Supporting Adaptive Hypermedia Authors with Automated Content Indexing

Sergey Sosnovsky, Peter Brusilovsky, Michael Yudelson

University of Pittsburgh, School of Information Sciences
135 North Bellefield Avenue, Pittsburgh, PA 15260, USA
{sas15, peterb, mvy3}@pitt.edu

Abstract. The main hindrance to expanded use of adaptive educational hypermedia systems is the need for content to be properly described in the terms of domain concepts. This requirement slows down the authoring process and creates an obstacle to the broader distribution of such systems. In the current paper, we propose an approach to providing automated content indexing for adaptive educational hypermedia systems. Both stages of automated content indexing (content parsing and prerequisite/outcome identification) are described here in detail. The approach we have developed has been implemented by indexing the content of the NavEx system and has proven itself by creating meaningful recourse for adaptive example navigation support.

1 Introduction

More and more adaptive hypermedia systems [2] are reaching the point where they can be used in the context of real education, an area that is now almost exclusively served by traditional non-adaptive web-based educational (WBE) systems [4]. Thanks to years of research, the problems of representing the domain model, knowledge about the student, as well as development of the interface can now be solved in a number of domains by relatively small research teams. The choice of the Web as an implementation platform can help a small team solve problems of delivery, installation, and maintenance, expanding the systems availability to hundreds and thousands of students. Yet, there "the last barrier" exists: The traditional static, non-adaptive WBE systems and courses have something that almost no intelligent system developed by small research teams can offer – large amounts of diverse educational material. A high-quality traditional WBE course may have thousands of presentation pages, and hundreds of other fragments of learning material – examples, explanations, animations, and objective questions created by a team of developers. In comparison, the number of presentation items in the best adaptive web-based educational systems is well under two hundred and the number of other fragments of learning material, such as problems or questions is usually no more than a few dozen. These numbers are certainly sufficient for a serious research study of the system in classroom use, but the number is still small for the needs of practical web-based education, i.e., the ability to support reasonable fragments of practical courses which can be taught to large numbers of students, semester after semester.

We think that the key to solving this last problem is teacher-oriented authoring tools for providing the content for adaptive educational hypermedia systems. The pioneering paper (describing the PAT Algebra tutor [17]) provides a good analysis of problems and a set of design principles developed to use when authoring problems for a cognitive, rule-based tutoring system. The situation described in this paper is when the content to be created is really "intelligent content." The power of intelligent content is that knowledge is hidden behind every fragment of it. Even the simplest "presentation" fragments of external content should be connected to the proper elements of domain knowledge (concepts) so that an intelligent educational system (IES) can understand what it is about, when it is reasonable to present it, and when it is premature. More complicated content format, such as examples and problems, require additional coding in order to enable an IES to run the example or to support the student's ongoing solution of a problem.

There are very few domains where the knowledge behind a fragment of educational content can be deduced automatically by the system from the problem statement. For example, straightforward representations might be used in an IES which teaches derivation in calculus [6], expression evaluation in C [5], or equation-solving in algebra [16]. In these domains it is quite easy to identify rules or concepts behind a problem or an example and no other knowledge except core domain knowledge is required to support the problem solving process. In these "lucky" domains, authoring of additional problems or examples is easy; the author only needs to provide the problem statement in a traditional form. Yet, even in such simple domains, teacher involvement is required for the advanced hypermedia systems to distinguish multiple-concept sequences, such as those including prerequisites or outcomes [3]. In less formalized domains, the knowledge behind a content fragment can't be easily extracted and has to be provided during the authoring process, such as in adaptive hypermedia systems like KBS-Hyperbook [12] or SIGUE [10].

In this paper, we present a collaborative approach to authoring intelligent content for adaptive hypermedia. In our approach the work is distributed between an intelligent authoring system and a teacher. The teacher informs the system about his or her preferred way of teaching by grouping prepared content into a sequence of topics or lecture sets. An intelligent authoring system extracts concepts from the provided fragments of content and classifies them as either prerequisite or outcome concepts on the basis of the teacher's preferred method of teaching. We have applied this approach in our recent system, NavEx.

2. Lack of Authoring Support in the Context of a Programming Course

We have used a number of different web-based educational systems in the context of an undergraduate course "Introduction to Programming" being taught at the University of Pittsburgh's School of Information Sciences. The results of every semester's evaluation and students' feedback led us to assume that the pedagogocal value of at least two of them could be increased by providing a system with metaknowledge about its content, thus an implementation of adaptive hypermedia technologies. Two of these systems are briefly described below.

The WebEx system serves out interactive, explained examples of programming solutions, via the Web. An author of an examp le or a later teacher can provide textual explanations for every line of the program code. The students can browse these comments at their own pace and order by selectively clicking on the commented lines (see Fig. 1). The first version of WebEx has been implemented and was reported on in 2001 [9]. Since that time, WebEx has been heavily used. Each semester it has offered an incrementally larger subset of examples from the classroom lectures. The results of evaluation demonstrated that a solid fraction of students wanted to see more explained examples than just the examples from the lecture. Moreover, the proportion of students who wanted "more examples" was growing, even though we were incrementally making more and more lecture examples available each semester. Typically, our course has grown to about 60 examples. This is a relatively large amount, but students have no trouble finding the relevant examples because each example is linked to a specific lecture. In contrast, examples from another class or a digital library are a navigational burden to the student. With no clear guidelines as to which of these examples should be accessed and when, students may easily choose examples that are either too complicated or too simple for the student's current progress.

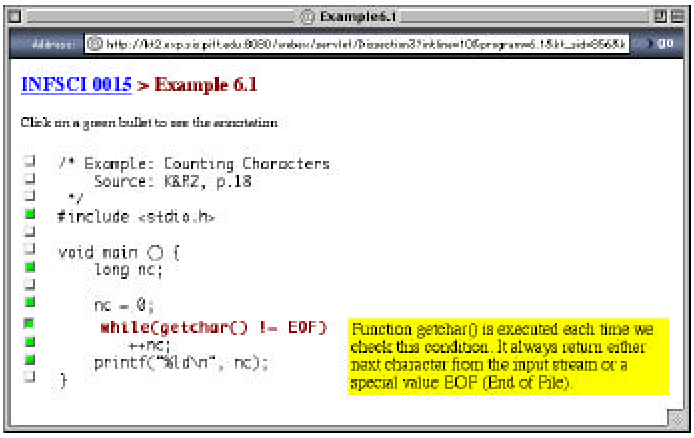

Fig. 1. Interactive example in WebEx.

Another application is QuizPACK. This system authorizes and delivers parameterized web-based quizzes for C-programming courses. The detailed description of QuizPACK can be found in the example for [15]. Figure 2 demonstrates the student interface of the system. QuizPACK evaluation and individual feedback, though positive, showed that students suffer from lack of guidance. Since QuizPACK is used mostly for self-assessment, students need to estimate what their weakest topics are and then decide what quiz is necessary to take, on their own. However, they often get lost "in space" containing about 50 quizzes, which was necessary to develop in order to cover all of the course material. To assist students system needs to provide a valuable feedback which helps them to locate themselves in the course-knowledge space. Need in adaptive navigation support leads to the necessity of creating "intelligent content."

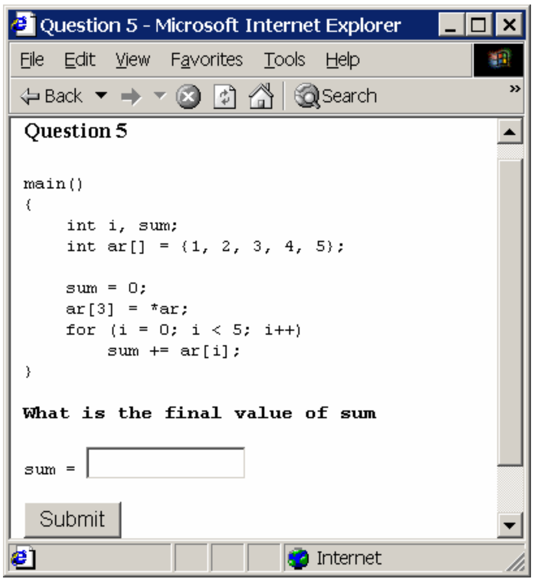

Fig. 2. Typical question in QuizPACK.

3. Automated Content Indexing

There are no universally-accepted recommendations as to which level is best to use when defining a concept in computer programming domains. Some authors suppose that it has to be done on the level of programming patterns or plans [13], other believe, that the concepts is to be closer to the elementary operators [1]. According to the first point of view, the notion of pattern is closer to the real goal of learning programming, since the patterns are what programmers really use. However, the second way is more straightforward and makes the burden of indexing more feasible. With the notable exception of ELM-PE [18], all adaptive sequencing systems known to us work with operator-level concepts. Current implementation of the proposed indexing algorithm also uses the operator-level approach.

This algorithm has two main stages. In the first stage, it extracts concepts from the content elements (examples, questions, presentation pages) combined by the teacher into activity pool. For example, all WebEx examples form one pool; all QuizPACK quizzes belong to another pool. In the second stage, the prerequisite/outcome structure of the course is built in terms of concepts, describing content elements. This sequential structure, along with the indexed content, provides the basis for adaptation. The following two subsections explain both stages in detail.

3.1. On-line Content Parsing

Traditionally, the extraction of grammatically meaningful structures from textual content and the determination of concepts on that basis is a task for the special class of programs called parsers. In our case, we have developed the parsing component with the help of the well-known UNIX utilities: lex and yacc. This component processed source code of a C program and generates a list of concepts used in the program. Currently, fifty-one concepts have been determined for the subset of C language studied during our course. Each programming structure in the parsed content is indexed by one or more concepts, depending upon the amount of knowledge students need to have learned in order to understand the structure. For instance, the parser generates the following list of concepts for the program code in Fig. 1, the WebEx example:

> *include, void, main_func, decl_var, long, decl_var, assign, ne_expr, pre_inc, while, compl_printf*

It is necessary to mention that each concept here represents not simply a keyword, found in the code, but a grammatically complete programming structure. For instance, concept *while* is recognized by the parser only after the whole while-loop, including the keyword *while*, the iteration condition and the loop body, is found. This is why the concept *while* in the index list above must come after the concepts *ne_expr* (not-equal expression: *!=*) and *pre_inc* (pre-increment: *++identifier*).

The parsed code is accessed through the http-protocol. It can be represented as a simple source file in text format or as a properly formatted HTML-file, which distinguishes the code samples from the rest of the HTML content with one of these commonly accepted tag pairs: <code> – </code>, <pre> – </pre>, or <tt> – </tt>. Usage of other HTML-tags, such as the nesting of <b>, <a>, <div>, etc. between the pair of code-tags does not influence the result of the parsing. These tags are simply ignored; at the same time an HTML escape-sequences (like   < & etc.) are processed and converted into corresponding symbols or symbol sequences. For example, Table 1 demonstrates two code samples, which present alternative ways to format HTML code for the QuizPACK question showed in Fig. 2. The parsing component processes all three samples (including the pure C source code of QuizPACK question) in the same way and generates for them the same list of concepts:

> *main_func, int, decl_var, decl_var, int, decl_array, init, assign, assign, derefer, assign, l_expr, post_inc, add_assign, for*

Hence, the web-parser we have developed could be used for indexing the great amount of created on-line C content, such as code libraries and web-based tutorials. Given the URL of the content resource it generates list of concepts, extracted from the C code used in this resource.

Table 1. Two alternative html formats for a simple C program.

```
<html>
…
<code>
main()
{
  <b>int</b> i, sum;
  <b>int</b> ar[] = {1,2,3,4,5};

  sum = 0;
  ar[1] = *ar;
  <b>for</b> (i = 0; i < 5; i++)
    sum += ar[i];
}
</code>
…
</html>
```

```
<html>
…
<pre>
main()<br>
{<br>  
int i, sum; <br>  
int ar[] = {1,2,3,4,5};
<br><br>  
sum = 0; <br>  
ar[1] = *ar; <br>  
<a href = "for.html">
for</a>(i = 0; i &lt; 5; i++)
<br>    
sum += ar[i];<br>
}<br>
</pre>
…
</html>
```

3.2. Prerequisite/Outcome Identification

The outcomes of the parsing stage are index lists for all content elements. However, it is not enough for our purposes, since these lists are not yet connected to each other and the content still does not possess any structure on which we could base the adaptation process. In the next stage, all concepts related to each content element are divided into prerequisite and outcome concepts. Prerequisites are the concepts that the student needs to master before starting to work with the current element. Outcomes denote concepts that the element leads one toward learning, i.e. the learning goals of the element.

We use an original algorithm for the automatic identification of outcome concepts (see Fig. 3). This algorithm is flexible enough to be influenced by an instructor-specific way of teaching the course. The source of knowledge for this algorithm is a sequence of groups of content elements. Each group is formed by the elements introduced in the same lecture. Groups are ordered according to the order of lectures in the course, forming a sequential structure of the learning goals of the course. The prerequisite/outcome division algorithm is based on the following assumptions:

- While analyzing content element from some lecture, concepts corresponding to this element and introduced in a preceding lecture are considered to be already learned.
- In each content element, all concepts introduced in the previous lectures are considered as prerequisites, while the concepts first introduced in the current lecture are regarded as outcomes.
- The set of new concepts found in all content elements associated with the lecture becomes the learning goal of the lecture.

