this.grade = 5 implies water.ready_to_read = true
```

This condition states that if the performance of a student in the pretest is five out of five, expressed in an attribute of name `grade`, then the interaction concept “Water Fire Extinguishers” is considered as ready for study by the particular student. The two concepts are connected with an association in **Figure 4** and the word `water` expresses the *role* of the “Water Fire Extinguishers” in the given association. Note that this rule defines a specific instructional design strategy for teaching the specific topic of Fire Safety. This strategy is described abstractly and is not connected to any implementation concern, for example navigation or user interface. These concerns are dealt during later phases of design according to CADMOS-D.

In the rest of this section we present how the Teaching (or Rules) Model can be encoded by appropriately applying rules to our RDF-based conceptual description of Educational Adaptive Hypermedia. Again, these rules only express instructional decisions, that is, they only refer to teaching strategies concerning the specific domain, e.g. the sequence of activities to be performed by the learner given his/her performance assessment activities.

We propose the use of the Rule Markup Language, or RuleML, for the application of specific rules in the elements of the RDF encoded Conceptual Model. RuleML is an "XML-based markup language that permits Web-based rule storage, interchange, retrieval, and firing/application". This language is intended to provide rules on Semantic Web OWL and RDF ontologies.). We are currently developing a tool for the translation of OCL rules to RuleML to facilitate the automatic transformation of UML models to Semantic Web descriptions beside XSL Transformations.
The same rule expressed earlier in OCL is expressed in RuleML as follows:

```
<imp>
  <_head>
    <atom>
      <_opr><rel &fireSafety;#ready_to_read"</_opr>
      <var>concept</conc>
      <ind rdf:datatype="&xsd;boolean>true</ind>
    </atom>
  </_head>
  <_body>
    <and>
      <atom>
        <_opr><rel rdf:resource="http://&fireSafety;#WaterConcept"/></_opr>
        <var>concept</var>
      </atom>

      <atom>
        <_opr><rel rdf:resource="http://&fireSafety;#Pretest"/></_opr>
        <var>pretest</var>
      </atom>

    <atom>
        <_opr><rel rdf:resource="http://&fireSafety;#isTested"/></_opr>
        <var>concept</var>
        <var>pretest</var>
        </atom>