The direct outcome of this algorithm is a fully-indexed set of content elements belonging to each lecture and a sequence of learning goals associated with the lectures. This sequence represents the specific approach to teaching C-programming that is employed by this specific instructor [3]. Once the content elements are indexed and the goal sequence is constructed, any future additional element can be properly indexed by the algorithm and even associated with a specific lecture in the course. More precisely, an association with a specific lecture is the first step in this process. The element is associated with the last lecture that introduces its concepts (i.e., the latest lecture, whose learning goal contains least one concept belonging to this element's index). After that, the element is associated with this lecture. It is important to stress that the outcome identification is adapted to a specific way of teaching a course "mined" from the original sequence of content elements. It is known that different instructors teaching the same programming course may use a very different order for their concept presentation [14]. Naturally, content sequencing in a course is to be adapted to the instructor's way of teaching.

```
learnt_concepts = ∅
for i = 1 to no_of_chapters
{
  for j = 1 to chapter[i].no_of_examples
  {
    chapter[i].example[j].prereq = learnt_concepts ∩ chapter[i].example[j].all_concepts
    chapter[i].example[j].outcome = chapter[i].example[j].all_concepts \ learnt_concepts
  }
  for j = 1 to chapter[i].no_of_examples
    learnt_concepts = learnt_concepts ∪ chapter[i].example[j].all_concepts
}
```

Fig. 3: Pseudo-code for prerequisite/outcome identification.

Although, the described indexing approach, using a parsing component, is specifically for programming and for the learning content based on the programming code (questions, code examples), we believe that the proposed general idea is applicable for a broad class of domains and types of content. In less formalized conditions, where concepts do not have a salient grammatical structure, the classic information retrieval approach could be used instead of parsing.

Currently, the described approach is implemented in the NavEx system, which provides adaptive annotations for programming code examples. The indexed content elements in NavEx are the same programming examples used in WebEx (see Fig. 1). Preliminary evaluation shows that implemented indexing algorithm along with the mechanism for building inter-concept hierarchies from the given, flat content provides meaningful recourse for adaptive example-navigation support in NavEx. The NavEx mechanism of adaptation as well as the system interface will be briefly described in the following section.

4. Adaptive Navigational Support in NavEx

The interactive window of the NavEx system is divided into 3 frames (see Fig. 4). The leftmost frame contains a list of links to all examples/dissections available to a student in the current course. The links are annotated with colored icons. A red bullet means that the student has not mastered enough prerequisite concepts to view the example. The link annotated with the red bullet is thus disabled. A green bullet means that the student has enough knowledge to view the example. A green check mark denotes that the example has already been seen by the student. A green “play” bullet means that this line of code in the example is currently being viewed. The order of links to examples is fixed, so that students can find them in the same place, no matter what the student’s progress through the course has been.

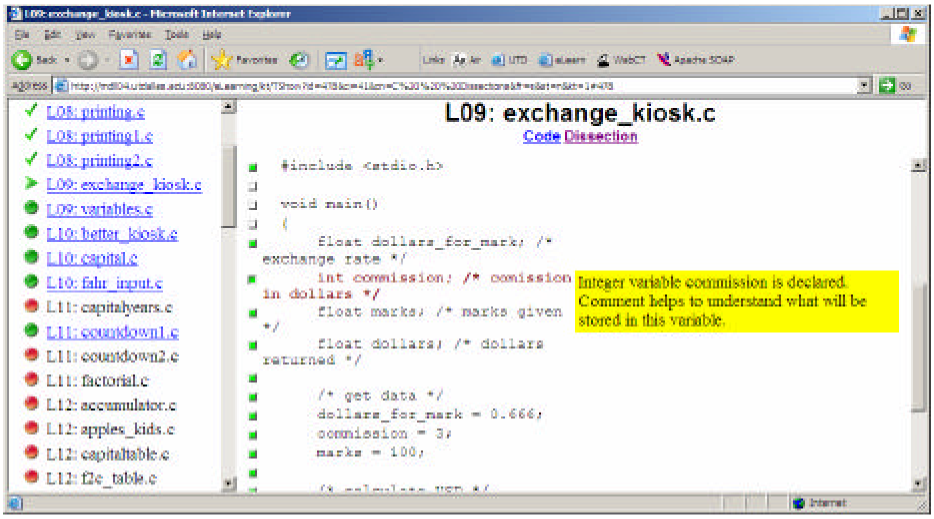

Fig. 4: The interface of NavEx.

The central frame displays the name of the current example. Underneath it are two links: one loads the source code of the example into the central frame (where it will be copied, compiled, and explored); the other link loads interactive example dissection (served directly by the WebEx system). Dissection means that one takes the original source code and comments on it. These comments address the meaning and purpose of this line of code and help the student to understand the example better. Extended comments are shown to the left of the code and can be activated by clicking on the bullet next to the line of the code. If the comment is available, the bullet is green; otherwise, it is white.

NavEx is considered a value-added service of the KnowledgeTree architecture [8], and implements several common protocols including student modeling and transparent authentication. As a typical value-added service, NavEx stands between e-Learning portals and reusable content elements and provides additional value for both teachers and students who use this content through the portal.

Adaptive navigational support in NavEx is done on the basis of the overlay student model [11]. Student's knowledge is represented as a binary vector k, where $k_i = 1$ means that the student has successfully mastered concept i, and $k_i = 0$ means the opposite.

When the student logs into the system a new session is created and information about the current state of his/her knowledge is retrieved from the student model. This information contains concepts the student has mastered and a list of examples the student has reviewed.

Knowing the student's knowledge of each domain concept and the prerequisite-outcome profile of examples, the sequencing mechanism can dynamically compute the current educational status for each example, which is then presented to the student in the leftmost frame of the system window (Fig. 4) in the form of adaptive annotations with the colored icons.

When the student clicks on an example link and reviews the example code, the outcome concepts of the example change their state to “learned” ($k_i = 1$). The changes in the knowledge state of the student are then propagated to the student model. The availability of each new example is determined by checking whether any of the previously unavailable examples now have all of their prerequisite concepts mastered. As the student reviews examples, newer examples become available. The knowledge-based adaptive annotation approach used in NavEx is a variation of a popular adaptive annotation approach introduced originally in the ISIS-Tutor system [7].

5. Summary and Future Work

We have discussed the development of an approach for the automated indexing of content based on C programming code. The first stage of this approach is performed by the implemented parsing component, which is able to extract C concepts from on-line content formatted as HTML or as pure C code. Hence, this tool could be used for indexing the great amount of created on-line C code which is contained in on-line libraries and web-based tutorials.

The second stage is the prerequisite/outcome identification and, building on its base, the inter-concept hierarchical structure of the content. The outcome we receive is the instructor-adaptive structure of the course reflecting his/her way of teaching the course.

The designed approach has been implemented for the NavEx system development. NavEx content being indexed consists of the programming code examples. Preliminary evaluation shows that the implemented indexing algorithm along with the mechanism of building inter-concept hierarchies of the content provide meaningful recourse for adaptive example navigation support in NavEx.

Our next goal is to build-in this algorithm as part of the authoring interface for the next adaptive version of QuizPACK system. Since every QuizPACK question is simply a small C program, we expect our algorithm to work successfully for this application and considerably facilitate the authoring process in QuizPACK.

References

1. Barr A., Beard, M., & Atkinson, R. C.: The computer as tutorial laboratory: the Stanford BIP project. International Journal on the Man-Machine Studies 8, 5 (1976) 567-596.

2. Brusilovsky, P.: Adaptive hypermedia. User Modeling and User Adapted Interaction 11, 1/2 (2001) 87-110, available online at http://www.wkap.nl/oasis.htm/270983
3. Brusilovsky, P.: Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In: Murray, Blessing and Ainsworth (eds.): Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software. Ablex, Norwood (2003) 377-409
4. Brusilovsky, P. and Miller, P.: Course Delivery Systems for the Virtual University. In: Tschang, T. and Della Senta, T. (eds.): Access to Knowledge: New Information Technologies and the Emergence of the Virtual University. Elsevier Science, Amsterdam (2001) 167-206.
5. Brusilovsky, P. and Su, H.-D.: Adaptive Visualization Component of a Distributed Web-based Adaptive Educational System. In: Intelligent Tutoring Systems. Vol. 2363. Springer-Verlag, Berlin (2002) 229-238
6. Brusilovsky, V.: Task sequencing in an intelligent learning environment for calculus. In: Proc. of Seventh International PEG Conference, Edinburgh (1993) 57-62
7. Brusilovsky, P. & Pesin, L.: Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. Journal of Computing and Information Technology 6, 1 (1998) 27-38.
8. Brusilovsky, P. & Nijhawan, H.: A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities. In: Driscoll, M. and Reeves, T. C. (eds.) Proc. of E-Learn 2002, Montreal, Canada, AACE (2002) 154-161.
9. Brusilovsky, P.: WebEx: Learning from examples in a programming course. In: Fowler, W. and Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet, Orlando, FL, AACE (2001) 124-129.
10. Carmona, C., Bueno, D., Guzman, E., and Conejo, R.: SIGUE: Making Web Courses Adaptive. In: De Bra, P., Brusilovsky, P. and Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002) Proceedings, Málaga, Spain (2002) 376-379
11. Greer, J. & McCalla, G. (eds.): Student modelling: the key to individualized knowledge-based instruction. NATO ASI Series F, Vol. 125, Springer-Verlag, Berlin (1993).
12. Henze, N. and Nejdl, W.: Adaptation in open corpus hypermedia. International Journal of Artificial Intelligence in Education 12, 4 (2001) 325-350, available online at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/henze.html
13. Lutz, R.: Plan diagrams as the basis for understanding and debugging pascal programs. In: Eisenstadt, M., Keane, M. T. and Rajan, T. (eds.): Novice programming environments. Explorations in Human-Computer Interaction and Artificial Intelligence. Lawrence Erlbaum Associates, Hove (1992) 243-285.
14. Moffatt, D. V. & Moffatt, P. B.: Eighteen pascal texts: An objective comparison. ACM SIGCSE bulletin 14, 2 (1982) 2-10.
15. Pathak, S., Brusilovsky, P.: Assessing Student Programming Knowledge with Web-based Dynamic Parameterized Quizzes. In Proceedings of ED-MEDIA'2002, Denver, Colorado, USA: AACE, 1548-1553.
16. Ritter, S. and Anderson, J. R.: Calculation and strategy in the equation solving tutor. In: Moore, J. D. and Lehman, J. F. (eds.) Proc. of the Seventeenth Annual Conference of the Cognitive Science Society, Hillsdale, NJ, Erlbaum (1995) 413-418
17. Ritter, S., Anderson, J., Cytrynowicz, M., and Medvedeva, O.: Authoring Content in the PAT Algebra Tutor. Journal of Interactive Media in Education 98, 9 (1998), available online at http://www-jime.open.ac.uk/98/9/
18. Weber, G. & Bögelsack, A.: Representation of programming episodes in the ELM model. In: Wender, K. F., Schmalhofer, F. and Böcker, H.-D. (eds.): Cognition and Computer Programming. Ablex, Norwood, NJ (1995) 1-26.

Developing Active Learning Experiences for Adaptive Personalised eLearning

Declan Dagger, Vincent P. Wade, Owen Conlan

Knowledge and Data Engineering Group, Department of Computer Science, Trinity College Dublin, Ireland
{Declan.Dagger, Vincent.Wade, Owen.Conlan}@cs.tcd.ie
http://kdeg.cs.tcd.ie

Abstract. A pedagogical underpinning is a fundamental requirement for any learning activity or event. Many personalized eLearning systems, however, focus primarily on adaptive content retrieval based on a user profile and neglect the pedagogical requirements of educational systems. This form of intelligent search and insertion is effective at assembling personalized manuals, but not necessarily suitable for creating pedagogically sound eLearning offerings. This paper introduces a methodology for developing adaptive personalized eLearning experiences and the Adaptive Course Construction Toolkit, a tool built on this methodology, for creating pedagogically-based personalized eLearning solutions.

1 Introduction

Typically the approach to adaptive eLearning is to perform adaptivity of the content retrieval mechanism based on the user's personal preferences, prior knowledge, etc. The pedagogical considerations in terms of presentation, structure and narrative in some cases is completely absent and in others very weakly applied. In such adaptive eLearning courses where pedagogy exists it is inherently embedded in the content itself making it difficult to reuse or apply different pedagogies across the same adaptive content. This level of inflexibility leads to the development of pedagogically static and restricted courses.

However, eLearning research in the past 10 years has had one constant unambiguous finding; that pedagogy is absolutely fundamental to the success of a course and must be considered first and foremost when developing learning experiences [1]. The adaptivity should therefore support the appropriate selection of pedagogical strategy(s) and then apply adaptivity to the content and activities within the scope of the selected strategy(s). With today's courses built on Adaptive Hypermedia Systems (AHS) it could be argued that pedagogy is less directive in ensuring the overall experience of the course and much greater effort seems to be expended on accurate selection of subject areas.

This paper argues that next generation adaptive eLearning systems will increase their effectiveness when the adaptivity is applied to the selection of an appropriate (personalized) pedagogy and to the activities, communication and content within pedagogical elements of the course. This leads to adaptive pedagogically-driven eLearning where the pedagogy is central to the learning experience and not accidental and where the power of adaptivity enhances the effectiveness of the pedagogy.

2 Adaptive Course Construction Methodology

During the development of an adaptive online learning experience, several key processes must be realized as illustrated in Fig. 1. Firstly we must identify the goals and objectives of the course. Theses goals and objectives will form the base requirements and initial evaluation scale of the evaluation process. When we are happy with the specified course goals and objectives we must identify and select the appropriate pedagogical strategy(s) for the course based on the goals and objectives. Next we must model the knowledge domain within which the course will reside. Once the subject matter area is recognized and described it can be applied to the chosen pedagogical strategy(s). The next logical phase is to begin the content selection and grouping process whereby the appropriate learning resources are grouped with the pedagogical elements. The customized pedagogical strategy can now be made adaptive by applying the appropriate adaptivity to the pedagogical structure based on the goals and objectives of the course, the selected content of the course and the pedagogical strategy(s) of the course. The next phase of the development process is to test the semantics and the functionality of the adaptive course by verification through the Adaptive Personalized eLearning Service (APeLS). This cyclical approach supports an expandable course construction process.