       <atom>
        <_opr><rel rdf:resource="http://&fireSafety;#grade"/></_opr>
        <var>pretest</var>
    <ind rdf:datatype="&xsd;integer>5</ind>
      </atom>
    </and>
  </_body>
</imp>
```

Figure 8. Expressing a rule using RuleML.

The syntax of RuleML is based on Horn clauses (Horn, 1951). The head *atom* is evaluated to "true" if all the atoms in the *body* section are true. Thus, the above expression means that "if variable *concept* is of type *WaterConcept*, variable *pretest* is of type *PretestConcept*, *concept* is tested by *pretest* and the grade of this concept has value *5*, then the *WaterConcept* property *ready_to_read* is *true"*. Note that here we only provide the syntactic basis of the rules involvement in creating adequate ontologies for describing adaptive educational applications. The semantic implications, e.g. the types of rules (event-condition-action rules, constraints, etc.) are considered to be handled by specific implementations and they are not treated here. This is only a semi-formal notation for the description of certain instructional design decisions that pertain to adaptivity.

Other examples of rules could be the update of the knowledge about learners represented in the User Model (Brusilovsky, 1996) of an Adaptive System, according to their history of interaction with the educational application or their performance in assessment activities.

Conclusions

In this paper we have presented how the outcomes of the Conceptual Design stage of a method for developing Educational Adaptive Hypermedia Applications can be encoded using RDF-based ontologies. Initially, the Conceptual Model is documented using the Unified Modeling Language, which is a more suitable representation for the human developers of EAHA. However, the proposed encoding of the Conceptual Model has certain benefits, such as reusability of design, formalization of the description and use of emerging technologies for the sharing of information based on the semantic web.

These descriptions can be automatically derived from existing UML models. However there is no need for a specialized UML tool for this purpose. UML models created by using existing CASE tools such as the IBM Rational Rose (http://www.rational.com) and described in XMI format can be easily transformed to RDF descriptions using XSL Transformations. Currently, we apply adaptation rules in UML models using OCL. We are developing a tool for the translation of OCL rules to RuleML to facilitate the automatic transformation of UML models to Semantic Web descriptions beside XSL Transformations.

References

Alexaki, S., Karvounarakis, G., Christophides, D., Plexousakis, D., & Tolle, K. (2001). The ICS-FORTH RDFSuite: Managing voluminous RDF description bases. *In Proceedings of the 2nd International Workshop on the Semantic Web*, Hongkong, May 1, 2001, 1-13.

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

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

Brickley, D., & Guha, R. V. (2003). *RDF vocabulary description language 1.0: RDF Schema*. Retrieved December 10, 2003, from http://www.w3.org/TR/rdf-schema/.

Chang, W. W. (1998). *A discussion of the relationship between RDF-Schema and UML*. Retrieved December 10, 2003 from http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804/.

Clark, J. (1999). *XSL Transformations (XSLT) version 1.0*. Retrieved December 10, 2003, from http://www.w3.org/TR/xslt.

De Bra, P., Houben, G., & Wu, H. (1999). AHAM: A Dexter-based reference model for adaptive hypermedia. In J. Westbomke, U. K. Wiil, J. J. Leggett, K. Tochtermann & J. M. Haake (Eds.), *Proceedings of the tenth ACM conference on hypertext and hypermedia: returning to our diverse roots,* Darmstadt, Germany, February 1999, NY: ACM Press.

Ford, P., Goodyear, P., Heseltine, R., Lewis, R., Darby, J., Graves, J., Sartorious, P., Harwood, D., & King, T. (1996). *Managing change in higher education: A learning environment architecture*, Buckingham: Open University Press.

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.

Horn, A. (1951). On sentences which are true of direct unions of algebras. *J. Symbolic Logic,* 16, 14-21.

IMS (2003). *IMS Simple Sequencing Specification. Retrieved* December 10, 2003, from http://www.imsglobal.org/simplesequencing/.

Isakowitz, T., Stohr, E., & Balasubramanian, P. (1995). RMM, A methodology for structured hypermedia design, *Communications of the ACM.* 38 (8), 34-48.

Lee, T. B. (1998). *Evolvability*. Retrieved December 10, 2003, from http://www.w3.org/DesignIssues/Evolution.html.

Lowe, D., & Hall, W. (1999). *Hypermedia & the web, an engineering approach,* West Sussex, England: John Wiley & Sons.

Manola, F., & Miller, E. (2003). *RDF primer*. Retrieved December 10, 2003, from http://www.w3.org/TR/rdf-primer/.

Microsoft Corporarion (2003). *LRN 3.0 toolkit.* Retrieved December 10, 2003, from http://www.microsoft.com/elearn/.

Nilsson, M. (2002). *IEEE learning object metadata RDF binding*. Retrieved December 10, 2003, from http://kmr.nada.kth.se/el/ims/metadata.html.

Object Management Book. (2003). OMG unified modeling language – version 1.5. Retrieved December 10, 2003, from http://www.omg.org/uml.

Papasalouros, A., & Retalis, S. (2002). Ob-AHEM: A UML-enabled model for adaptive educational hypermedia applications. *Interactive Educational Multimedia*, 4, 76-88.

Retalis, S., & Skordalakis, E. (2001). An approach for solving instructional problems using networked technologies. *Computers in Human Behavior*, 17 (4), 393-407.

Schwabe, D., & Rossi, G. (1995). The object-oriented hypermedia design model. *Communications of the ACM,* 38 (8), 45-46.

Schwabe, D., & Rossi, G. (1998). An object oriented approach to web-based application design. *Theory and Practice of Object Systems,* 4 (4), 207-225.

Smith, P. L., & Ragan, T. (1999). *Instructional design*. New York: John Wiley & Sons.

Sommervile, I. (1997). *Software engineering*, Harlow, England:Addison-Wesley.

Warmer, J., & Kleppe, A. (1999). *The object constraint language: Precise modeling with UML*, Reading, MA: Addison Wesley.