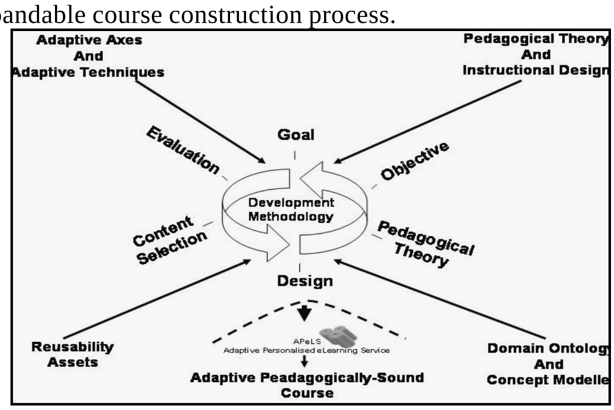

Fig. 1. A Sample Adaptive Course Construction Methodology

3 Creating Adaptive Courses with ACCT

The Adaptive Course Construction Toolkit (ACCT) was developed to address the complex and time-consuming nature of adaptive course construction methodology. The ACCT supplies the pedagogical foundation of an adaptive course developer-support framework. The design-time nature of the ACCT allows the course developer to construct customized domain ontologies to represent the subject matter expert's view of the knowledge domain. It facilitates the design of customized course narratives (the embodiment of pedagogy, abstract course overview and adaptive axes as applied) based on sound pedagogical models. Adaptivity models allow the course developer to create customized adaptive pedagogy through a support oriented drag and drop type association mechanism. The course developer can search for and select learning resources based on keywords, types, modes, contextualized prior usage, etc. The ACCT provides a course verification service, allowing the pedagogically-driven adaptive course to be viewed and verified prior to publication. The ACCT provides a publication mechanism allowing the course developer to export and publish their pedagogically-formed adaptive course.

3.1 Describing the Knowledge Domain

An integral part of a course creation process is the representation of a knowledge domain. Knowledge domain representation allows the subject matter expert to model their understanding and experience of an information domain. The ACCT provides an environment where the course developer can describe the knowledge domain as a collection of abstract concepts with descriptions and usability guidelines and relationship definitions. The ACCT provides the course developer with a set of predefined relationships which can be applied to concept pairs or groupings. The ACCT allows the course developer to create new custom relationship definitions. This flexibility to describe and customize relationship definitions and subject matter concepts during the development of the domain ontology supports the course developer during the initial phase of course production, knowledge domain representation.

The ACCT provides an environment through which the course developer can graphically create, edit and distribute these knowledge domain representations. In this way, the ACCT actively promotes and supports the sharing of subject matter expertise between peer collaborators.

3.2 Developing an Adaptive Course

The key process of creating online adaptive learning experiences is the creation of pedagogically sound courses. By using the custom narrative editor the course developer can create a custom pedagogical strategy based on sound and expandable pedagogical models provided by the ACCT (fig.2). The course developer can choose, customize and create pedagogical elements based on the palette of tools provided by the ACCT (fig.2). The pedagogical models provided are supported by usability guidelines and best practice descriptions. As the course developer builds the custom pedagogical strategy for their online learning experience they can specify and assemble learning activities within the supported pedagogical structure. The ACCT provides an interface whereby the course developer can place concepts from their previously defined knowledge domain ontology into this customized pedagogical structure to form a pedagogically sound online course framework (fig.2). The ACCT allows the course developer to graphically assemble the concepts (both Narrative and Subject Matter) and learning activities of their course (fig.2).

The ACCT supports the course developer by allowing them to search and select learning resources from multiple remote learning resource repositories (fig.2). The ACCT graphically supports the selection of learning resources by allowing them to be dragged and dropped from the tools palette to the pedagogical course structure.

To make this pedagogical online course adaptive, the course developer must associate Narrative Attributes with the concepts and learning activities of the course (fig.2). By applying these Narrative Attributes to a pedagogical learning structure, the course developer is describing a requirement for certain aspects of the course to be delivered "adaptively". This ability to apply "adaptivity" to concepts and learning activities uses the approach of Candidacy [2]. When the learning resources are selected they form abstract candidate groups which the adaptive technique, chosen through the adaptive axes rendering process, uses during the reconciliation of the adaptive course.

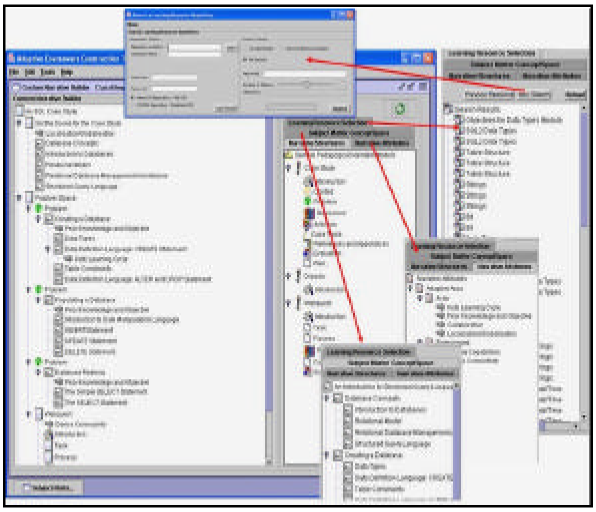

Fig. 2. ACCT Custom Narrative Builder

3.3 Publishing an Adaptive Course

One of the key advantages of the ACCT is its ability to allow the course developer to verify their adaptive course in real time. The ACCT provides a mechanism by which the course developer can test their adaptive course as a real web application. The ACCT exports a course package and application framework to APeLS (Adaptive Personalized eLearning Service). Included in the course package are the subject matter concept space, the custom narrative model with associated narrative attributes and all other related models, all transforms and class definitions required to run the adaptive course.

Through the ACCT application framework generically defined classes provide varying levels of interaction with respect to the learner and the teacher. It allows the course developer to view the learner model schema, the teacher domain scoping mechanism and the overall adaptive course structure. The produced adaptive course is independent of the content that may be used to render it. This allows for the rapid prototyping of adaptive course structures prior to the availability of the learning content. The ACCT produced course can also interact with physical content. Through the Candidacy architecture the ACCT produced course can adaptively render either concept descriptions or the concepts associated candidate learning resources.

4 Modeling Pedagogy

The ACCT is a pedagogy-driven course developer support environment. The pedagogy that is supplied by the ACCT forms the basis for fully customizable pedagogical strategies. The modeling approach involves the creation of XML models to represent the chosen pedagogical strategies, initially case-study, problem/enquiry, didactic and web-quest. The model contains descriptive information for each of the high level concepts/activities of the pedagogical strategy and suggests a possible sequencing of these pedagogical elements. Through the models the ACCT can provide guidelines on how to use the provided pedagogical strategies, how they might be extended and the types of adaptivity that might be applied. This modeling of pedagogy provides the course developer with a solid foundation on which they can create adaptive pedagogically-driven eLearning in a support-oriented environment.

4.1 Representing Pedagogical Models

Narrative Structures are created to describe how the pedagogical strategy(s) can be realized, e.g. defining types of activities, suggesting possible sequencing of activities, opportunities for communication and collaboration and content selection. They represent the (re)usable elements of pedagogical strategies in a model-based (XML) form. These models can be used as a pedagogically sound foundation on which the construction of adaptive pedagogically sound courses can be based.

The rapid construction of online courses consisting of different “flavors” of pedagogy is facilitated through the use of these Narrative Structures. For example, case based learning, web-quest learning, discovery-based learning and didactic based learning pedagogical models can be combined to form the basis of a customized blended pedagogy. This allows the potential course developer to create customized courses based on “flavors” of the modeled approaches thus actively promoting and facilitating the reuse of not only learning content but also the strategies and pedagogy behind the delivery of such learning experiences.

4.2 Describing Pedagogical Elements

Pedagogical strategies, typically, can be represented as a series of high-level descriptive concepts representing learning activities to be undertaken. Pedagogical strategies are usually accompanied by a set of guidelines and scenarios intended to strengthen the course developers' confidence in using the strategy(s).

Narrative Concepts facilitate the abstract description of pedagogical elements within a content-independent context. Narrative Concepts allow the pedagogical expert to create and customize elements of pedagogical strategies in the process of creating pedagogically-sound adaptive online learning experiences.

4.3 Enabling Adaptive Pedagogy

To make a customized pedagogy adaptive, Narrative Attributes, which consist of adaptive axes (prior knowledge, learning styles, etc.), adaptive techniques (object inclusion/exclusion, link annotation, etc.) and usability/guideline descriptions, must be applied. The course developer can decide on which adaptive technique(s) is to be applied based on the selected adaptive axes. For example, they could use either object inclusion/exclusion or link hiding in adaptivity based on prior knowledge depending on content granularity.

Through the modeling of adaptivity, Narrative Attributes representing adaptivity based on learner preference, tutor scope preference, learning context and learning device have been created.

5 Related Work

Current Adaptive Hypermedia (AH) systems and authoring tools for AH, in the educational domain, concentrate on developing and providing adaptive content retrieval and display capabilities. To this, adaptive content retrieval/delivery, elements of pedagogy are added in an effort to create online adaptive learning. For educationally effective adaptive eLearning however, the pedagogy must be the focus of development. Once the pedagogy has been customized (i.e. selected and extended if required) based on the subject matter area and learner goals, adaptivity can be applied to the pedagogically sound online course structure to produce adaptive personalized pedagogically-driven eLearning.

Currently, there are a range of tools available to create online pedagogy. For example, the REDEEM system [3] allows the teacher to create pedagogical online courses by describing the structure and flow of the content of the course and also the sequencing of the content. It allows the teacher to divide the course into sections and describe the content that the course will use. REDEEM has been quite successful in construction courses however it supports no elements of adaptivity and dynamic personalization. From an active learning perspective the LAMS system [4], which is built upon the emergent Learning Design standard (Previously Educational Markup Language EML), allows the teacher to create, describe and sequence learning activities. However, LAMS likewise provides no support for adaptivity of pedagogical structure and content selection.

Due to the complex and dynamic process of authoring Adaptive Hypermedia, the need for author support in creating adaptive pedagogically sound personalized eLearning is evident [5], [6]. The reach and effectiveness of adaptive personalized eLearning systems is also limited due to the cost of application development. The large initial setup cost of adaptive hypermedia is too high for the mass adoption of AHS in education. From current work in adaptive hypermedia [7], [8] in personalized eLearning it is evident that there are two areas of research which need future development, the design of pedagogically sound adaptive courses and the support offered to the course developer during the process of developing pedagogically sound adaptive courses. Pedagogy can be supported by specifying a requirements-based framework in which pedagogy can be described, used, reused and distributed in an effort to actively promote the cost reduction of adaptive course creation. The course developer can be supported by offering structural support and guideline support during the process of creating adaptive and non-adaptive courses.

Based on the state of the art in adaptive hypermedia and online pedagogy authoring, the ACCT will support and provide innovative ways of applying adaptivity to pedagogy to produce personalized eLearning.

6 Conclusion

From our experience with adaptive and personalized eLearning in the past, several key points should be noted. It is critically important that the teacher/tutor be empowered within the learning experience and not disenfranchised. As a blended teaching approach, the adaptive and personalized eLearning does not replace the teacher. It transforms and enriches their role in the learning experience. Adaptive personalized eLearning is a tool which the teacher/tutor can use to increase the potential educational effectiveness of the entire learning experience. Adaptive eLearning is not just about adaptive content retrieval and construction. It is a mixture of pedagogy, domain knowledge (subject matter expertise) and adaptivity. The production of personalized courses is realized at the concept level not content level. Specific content selection is an aspect of learning experience development that should not concern the teacher/tutor.

References

1. Laurillard, D. Rethinking University Teaching: A Framework for the Effective Use of Educational Technology. London: Routledge, 270pp, (1993)
2. Dagger, D.; Conlan, O.; Wade, V. An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources. E-Learn 2003, World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, Phoenix (2003) pp. 49-56 (2003)
3. Ainsworth, S.E., Grimshaw, S.K. & Underwood, D.J. Teachers implementing pedagogy through REDEEM Computers and Education, 33(2/3),171-188. ISSN 0360-1315 (1999)
4. Dalziel, J. Implementing Learning Design: The Learning Activity Management System (LAMS), ASCILITE (2003)
5. De Bra, P., Aerts, A., Berden, B., De Lange, B., Escape from the Tyranny of the Textbook: Adaptive Object Inclusion in AHA!. Proceedings of the AACE ELearn 2003 Conference, Phoenix, Arizona, November 2003, pp. 65-71.

6. Eklund, J. and Brusilovsky, P. (1999) InterBook: An Adaptive Tutoring System Universe Science News Vol. 12. March 1999. p. 8-13.
7. L. Aroyo, A. Cristea & D. Dicheva, (2003) A Layered Approach towards Domain Authoring Support, ICAIí02, Las Vegas, US
8. Apted, T and J Kay, (2002) Automatic Construction of Learning Ontologies, Aroyo, L and Dicheva, D, (eds) Proceedings ICCE Workshop on Concepts and Ontologies in Web-based Educational Systems, ICCE 2002, International Conference on Computers in Education

Yet Another Approach to Support the Design Process of Educational Adaptive Hypermedia Applications

Symeon Retalis [1] and Andreas Papasalouros [2]

[1]University of Piraeus, Department of Technology Education and Digital Systems, 80 Karaoli & Dimitriou 185 34 Piraeus, Greece
retal@unipi.gr
[2]National Technical University of Athens, Software Engineering Laboratory, 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.

1 Introduction

An Educational Adaptive Hypermedia Applications (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.

EAHAs are currently a 'hot' topic of research in the broader field of adaptive hypermedia applications and several AHES systems have been built during the past years [3], [4], [5], [9]. 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 [10], [13], [16], [17] 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 [13], 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. 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 [7].

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. The paper ends with some concluding remarks.

2 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 [13] 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 product design model can be decomposed into three sub-models: conceptual, navigational and user interface (presentation) models.

2.1 The Conceptual Model

The *Conceptual Model* defines the learning activities that will happen during the instructional process of a specific subject with their semantic interrelationships. The learning activities are applied to the various thematic concepts-topics of the domain. In fact the thematic topics should be considered as the Ontology of the subject domain 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. They have arranged that body of knowledge hierarchically, subdividing the field into areas, which are then broken down further into units and individual topics. An overview of the body of knowledge appears in

http://www.computer.org/education/cc2001/final/chapter05.htm.

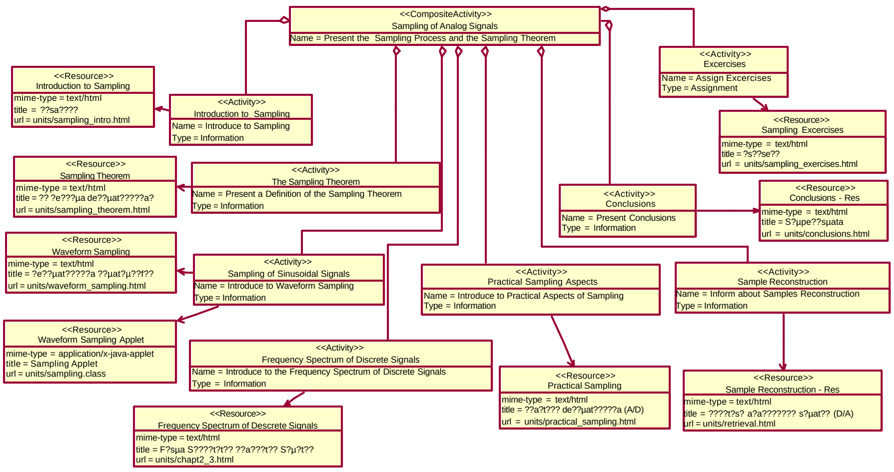

Fig. 1. Extract of the Conceptual Model of a course on digital signal processing which shows the relationships among learning activities and resources

Each learning activity is related to particular learning objectives, notions and terms to be taught, etc, according to the syllabus. 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. 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. For facilitating the construction of these diagrams, CADMOS-D has proposed an abstract object oriented meta-model, an instantiation of which is shown in Fig. 1 which concerns a hypermedia course on digital signal processing. The order of activities is defined by traversing the graph of activities from left to right in the conceptual model diagrams. A composite activity precedes its children, in the fashion of 'in-ordered' traversal of trees.

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.

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:

The Navigation Structure Model. This model defines the structure of the EAHA and defines the actual web pages and the resources contained in these pages. An example of this model is shown in Fig. 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
- *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 [10, 16] 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.

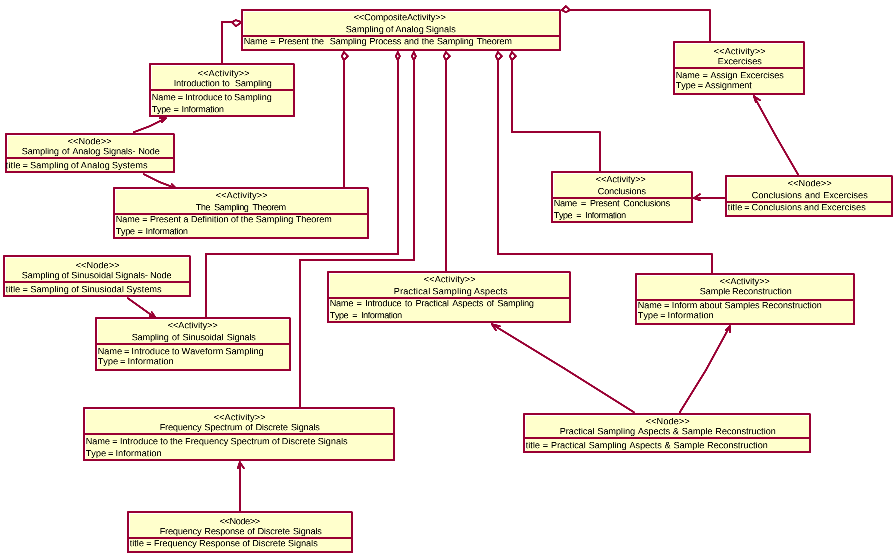

Fig. 2. An extract of the navigational structure model which shows the relationships of learning activities and hypermedia nodes

As shown in Fig. 2, a node will incorporate one or more learning activities. For example, a designer could decide 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. 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 [14], 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 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 EAHA, thus implementing content sequencing and adaptive navigation. An example of such a design model is shown in Fig. 3.

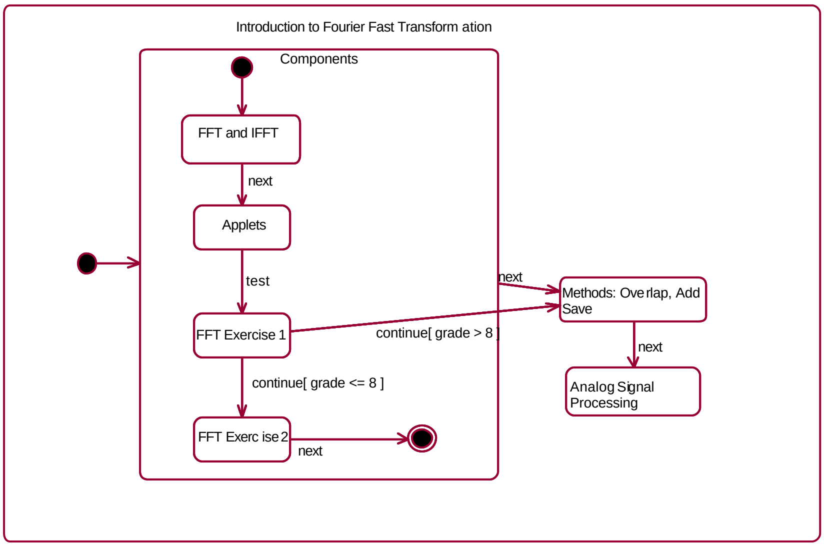

Fig. 3. Example of Navigation Behavior Model

2.3 The User Interface Model (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 is 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 web technology specifications elements such as HTML and CSS [http://www.w3.org] elements. More specifically, the Presentation Model contains the following stereotyped UML classes: “html”, that represent HTML elements, or aggregations of HTML ele ments and “css” that actually represent Cascading Style Sheet classes.

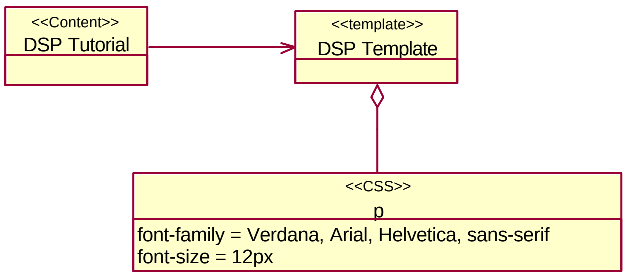

Fig. 4. Example of the basic templates that have been used in the DSP courseware

3. From the design models to automated generation of courseware

The design model of an AEHA can be created with CASE tools like the IBM Rational Rose tool. CADMOS-D suggests the utilization of such tools because UML models can be stored in XMI file, the OMG standard XML metadata interchange format files [15]. In that way, it is possible to process and manipulate that files 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 is transformed into a 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 AEHA.

The produced web 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.

An extract of the Content Packaging XML manifest file of the DSP coureware

```
<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>
```

This XML manifest file accompanied by the learning resources can be “uploaded” to any Learning Content Management Systems that support the IMS Content Packaging specification. We experimented with the SCORM Sample Run-Time Environment (RTE) version 1.3 and we produced a hypermedia application for a course on Digital Signal Processing.

At the moment, the CGA tool cannot make use of the information about the dynamic navigational behaviour, i.e. the state-transition diagrams. At the moment, the designer creates distinct design models per user type (users with different stereotype). Thus, we create an EAHA 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.

The lack of dynamic navigational structure is a limitation of our approach, but not an unsolved problem. Actually, it is a matter of time to produce the new release of the CGA tool. We are going to extend approaches like the one presented in [8]. More specifically, the UML activity diagrams will be transformed into IMS Simple Sequencing schema that will accompany the manifest content packaging file.

4. Conclusions

A design method like CADMOS-D, can be used as a framework [10] 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 by using the XML manifest file and the learning resources.

This work aspires the bridging of the gap between the conceptual description and the implementation of web applications as it is also suggested in [1]. Like approaches such as WebML [6], WCML [11], UWE [12], 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 current method 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 [8], which also uses the same model representation, XMI, and the same method for application generation, XSLT for adaptive applications. The main difference with this method is the provision for navigation and presentation issues, which is not covered in [8], and the support for Learning Technology Standards.

Acknowledgements

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

References

1. Aroyo, L., Dicheva, D. & Cristea, A.: Ontological Support for Web Courseware Authoring. In: Ceri, S., Guardères, G., Paraguaçu, F. (eds): Proceedings of Intelligent Tutoring Systems, 6th International Conference, ITS 2002, Biarritz, France and San Sebastian, Spain, June 2-7, 2002. Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag, Berlin Heidelberg New York (2002) 270–280
2. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide, Addison Wesley (1999)
3. De Bra, P., Houben. G., Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In: Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots, Darmstadt, Germany. ACM Press, New York (1999) 147–156
4. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User Adapted Interaction, **6 (**1996**)** 87–129
5. Brusilovsky, P.: Adaptive and Intelligent Technologies for Web-based Education. In: Rollinger, C., Peylo, C. (eds.): Special Issue on Intelligent Systems and Teleteaching. Kunstliche Intelligenz, **4** (1999) 19–25

6. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a modeling language for designing web sites. Computer Networks and ISDN Systems, **33** (2000) 13–157
7. Brusilovsky, P., Maybury, M.: From adaptive hypermedia to the adaptive web. Communications of the ACM, **45** (2002) 30–33.
8. Dolog, P., Nejdl, W.: Using UML and XMI for Generating Adaptive Navigation Sequences in Web-Based Systems. In: Stevens, P., Whittle, J., Booch, G. (eds.):UML 2003 - The Unified Modeling Language, Modeling Languages and Applications, 6th International Conference, San Francisco, CA, USA, October 20-24, 2003. Lecture Notes in Computer Science, Vol. 2863 Springer-Verlag, Berlin Heidelberg New York (2003)
9. Fischer, S.: Course and Exercise Sequencing Using Metadata in Adaptive Hypermedia Learning Systems. ACM Journal of Educational Resources in Computing, 1 (2001)
10. Garzotto, F., Schwabe, D., Paolini, P.: HDM- A Model Based Approach to Hypermedia Application Design. ACM Transactions on Information Systems, **11** (1993) 1–26
11. Gomez, J., Cachero, C., Pastor, O.: Conceptual modeling of device-independent Web applications. IEEE Multimedia, **8** (2001) 26–39
12. Hennicker, R., Koch, N.: Systematic Design of Web Applications with UML. In: Siau,K., Halpin, T., (eds.): Unified Modeling Language: Systems Analysis, Design and Development Issues, Idea Group Publishing, Hershey, PA, (2001)
13. Lowe, D., Hall, W.: Hypermedia & the Web, an Engineering Approach, John Wiley & Sons, West Sussex, England (1999)
14. De Oliveira, M., Turine, M., Masiero, P.:. A Statechart-Based Model for Hypermedia Applications, ACM Transactions on Information Systems, **19** (2001) 28–52
15. OMG. UML Version 1.4 Specification. Available at http://www.omg.org/technology/documents/formal/uml.htm
16. Rossi, G., Schwabe, D., Lucena, C., Cowan, D.: An Object-Oriented Model for Designing the Human-Computer Interface of Hypermedia Applications, In: Proceedings of the International Workshop on Hypermedia Design (IWHD'95), Montpellier, 5, Springer Verlag (1995)
17. Schwabe, D., Rossi, G.: The Object-Oriented Hypermedia Design Model. Communications of the ACM, **38** (1995) 45–46

Authoring and Delivering Adaptive Courseware

Craig Stewart[1], Alexandra Cristea[2], Adam Moore[1], Tim Brailsford[1] and Helen Ashman[1]

[1] School of Computer Science and Information Technology, University of Nottingham NG8 1BB, Nottingham, U.K.
{cds,axm,tjb,hla}@cs.nott.ac.uk
[2] Department of Computer Science Eindhoven University of Technology P.O. Box 513, Eindhoven, The Netherlands
a.i.cristea@tue.nl

Abstract. Adaptive Educational Hypermedia (AEH) may be the answer to the imperative need of personalization in web-based education. Up to now, however, adaptive hypermedia environments have focussed more on end-delivery, and less on the authoring problems. This paper addresses the complex issue of creation and delivery of personalized material by examining two aspects of existing AEHs: the authoring system of MOT, and the delivery system of WHURLE, with the aim of establishing a "write once, use many" methodology for the creation of content and adaptation within current AEH environments.

1 Introduction

Personalisation of presentation has always been important in any consumer industry. With the expansion of the WWW into our everyday life, education has become increasingly a consumer commodity. Today's learners expect high quality, relevant educational materials, delivered to them in a timely and appropriate manner. Adaptive educational hypermedia (AEH) [1],[2] systems aim to personalise the delivery of educational materials to the needs of the user – both to their stated requirements as well as to their less obvious desires. This has led to the development of many on-line educational delivery systems (e.g., Interbook [3], AHA! [8], TANGOW [4], WHURLE [11]). Many of these systems adapt their educational content to different *dimensions* of each learner, such as: current knowledge levels, computed user goals, immediate tasks, educational context (e.g., are they in school, university, or learning from home?), and more recently learning styles in adaptive hypermedia (e.g., LSAS [10]). Each of these AEH systems uses its own content model, coding methodology, and style. This leads to an unnecessary and undesired level of complexity for the content or lesson author. Ideally, the author would create materials once and then use these materials in any AEH system. Transfer between these educational environments would be simple. This would also encourage cost-effective re-use of materials in a variety of ways. Authoring for adaptive hypermedia education systems has not been addressed until recently. This paper describes the translation of educational materials between two AEH systems, My Online Teacher (MOT) [12] and WHURLE. We discuss how using MOT as the authoring system and WHURLE as the delivery environment fosters a 'create once, use many' approach to educational content. A previous paper addresses similar issues in translating from MOT to AHA! [13]. Please note that these systems have been developed totally independent of each other and this type of conversion exercise also is a test of the systems' expressivity and flexibility.

2 Designing Instructional Strategies in MOT

2.1 MOT Basics

MOT is an online environment for the authoring of adaptive educational hypermedia. It is based on the theoretical framework LAOS [5], and aims at providing maximum flexibility whilst concomitantly reducing the author load.

2.2 MOT Static Components

MOT allows authors to either create content, using adaptation dynamics written by others, or to design their own adaptation dynamics (and, e.g., use contents created by others). The static part is layered, based on the information type stored (as explained below), so that each layer can have different authors.

The first static layer is the *domain concept model*, containing the learning resources organized hierarchically as concept maps and their descriptions (i.e. metadata, organized as concept attributes). The default attribute set contains: *title*; *keywords*; *introduction*; *pattern*; *text*; *explanation*; *conclusion* and *exercise* (although only the title is obligatory). In Figure 1, an author has created the attributes *text_adv* and *text_beg* for advanced and beginner learners respectively.

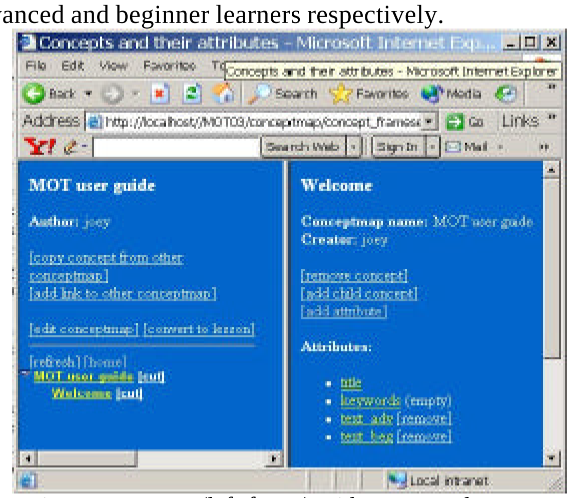

Fig. 1. A MOT domain concept map (left frame) with an example concept, called 'Welcome' (right frame) with 'text_beg' and 'text_adv' attributes

The second static layer is the *goal and constraints* (also called *lesson*) *model*. It contains selections filtered from the domain model based upon instructional views and goals [9] (such as expected timeframe, level, background knowledge, learner styles, etc.). For instance, an introductory short lesson on 'Neural Networks' contains, among others, the selection of 'beginner' concepts 'The Von Neuman computer versus the human brain' and 'The biological neuron' (Figure 2). The objective is to separate purely domain-dependent information (such as learning resources) from additional pedagogic information added by the teacher. For instance, weight information about the relevance of the resource to the different types of learner (adv: advanced or beg: beginner) may be stored here.

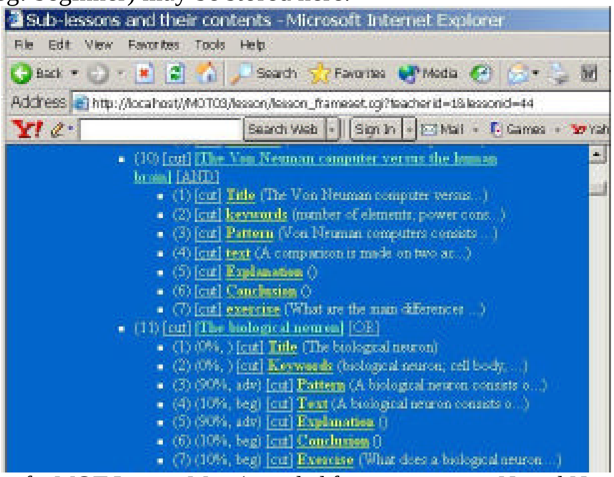

Fig. 2. A section of a MOT Lesson Map (sampled from a course on Neural Networks), showing the concept 'The biological Neuron' with the weights attached to each of that concept's attributes (e.g., '90%, adv' for 'Pattern': only advanced learners see this)

2.3 MOT Dynamic Components

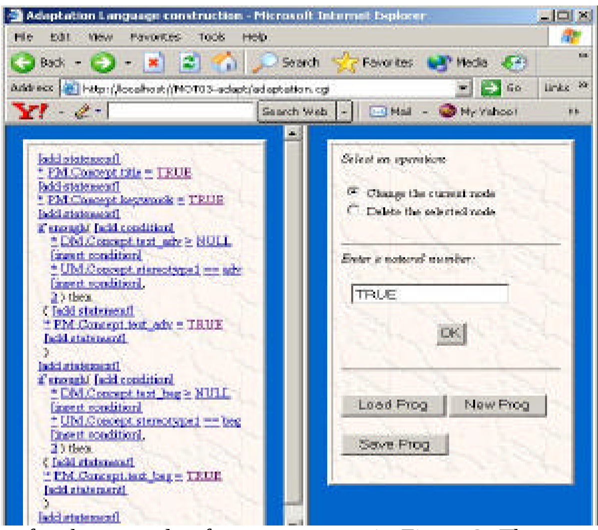

Fig. 3. A strategy for the example of a concept map in Figure 2. The strategy checks if the special attributes text_adv and text_beg exist, and then shows text_adv for advanced users and text_beg for beginners. Title and keywords are shown for all

The dynamic, adaptive behaviour of the courseware elements (in accordance with the learner characteristics determined by a user model), is designed in MOT using an

Adaptive Strategy Interface (Figure 3). This environment is built based on the *3-Layers of Adaptation Granulation* theoretical framework (LAG) [7]. MOT has a frame interface that allows the author to use a specifically designed adaptive language, which enables the building of adaptive strategies that correspond to instructional strategies [5]. Figure 3 shows a small instructional strategy corresponding to the new attributes text_adv and text_beg (introduced in Figure 1). The power of this adaptive instructional strategy is not that it can be used on the concept map in Figure 1, but that it can be used on *any* concept map that has the attributes text_adv and text_beg.

3 WHURLE

3.1 WHURLE Basics

WHURLE is a flexible, discipline-independent integrated learning environment, designed to deliver adaptive content over the web. The learner is presented with a lesson, which is constructed from a collection of underlying educational resources, according to a default narrative that is defined by a lesson author in a *Lesson Plan,* and filtered according to rules specified in the user model. WHURLE uses an XML-pipeline (i.e. the results of one process are fed into the next) where components are present in a modular fashion – this means that any particular user or pedagogical model can be utilised within the pipeline, as an appropriately designed adaptation filter (Figure 4) [15]. The list of resources defined in the *Lesson Plan* is passed to the *Adaptation Filter*, which passes those deemed necessary, according to information in the user profile, to the *Display Engine* which, in turn, creates the *Virtual Document* – rendered in a user interface defined in a skin.

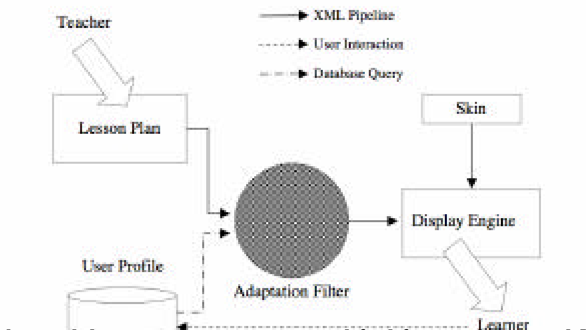

Fig. 4. The modular WHURLE system, modified from Zakaria et al [15]

3.2 WHURLE Static Components

WHURLE content resources are called *chunks.* Each *chunk* is a conceptually discrete piece of information (i.e. there are no interdependencies between chunks, and no links to other resources). An example of a WHURLE chunk would be a captioned image or a self-contained paragraph of text on a single topic. Owing to the flexibility provided by WHURLE's use of chunks, adaptation may be implemented at the content level (using conditional transclusion [14]) to determine which chunks are made available to the learner. Another static resource used in WHURLE is the *Linkbase*, which describes links between a chunk, in any given context, and other learning resources (e.g. other chunks or an external WWW page). These links are robust, bi-directional, and multiply typed as described in [14].

3.3 WHURLE Dynamic Components

WHURLE's dynamic adaptation is implemented by the adaptation filter, which determines the chunks that are delivered to the learner (Figure 5). This decision is determined by rules defined in the user model acting upon metadata in the *Lesson Plan* and data contained in the user profile. The current implementation allows for domain knowledge based adaptation [16]; where each chunk reference in the lesson plan contains the information that the adaptation filter uses to determine inclusion in the virtual document. Hence, in producing a lesson for WHURLE, not only does an authoring system have to produce the content but the lesson plan must contain the metainformation used as the basis for adaptation.

4 Conversion from MOT to WHURLE

In the following, we explain three ways of conversion of some of the MOT components presented previously. The main process being that the MOT interface feeds WHURLE information about what content should be provided to the learners, as well as how this content should be filtered and adapted.

4.1 Conversion of Concepts

The Domain Concept Model used by MOT is a hierarchical structure organised by 'concept' (Figure 1). The Lesson Plan used in WHURLE is also a hierarchical structure, organised by 'level' (Figure 5). Each MOT concept has 'attributes' and each WHURLE level has 'chunks' (collected into a 'page') that define the actual content. From this basic description we can begin to derive a conceptual mapping of MOT to WHURLE. MOT has several default standard attributes, of which 'title' and 'keywords' are common to WHURLE chunks. Therefore, in any conversion it is necessary that these common elements are included in every chunk created. We shall illustrate the process of conversion using the example shown in Figure 1. The MOT concept in Figure 1 will be converted into two chunks, as the contents of the 'text_beg' and 'text_adv' attributes can be seen as pedagogically separate. The 'title' and 'keyword' attributes, as common elements, are included in both chunks. The conversion process so far would produce two discrete chunks (content) but no surrounding structure tying them together into a single pedagogic unit. The next part of the conversion process is the production of a WHURLE Lesson Plan to provide this structure. At this stage each MOT concept is equivalent to a WHURLE 'page' (see Figure 5). For example, the concept in Figure 1 would transform into the

WHURLE Lesson Plan shown in Figure 5. This adaptive behaviour is an instance description and can only be applied on the resources from the specific concept map in Figure 1.

```
....
<level name="user" title="Welcome">
  <page>
    <chunk domain="100"
           stereotype1="beg">MOT-12-324</chunk>
    <chunk domain="100"
           stereotype1="adv">MOT-12-325</chunk>
  </page>
</level>
```

Fig. 5: a section of a WHURLE Lesson Plan, the result of the conversion of the ‘Welcome’ concept shown in Figure 1

Of course for this process to be meaningful the conversion engine needs to be informed of the semantic relationship between the MOT attribute name and the WHURLE stereotype value, for example, that ‘text_beg’ is equivalent to ‘stereotype1 = “beg”’. The value of the WHURLE ‘domain’ attribute (in Figure 5, this is ‘100’), is linked to a table of values in a database. When transforming a MOT Domain Concept Map, the root concept title is registered in the database as the domain name. This value will thus be the same for all chunks produced from the same Domain Map. Figure 6 shows the outcome of such a conversion with a beginner and an advanced view.

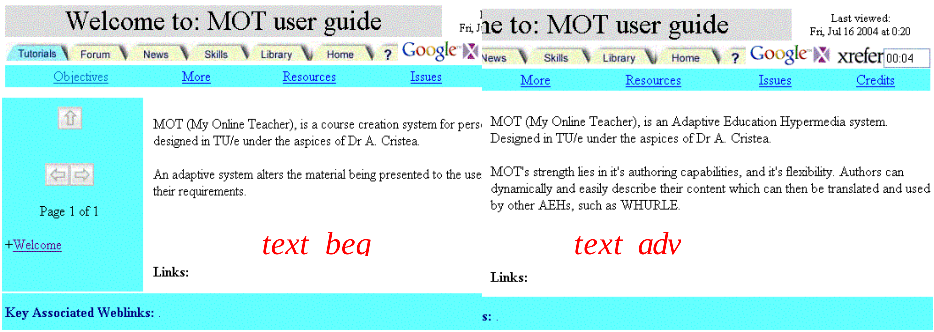

Fig. 6. WHURLE rendering of the beginner (a) and advanced (b) user version conversion from the structure in Figure 1

4.2 Conversion of Lessons

One of the benefits of MOT is the ability to create any attribute the author requires to fully describe a concept. This feature cannot be fully exploited when involving the conversion described in section 4.1, since the conversion engine needs to be altered each time an author creates a new conceptual relationship between a MOT concept attribute and the WHURLE ‘stereotype’ attribute. An attribute in a MOT Lesson Map can have a weight and a label attached to it. When translating from a MOT Lesson Map to a WHURLE Lesson Plan we can use these weight values to our advantage. Figure 2 shows a simple lesson concept, its attributes and their assigned weights. The conversion engine can determine the WHURLE stereotype from a table of weight values. Table 1 shows an example of such a table (a ‘0’ weight represents a common element; so is not shown in the table). Alternatively, the labels attached to the weights can be interpreted. This allows greater flexibility for the author, since they can create/modify/delete their own common elements – although with the stipulation that the ‘title’ and ‘keyword’ elements are always present and common.

Table 1. A simple table of MOT weight values to WHURLE stereotype value

Weight	Stereotype
1-49	beg
50-89	int
90-99	adv

Table 2 shows how the concept ‘The biological neuron’ from Figure 2 is processed to produce four WHURLE chunks. Chunk ‘C1’ from the conversion in Table 2 includes *only* the common elements. Chunks ‘C2’, ‘C3’ and ‘C4’ are produced by combining the common elements with each successive attribute. This is done irrespective of weight value. Even though attributes may share the same weight, they may nevertheless be pedagogically distinct.

Table 2. A stylised version of the concept shown in Figure 2, alongside a representation of which attributes are included in the chunks produced (C1-C4). As the attributes ‘Explanation’ and ‘Conclusion’ are empty, they will not result in a chunk being produced

MOT attributes		WHURLE chunks			
Attribute	*Weight*	*C1*	*C2*	*C3*	*C4*
Title	0	✓	✓	✓	✓
Keywords	0	✓	✓	✓	✓
Pattern	90		✓		
Text	10			✓	
Explanation	90		No attribute contents		
Conclusion	10		No attribute contents		
Exercise	10				✓
			✓= included in the chunk		

For example: chunk ’C2’ would only be relevant to an advanced learner (in the current domain of Neural Networks) and chunk ‘C3’ relevant to a beginner learner. Creation of the chunks is, of course, still only the first step in the conversion process. A WHURLE Lesson Plan must also be produced. Using the hash table (Table 1) and the chunks C1-C4 from Table 2 would result in the Lesson Plan shown in Figure 8 (note that it only shows a subset of the complete Lesson Plan resulting from the conversion of the concept in Figure 2).

```
....
<level name=”neuron” title=”The Biological Neuron”>
  <page>
    <chunk domain="general"
           stereotype1="">C1</chunk>
    <chunk domain="110"
           stereotype1="adv">C2</chunk>
    <chunk domain="110"
           stereotype1="beg">C3</chunk>
    <chunk domain="110"
           stereotype1="beg">C4</chunk>
  </page>
</level>
....
```

Fig. 7. A section of a WHURLE Lesson Plan resulting from the conversion of the ‘The Biological Neuron’ concept shown in Figure 2. Note: a ‘domain’ value of ‘general’ indicates that a chunk C1 (our generic *title* chunk) will be seen by every user; a ‘domain’ value of ‘110’ is the value WHURLE uses to indicate the Neural Network domain

Another feature of the MOT Lesson Map is that it combines concepts from multiple Domain Concept Maps, so the WHURLE ‘domain’ attribute has to be drawn from the root ‘title’ attribute of the parent Concept Map for each concept in the lesson, stored in the relevant table and assigned a value. It is this value that is inserted into the WHURLE chunk attribute ‘domain’.

4.3 Conversion of Adaptation Dynamics

Whilst the conversion described in section 4.2 is far more flexible than that described in section 4.1, it is still limited. The author is restricted to a specific table, which itself only describes a single pedagogical model. In the example given in section 4.2, this conversion is restricted only to domain knowledge.

The rule set described in the LAG model [7] is far more flexible and powerful. For instance, an author can create their own adaptation rules or use pre-existing adaptation rule templates.

For the two example conversions previously presented (Figures 5&7) there doesn’t have to be a hardwired MOT-to-WHURLE interpretation. By using the MOT adaptation strategy editing interface (section 2.3), rules can be created specifying the presentation conditions and order related to specific conceptual model structures. These can be used to do the actual conversion, the result for the cases above being the same: Figure 5 would be the result of the conversion of the domain concept map in Figure 1 with the interpretation given by the adaptive strategy in Figure 3. Respectively, Figure 7 would be the result of the conversion of the domain lesson map in Figure 2 with the interpretation given by the following adaptive strategy (Figure 8). The current conversion programs, however, follow the models given in sections 4.1 and 4.2.

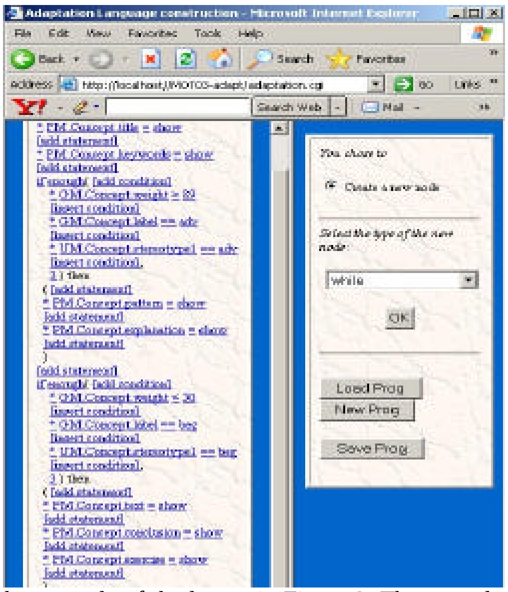

Fig. 8. A strategy for the example of the lesson in Figure 2. The procedure interprets weights and labels for advanced & beginner users in the Lesson map (adv, beg); then shows adv for advanced users and beg for beginners. Title and keywords are shown for all

Future development of our AEH authoring system will involve using these adaptation rules and templates. A ‘teacher’ would declare the rules they wished to employ, and use these as the basis of the adaptation of their lesson and its conversion into a WHURLE Lesson Plan.

5 Conclusion

Most traditional on-line learning environments have a serious pedagogic limitation, in that they are unable to adapt to the requirements of the user. The 'lesson' content is presented in a simple, static manner and the user is expected to use these materials until they have completed whatever the objectives of the lesson may be. Whilst there is often a help system in place to aid the student, the presentational and pedagogic structure always remains the same. Research has shown [1], however, that learners differ in their ability to learn. Although it may seem trite to state that every student is unique, many systems ignore this simple fact.

In this paper we have briefly presented MOT, focussing on its capabilities as an AEH authoring system and WHURLE, focussing on its capabilities as an AEH delivery system, and we have described the first steps towards interfacing these two systems. We have shown, with the help of examples, how, from the generic level of authoring for any concept map (as in MOT) we can obtain an instance representation (as in WHURLE). There is still much work to be done to be able to use more of the MOT flexibility in WHURLE. Moreover, both systems are still growing, so this interfacing exercise also provides pointers to future development possibilities, as well as necessities. At first glance, the delivery systems in today's AEHs are quite different. For instance, the main difference between WHURLE and AHA! v. 3.0 is that in AHA! the adaptive behaviour of the concepts is contained in the concept description itself. This means that the same concept will behave the same way given the same variable instantiations, no matter where and when it is called. WHURLE doesn't have concepts, just 'chunks' and these behave as specified by the lesson plan, and thus can behave differently when used with different lesson plans. More importantly, AHA! allows an overlay approach to user model variables, whereas one instance of user modelling implemented in WHURLE works with a hybrid of stereotypes and domain knowledge. Commonalities, however, can be found. For instance, WHURLE chunks would correspond somewhat to AHA! XHTML resources (although these also historically contained some adaptive inclusion information, which is now becoming obsolete). This would make AHA! concepts behave somewhat like WHURLE lesson plans. Other, more general commonalities are that both systems implement user models, basic educational resources, adaptive rules for adaptive behaviour, etc. It is constructive to strive towards being able to create material in one system, e.g., MOT, and being able to deliver it on various platforms. In this way we tackle the important problem of authoring for adaptive educational hypermedia. By examining the challenges and problems of translation between AEH systems first *pair-wise*, then *one-to-many*, we can, in principle, determine the elements needed for a generic interface. This is one of the necessary steps towards extracting patterns of adaptive authoring and being able to reuse not only the static, but also the dynamic material created within any system, into any other system.

7 Acknowledgements

This work is supported by the ADAPT EU Minerva project, (101144-CP-1-2002-NL-MINERVA -MPP).

References

1. P. Adey, R. Fairbrother, D. Wiliam, B. Johnson, and C. Jones, *Learning styles & strategies: a review of research* London, King's College, Centre for the Advancement of Thinking, 1999.
2. P. Brusilovsky, Adaptive Educational Hypermedia. *Proc. of 10th Tenth Intl PEG Conference*. Tampere, Finland. pp 8-12, 2001.
3. P. Brusilovsky, J. Eklund & E. Schwarz, Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, In *Proc. of 7th Intl World Wide Web Conference*, 30 (1-7), 291-300, 14-18 April 1998.
4. R. M. Carro, E. Pulido, P. Rodríguez, Designing Adaptive Web-based Courses with TANGOW. *Proc. of the 7th Intl Conference on Computers in Education, ICCE'99*, Chiba, Japan, V. 2, 697-704, November 4 - 7, 1999.
5. A. Cristea, Adaptive Course Creation for All, *Proc. of the International Conference on Information Technology*, Las Vegas, US, IEEE, April, 2004.
6. A. Cristea, & A. De Mooij, LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. *Proc 14th Intl. World Wide Web Conference*, Budapest, ACM, Hungary 20-24 May 2003.
7. A.I. Cristea, & L. Calvi, The three Layers of Adaptation Granularity. *Proc UM'03*, Springer.
8. P. De Bra, A. Aerts, B. Berden, B. De Lange, B. Rousseau T. Santic D., Smits & N. Stash, AHA! The Adaptive Hypermedia Architecture. *Proc. of the fourteenth ACM conference on Hypertext and Hypermedia*, Nottingham, UK, 81-84, 2003.
9. F. Garzotto and A. Cristea, ADAPT Major Design Dimensions for Educational Adaptive Hypermedia, *Proceedings of ED-MEDIA'04*, AACE (to appear).
10. LSAS, Learning Style Adaptive System, described at: http://www.archives.ecs.soton.ac.uk/users/nb99r/intro_short/frame.htm
11. A. Moore, T. J. Brailsford and C. D. Stewart, Personally tailored teaching in WHURLE using conditional transclusion. *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, Denmark., 2001
12. MOT, My Online Teacher, described at: http://wwwis.win.tue.nl/~acristea/mot.html
13. N. Stach, A. Cristea & P. De Bra, Authoring of Learning Styles in Adaptive Hypermedia: Problems and Solutions. *Proc. 13th Intl WWW Conference*, NY, 2004.
14. A. Moore, C. Stewart, D. Martin, T. Brailsford, H. Ashman, Links for Learning: Linking for an Adaptive Learning Environment. IASTED Intl Conference on Web-Based Education, Innsbruck, Austria, Feb 2004.
15. M.R. Zakaria, A. Moore, C.D. Stewart, T.J. Brailsford, "Pluggable" User Models for Adaptive Hypermedia in Education. *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia (Hypertext '03)*, 2003, pp 170-171.
16. M. R. Zakaria and T. J. Brailsford, User modeling and adaptive educational hypermedia frameworks for education, New Review of Hypermedia and Multimedia (NRHM 2002), 8 (2002), 83-97.

An authoring environment for adaptive testing[1]

Eduardo Guzmán, Ricardo Conejo, Emilio García-Hervás

Departamento de Lenguajes y Ciencias de la Computación.
E.T.S.I. Informática. Bulevar Louis Pasteur, 35. 29071, Málaga. SPAIN
{guzman, conejo, egarcia}@lcc.uma.es

Abstract. SIETTE is a web-based adaptive testing system. It implements Computerized Adaptive Tests. These tests are theoretically based tailor-made tests, where questions shown to students, the finalization of the test, and the student knowledge estimation is accomplished adaptively. To construct these tests, SIETTE has an authoring environment. It is a suite of tools that help teachers to create questions and tests properly, and to analyze of the students' performance after taking a test. In this paper, we present this authoring environment in the framework of adaptive testing. As will be shown, this set of tools, that own some adaptable features, can be useful to help teacher without too much skills on this kind of tests.

1. Introduction

Testing is among the most widely used tools in higher education [1]. The main goal of testing is to measure the student knowledge level in one or more concepts or subjects, i.e. in pieces of knowledge that could be assessed. This kind of assessment has been used for student knowledge diagnosis in adaptive educational systems like in EML-ART [16] or DCG [15], but most of these systems used heuristic-based testing techniques. However, there is another kind of tests, the adaptive tests, which are based on a theoretical-sound theory, the *Computerized Adaptive Testing* (CAT) theory [14]. This theory defines which questions (called *items*) is the most adequate to be posed to students, when the tests must finish, and how the student knowledge can be inferred from students' performance during the test. To this end, CAT uses an underlying psychometric theory called *Item Response Theory* (IRT) [8].

Adaptive test elicitation is a task that requires a special effort from the teacher, since the construction of this kind of tests must be accomplished fulfilling some features. These features must be kept to ensure the correct operating of adaptive tests. For instance, teachers must ensure that the stem of one item does not provide any trail to correctly answer other items, i.e. items must be independent among themselves. Additionally, adaptive testing selection techniques must have available a considerably big set of items with a wide range of difficulties. These requirements demand for adaptive testing system the availability of an authoring environment that helps

[1] This work has been partially financed by LEActiveMath project, funded under FP6 (Contr. Nº 507826). The author is solely responsible for its content, it does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

teachers to construct items and tests. Additionally, this kind of systems needs some tool to analyze data of student test session, in order to study if the set of items owns the properties that it should have. Unfortunately there are only a few set of systems able to generate adaptive tests [1]. Also their authoring interfaces do not include adaptive and/or adaptable features.

SIETTE [3] is a web-based system for adaptive test generation. Moreover, this system is able to deliver conventional (heuristic-based) tests. Through a web interface, students can take tests for self-assessment, where item correction is shown after each item, with optionally some kind of feedback; or teachers can make grading tests in order to assess their students, even for academic purposes. To construct and modify the test contents, SIETTE offers an authoring environment. This is a suite of tools that permits teachers to principally edit tests. In this environment has been included a tool for analyzing the student's performances.

This paper is aimed at showing the authoring environment of the SIETTE system. Next section briefly explains what adaptive tests are. Section 3, shows the components of the SIETTE architecture. Following, section 4, is devoted to the test editor, showing its operation mode and its adaptable capabilities [11]. Section 5 is devoted to the result analyzer, a tool that allow teachers to study the students' performance in the tests they have made. Finally, in section 6, the conclusions of this work are summarized.

2. What is an adaptive test?

CAT theory tries to mimic the usual assessment procedure followed by a human teacher. That is, first to administer the student a medium difficulty item. If the student answers correctly, next administer a little more difficult item, and in other case, administer other less difficult item. This process should be repeated until the teacher considers that he has enough evidences to determine the student knowledge level. Consequently, in CAT theory, this process has been automatized. Items are posed one by one. After posing an item, a temporary student knowledge level estimation is accomplished. In terms of this estimation, the next item to be posed is chosen in such a way that this estimation is more accurate. In more precise terms, an adaptive test can be seen as an iterative algorithm that starts with an initial estimation of the student's knowledge level, and has the following steps:

1. All the items that have not been administered yet, are examined to determine which is the best item to ask next according to the current estimation of the examinee's knowledge level.
2. The item is asked, and the examinee responds.
3. According to the answer, a new estimation of the knowledge level is computed.
4. Steps 1 to 3 are repeated until the stopping criterion defined is met.

IRT postulates that there is a relationship between the student knowledge level and the probability of successfully answer an item. This dependence relationship is probabilistically expressed by means of a function called *Item Characteristic Curve* (ICC). Accordingly, this function collects, for each knowledge level, the probability of that a student with this level answers correctly. If this probability function is available for every item of a test, the student knowledge can be directly inferred. In CAT theory, IRT is used to estimate the student knowledge level, in order to determine the next item to be posed, and to decide when to finished the test. This theory ensures that the obtained student knowledge estimations do not vary in terms of the items used in the estimation process.

The main advantage of adaptive tests is that they are fitted to students. This means that the number of items posed is different for each student, and depends on his knowledge level. As a consequence, students neither get bored for being administered very easy items, nor feel stressed for being administered very difficult items. In addition, different sets of items are posed for different students. Consequently, this reduces the possibility of cheating. In contrast, the main disadvantage of an adaptive test is that its construction is costly. Each ICC must be determined (calibrated) before an adaptive test could be applied. To this end, a big student population must be administered this test non-adaptively, and after from these data the calibration can be done.

3. The architecture of SIETTE

SIETTE allows CAT elicitation and delivering through web interfaces. It can work as a standalone assessment tool or inside other web-based adaptive systems, as a diagnosis tool. It is a multilingual system, currently available in Spanish and English, but open to include other new languages. Fig. 1 collects the architecture of the system. It comprises two main parts: the student workspace and the authoring environment.

The *student workspace*: This is the place where students can take tests. The main component of this part is the *test generator*, which is in charge of test delivering. Two interfaces can be used to access to generated tests:

- *Student classroom*: Here, students can take tests for self-assessment, and teachers can administer tests for grading.
- *Interface for external connections*: This interface permits SIETTE to work as a diagnosis tool in other web-based adaptive hypermedia educational systems. An own simple protocol [5] has been defined to this purpose.

The *authoring environment*: It is a suite of tools used by teachers. They allow content creation and update, as well as analyzing the performances of students that have taken tests. This suite is composed by the following tools:

- The *test editor*: Through this tool, teachers can create subjects. Related to the topics of each subject, different sets of items can be defined. Teachers can also define different tests that involve the subject topics.
- The *result analyzer*: This tool helps teachers to make analysis of the student performance.
- The *item calibration tool*: As been shown, ICCs functions predict the behavior of students that answer the corresponding item. They are determined by a set of parameters. These parameters are inferred by calibration techniques [4]. In this

part of the architecture, some of these calibrations techniques are being developed. Unfortunately, this tool is currently under development and, therefore will not be approached in this paper.

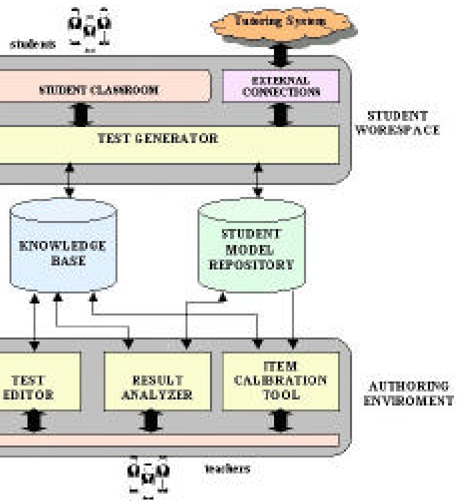

Fig. 1. Architecture of SIETTE

4. The test editor

In SIETTE, teachers can define different subjects. The curriculum of the subjects, the tests and items are stored in the knowledge base. Subject curricula are structured forming acyclic graphs of topics. Therefore, a subject can be divided in topics. Each topic can be also divided in subtopics and so on. As a result, each curriculum can be seen as a granularity hierarchy [10], where topics are related to their subtopics via aggregation relations. Items are assigned to topics in such a way that if an item is assigned a topic, this item is used to assess the student knowledge level in the topic. Items can be assigned to any topic of the hierarchy, included the subject, since it can be seen as a global aggregation of the whole curriculum. At last, tests are defined on topics. If a test involves a set of topics, after a testing session, SIETTE is able to return a student knowledge estimation for each test topic, and for each one of their descendant subtopics at any level.

In order to access to this tool, teachers must be provided with a pair identifier/password given by the system administrator. A snapshot of this tool, after selecting a subject to edit, is shown in Fig. 2. It is divided in two main frames. The left one is the curriculum hierarchy tree. Two different views of this tree can be seen: items or tests. When the "items" option is selected, the tree shows the subject curriculum hierarchy, composed by the topics and their items. Topics are represented by folders, and items by color balls. In terms of the kind of item, the ball color differs. If the "tests" option is selected, the tree shows the tests that have been defined for this subject. Under each test, the curriculum of the test topics is shown. At last, the look of the right side depends on the element selected in the tree. Subject, topic, item and test information can be added, modified or deleted through this frame.

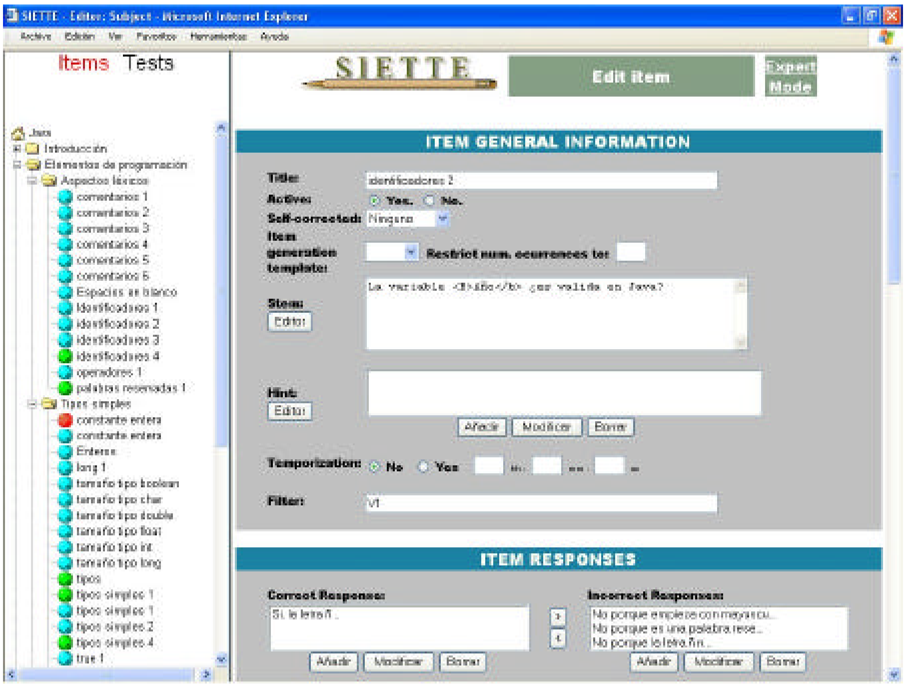

Fig. 2. The test editor

The element parameters of the editor can been seen in two different ways, depending on the teacher profile. The test editor adapts the content presentation in terms of the teacher profile. Two different teacher stereotypes [9] are managed: novice and expert. In terms of his mastery in the use of this tool, teachers can select either one stereotype or the other, and are free to change it at any time. The difference between them rests on the level of detail of the information shown. In novice profile, some information is hidden. When a teacher with this profile is editing an element, some of its parameters will take values by default. Expert profile has been conceived for teachers with more advanced mastery on the system and/or in the use of adaptive tests.

Different teachers can access to the same subject. For each subject, there is a teacher who is its creator, and has all permissions granted. He can grant, through the editor, different permissions to other teachers. As a consequence, the set of actions a teacher can accomplish is adapted to the permission he has on the subject. These permissions are: item reading or item modification, curriculum modification, test addition or test modification, etc.

4.1. Test definition

Chua Abdullah [2] pointed out the types of knowledge that teachers must take into account to get effective testing assessment: (1) *what to test*, that is the parts of the domain knowledge to be tested; (2) *who to test*, this is the student model; (3) *how to test*, that is the item selection criterion and the student assessment method. We have added another one: (4) *when to finish the test*, i.e. the decision of test finalization. In adaptive testing, this last decision is vital, since it will determine the accuracy and, as a result, the reliability of the student assessment. In the test definition stage of SIETTE, all these concerns are expressed by test configuration parameters. The first concern (*what*) is represented by the topics involved in the test, and the number of knowledge levels in which the students will be assessed. Although in IRT, the real number domain is used, in SIETTE, for simplicity, a discrete domain is used. Accordingly, if the number of knowledge levels is equal to *K*, students will be classified between *0* and *K-1*.

The *who* is clearly the student represented by his student model. Student models in SIETTE are essentially probability distribution curves, which contains, for each knowledge level, the probability of that the student knowledge will be this knowledge level. For each topic assessed in a test, SIETTE keeps a student distribution curve. When creating a test, SIETTE provides teachers the possibility of selecting the prior probability distribution their student will have before posing any item.

Finally, the *how* and *when* concerns will be undertaken in the following three subsections:

4.1.1. Item selection criteria

SIETTE provides two different item selection adaptive criteria:

- *Owen's adaptive criterion*: It uses a discrete version of Owen item selection approach [12]. It selects the item that minimizes the expectation of the variance of the posterior student knowledge distribution.
- *Difficulty-based criterion*: Owen found a simplification of his previous selection criterion (*op. cit.*), whose performance is very near to the former, and which it is very simple to apply. It selects the item whose difficulty (a parameter of the ICC) is the nearest to the current student knowledge level estimation.

4.1.2. Student assessment techniques

In the *how* concern, we do not just have to consider the item presentation order, it is necessary to make the decision of what mechanism must be used to infer the student knowledge level. In SIETTE, the adaptive assessment methods are based on a discrete Bayesian mechanism in which the student knowledge probability distribution is calculated after posing each item i (Equation 1). The estimation made after posing the last test item becomes the final estimation.

$$\mathrm{P}(q \mid \mathrm{u}_1,..,\mathrm{u}_\mathrm{i}) \propto P(\mathrm{u}_\mathrm{i}=1 \mid q)^{u_i}(1-P(\mathrm{u}_\mathrm{i}=1 \mid q))^{(1-u_i)} P(q \mid u_1,.u_{i-1}) \qquad \textbf{(1)}$$

In Equation 1, $P(q|u_1, .., u_{i-1})$ is the temporary student estimation before answering the item i, and $u_i=1$ represents that student has answered correctly the item. $P(u_i=1|q)$ is the ICC for the item i. As mentioned before, it expresses the relationship between the item correct answer and the knowledge levels. Once the new estimation distribution is calculated, the student knowledge level can be inferred in two different ways, in terms of the adaptive criteria used: *modal*, that is the most likely level; or *expectation-based*, where the estimated knowledge level is equal to the probability distribution expected value.

In SIETTE, items are assigned to the topics they assess. If an item Q is used to assess a topic T, applying the aggregation relations defined in the curriculum, item Q can be used to assess all the topics preceding T. In order to manifest this relation, each item has an ICC for each topic it can assess. Accordingly, after a single test, the system is able to return the student knowledge state in the test topics and in all their descendants [6].

4.1.3. Test finalization criteria

In order to ensure the test finalization, and to avoid the item overexposure, an item maximum number is defined for each test. While a test is being administered, every time an item is selected, this upper limit is compared to the number of items already administered. If this last number is equals or greater than the limit, the test is forced to finish. While this condition is not satisfied, test finalization can be decided by one of the following criteria:

- The student knowledge estimation variance is lesser than a certain threshold;
- the student estimated knowledge level probability is greater than a certain threshold;
- or, for temporized tests, the time limit has been reached.

Whereas the two former criteria are purely adaptive, the last one, although it is non-adaptive, it can be applied to adaptive tests as an alternative mechanism to avoid very long tests. SIETTE offers the possibility of configuring tests to be temporized. To this end, teachers only have to set the test time limit through the editor.

Additionally, other configuration parameters can be set for each test: its availability can be restricted to one or more groups of students; filters can be configured to restrict the items that can be administered in each test; teachers can allow students to retake a test at the same stage they left, if the test has been paused for any reason (for instance, connection failure); etc.

4.2. Item definition

In order to construct their tests, in SIETTE, teachers are supplied with several types of items:

- *True/false items*, where students have to select just one answer.
- Multiple-c*hoice items*, where students must select an answer or no one.
- *Multiple-response items*, where more than one answer can be correct.

- *Self-corrected items*, they are little programs, implemented by means of java applets or flash, which allow teachers to include more sophisticated exercises. They are correct by themselves, and this correction is given to SIETTE.

These types of items can be combined in the same test. The former items have the classical format of a stem and a set of answers. SIETTE offers other kind of item construction scaffoldings, a library of exercises templates. It collects most of the exercises that usually appear in textbooks. They can be added easily to a test, by instating the desired template. Additionally, SIETTE includes a mechanism of item generation. This mechanism has been implemented through item templates written in a web language (e.g. JSP, PHP, etc.). These templates generate questions of any of the previous types, after being pre-processed. For more information about the types of items and the item generation see [7].

5. The result analyzer

Student model repository stores information about the student test sessions. The result analyzer of SIETTE allows teachers to study these data. It contains the following two utilities:

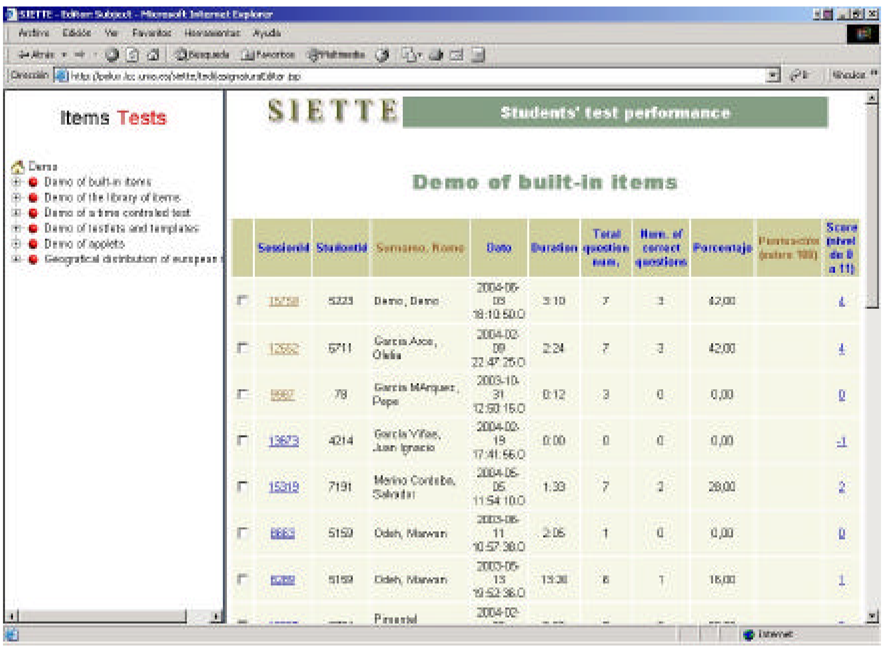

Fig. 3. The student performance facility of the result analyzer

A student performance facility: It contains for each test, the list of students that have taken the test. It shows for each student the identifier of the test session, his identifier and name, the date of the beginning of the test session, the total number of items posed, the number of items correctly answered and his final qualification. It allows seeing the complete test session, that is, the items that were given to the student in the same order posed, and with the student's response and the correct response. This tool gives detailed statistics of the final student's knowledge level estimation. For each topic, the estimation, the number of items posed and the number of correctly answered items topic, as well as a graphical representation of the estimated knowledge distribution is provided. Additionally, it offers the possibility of deleting student' test information from the student model repository.

An item statistic facility: It supplies statistic information about the student responses to the item in all test sessions in which the item has been posed. These data can be studied for each topic to which the item is directly associated, and for each one of its preceding topics, even including the subject. Once the topic to be studied has been selected, a table is shown. It contains a column for each answer of the item. Each row represents a knowledge level in which the subject can be assessed. Each cell c_{ij} of the table represents the number of students with final estimated knowledge level i that have selected the answer j. In addition, cumulative statistics are shown. That is, taking all the data of the student model repository as sample, the likelihood of that a student selects an answer given his final estimated knowledge level. This information is very useful for calibration purposes.

6. Conclusions

SIETTE is a well-founded testing system that generates adaptive tests for grading or self-assessment. These tests have a lot of advantages: tests are suited to students, the number of items requires for assessment is lesser than in the conventional testing procedures, estimations have high accuracy, etc. In SIETTE contents are structured in subjects. Each subject is composed by a set of topics, structured hierarchically using aggregation relations. Each topic has associated a set of items that can be used to assess it and all its preceding topics. Furthermore, SIETTE provides teacher with an authoring environment. It is a set of tools that allow teachers the knowledge elicitation, i.e. item, topic and test construction. It is a multi-user environment in which teachers can collaborate in the test creation process, although this collaboration can be restricted by applying different permissions on the elements of each subject.

SIETTE has on the one hand adaptive features: the item selection, student assessment and test finalization criteria. These criteria are based on the performance of the students while taking the tests. On the other hand, through the adaptable characteristics, the test editor is personalized to each teacher profile and permissions. For instance, the novice profile is very useful for teacher with no skills about adaptive test configuration. Also, different item construction scaffolding brings teacher the possibility of easily adapting the test presentation to the population that is going to make a test.

Both the student classroom and the test editor are available for any user through the following URL: http://www.lcc.uma.es/SIETTE. In the student classroom, there has been defined a subject "Demo" which includes several demo tests. These tests have been created to show some characteristics of SIETTE. Moreover, a "demo" teacher account has been created to freely access to the test editor and the result analyzer, in order to show their operability and the adaptable features.

References

1. Brusilovsky, P. and Miller, P. Web-based Testing for Distance Education. De Bra, P. and Leggett, J. Proceedings of WebNet'99, World Conference of the WWW and Internet. 149-154. 99. AACE.
2. Chua-Abdullah, S. Student Modelling by Adaptive Testing - A Knowledge-based Approach. Canterbury, University of Kent. (2003)
3. Conejo, R., Guzmán, E., Millán, E., Pérez-de-la-Cruz, J.L., and Trella, M. SIETTE: A web-based tool for adaptive testing. International Journal of Artificial Intelligence in Education (*forthcoming*).
4. Glas, C.A.W. Item calibration and parameter drift. In Van der Linden, W.J. and Glas, C.A.W. (eds.) Computerized Adaptive Testing: Theory and Practice. Kluwer Academic Publisher,(2000).
5. Guzmán, E. and Conejo, R. An adaptive assessment tool integrable into Internet-based learning systems. Méndez Vilas, A., Mesa González, J. A., and Solo de Zaldívar, I. Educational Technology: International Conference on TIC's in Education. I, pp. 139-143. Sociedad de la Información. (2002)
6. Guzmán, E. and Conejo, R. Simultaneous evaluation of multiple topics in SIETTE. Cerri, S. A., Gouardcres, G., and Paraguacu, F. Lecture Notes in Computer Science 2363. Proceedings of the 6th International Conference on Intelligent Tutoring Systems ITS 2002. 739-748. Berlin, Springer Verlag. (2002)
7. Guzmán, E. and Conejo, R. A library of templates for exercise construction in an adaptive assessment system. Technology, Instruction, Cognition and Learning (TICL) (*forthcoming*).
8. Hambleton, R.K., Swaminathan, J., and Rogers, H.J. Fundamentals of Item Response Theory. Sage publications, California, USA. (1991).
9. Kay, J. Stereotypes, Student Models and Scrutability.Proceedings of the International Conference on Intelligent Tutoring Systems (ITS 2000). (2000).
10. McCalla, G. I. and Greer, J. E. Granularity-Based Reasoning and Belief Revision in Student Models. Greer, J. E. and McCalla, G. Student Modeling: The Key to Individualized Knowledge-Based Instruction. 125, 39-62. 94. Berlin Heidelberg, Springer Verlag. NATO ASI Series. (1994).
11. Oppermann, R., Rashev, R., and Kinshuk. Adaptability and Adaptivity in Learning Systems. In Behrooz, A. (ed.) Knowledge Transfer. pAce, London, UK (1997).
12. Owen, R.J. A Bayesian Sequential Procedure for Quantal Response in the Context of Adaptive Mental Testing. Journal of the American Statistical Association 70, 351-371 (1975).
13. Promissor Inc. CATGlobal. http://www.catglobal.com. [2004, July 12].
14. van der Linden, W.J. and Glas, C.A.W. Computerized Adaptive Testing: Theory and Practice. Kluwer Academic Publishers, Netherlands (2000).
15. Vassileva, J. Dynamic Course Generation on the WWW. du Bolay, B. and Mizoguchi, R. (eds.). Knowledge and Media in Learning Systems. Proceedings of the 8th World Conference on Artificial Intelligence in Education AIED'97.IOS Press, pp. 498-505. (1997).
16. Weber, G. and Brusilovsky, P. ELM -ART: An Adaptive Versatile System for Web-based Instruction. International Journal of Artificial Intelligence in Education 12, pp. 351-383 (2001).

Specification of Adaptation Strategy by FOSP Method

Milos Kravcik

Fraunhofer Institute for Applied Information Technology,
53754 Sankt Augustin, Germany
Milos.Kravcik@fit.fraunhofer.de

Abstract. The aim of this paper is to propose a novel method for collaborative authoring of adaptive educational hypermedia, which can be generalized also for other application areas. It addresses the objective to simplify the authoring process and make it more efficient. Its main idea is to separate the partial results produced by different authors in such a way that they can be reused. This concerns also adaptation strategies that specify how the domain model and the context model attributes should be processed to present the content to the user accordingly. An instruction designer specifies adaptation as sets of content object preferences for different contexts. We have identified a pattern in the adaptation process that consists of four operations – Filter, Order, Select, and Present.

1 Introduction

Relevant authorities [8] have identified as one of the five main challenges in information systems provision of learning environments that can efficiently enable each student to have his or her own teacher. The key research areas in this respect include cognitive tutors, collaborative authoring, and context learning. Efficient learning must be individualized, but systems that adapt to user preferences are not easily available. In the development of electronic documents in general, the main aim was typically to facilitate development and organization of ideas by bringing the medium closer to the cognitive events of the user [14]. To make documents prepared in different ways interchangeable and reusable, we have to deal separately with their various aspects (e.g. logical structure, physical appearance). Then authors can concentrate on document semantics (structure and contents), without having to deal with its layout, which can be created by designers and selected according to the use context. Based on this, standards have been created. *Standard Generalized Markup Language* (SGML) is an ISO standard for document structuring and interchange at the authoring and editorial stages. To associate formatting rules with SGML documents, another ISO standard appeared. *Document Style Semantics and Specification Language* (DSSSL) describes the way that text and graphics should be presented in two-dimensional environment.

In the following, we describe how these ideas have been reflected in formal hypermedia models, but have not been consequently implemented in the systems. Authoring adaptive hypermedia is still a complex process that needs to be simplified. Based on our experience [10, 11] we propose a method for specification of adaptation strategies, which should support collaborative authoring. We focus on adaptive educational hypermedia, but the method can be generalized for other fields as well.

2 Hypermedia Models and Specifications

In 1990 the basic formal hypertext model was presented – *Dexter Hypertext Reference Model* [9]. Its goal was comparison of existing systems as well as development of interchange and interoperability standards. The model distinguishes three layers of a hypertext system and two interfaces between them: *Runtime Layer* (presentation of the hypertext; user interaction; dynamics), *Presentation Specifications,* and *Storage Layer* (a "database" containing a network of nodes and links), *Anchoring, and Within Component Layer* (the content / structure inside the nodes). Based on the Dexter model, a reference model for adaptive hypermedia was developed [6], called *Adaptive Hypermedia Application Model* (AHAM). The model provides a framework to express the functionality of adaptive hypermedia systems by dividing the storage layer into three parts: *Domain model* (describes how the information content is structured), *User model* (describes the information about the user), and *Adaptation model* (adaptation rules defining how the adaptation is performed). AHAM uses Condition-Action rules and due to their complexity, it is not supposed that authors will write all the rules by hand. Some other models build upon AHAM identifying additional relevant layers in the model [5] and particularly in the adaptation model [4]. The objective is to enable reusability at various levels, focusing mainly on adaptation strategies and techniques. Additionally to the new formal models, also the electronic document standards have been adjusted for hypermedia, especially for the web. A simplified version of SGML has been created and named *Extensible Markup Language* (XML). It enables development of user-defined document types on the web and provides meta-data for web-based applications. *Cascading Style Sheets* (CSS) and *Extensible Style Language* (XSL) can specify presentation of XML documents.

3 Authoring Adaptive Educational Hypermedia

A major current shortcoming is that the different layers and factors proposed in the formal models are not clearly separated in real adaptive hypermedia systems. User-friendly tools efficiently supporting the complex process of authoring adaptive hypermedia applications are difficult to find. To simplify the authoring process we need reusability at various levels as well as interoperability between different platforms. The first overview of adaptive hypermedia authoring tools is relatively fresh [1]. Authoring adaptive educational hypermedia is more difficult in comparison with the ordinary hypermedia, as the authors have additionally to create the knowledge structure and its interconnection with the educational materials. The state of the art in this area has become a theme of a specialized workshop [3]. The main objective is to simplify the authoring process. This can be achieved by reusability not only on the level of learning objects, but also concerning adaptation techniques and pedagogical approaches. Better understanding and formulation of possible patterns in the authoring process can help. A pattern describes an often repeated problem and its solution that can be used always when the problem occurs. Collaborative authoring issues have been seldom addressed [10]. Some tools support authoring on the markup language level (e.g. AHA! [7] by conditional comments in HTML pages), other represent knowledge in the form of teaching tasks, defining their composition by rules (the TANGOW [2] approach). Just a few tools focus primarily on the simplification of the authoring process, without the necessity of programming skills, and provide form based user interface (NetCoach [16] is one of them). ALE [10, 11] offers template based user interface to make the authoring process more intuitive. The systems vary in following the formal models and separating individual layers. This is most critical in the case of the adaptation model, where there is no known satisfactory solution yet. To specify adaptation some tools use a markup language directly in the content (e.g. AHA!), other encode it in the learning environment (e.g. ALE [12]). However, such adaptation specifications are not reusable.

4 Stakeholders

Table shows various parts of an adaptive course together with different professions participating in their authoring. Ideally a typical teacher as the author of an adaptive course should fulfill mostly the composer and annotator role. Although it does not exclude writing of complementary explanatory texts and simple authoring of content fragments (e.g. photos or schemas). So the author chooses from the available basic building blocks those that suit best for the current purpose and composes the individual learning objects from them. The graphical designer creates an appropriate layout for them. Another role of the author is annotation of the learning objects, which should be as automatic as possible. Note that the annotator does not have to be the same person as the composer, as the author can reuse a learning object and change just its metadata, for instance if it is to be used for a different target group.

Table 1. Authors of adaptive educational hypermedia

Content fragments	Expert
	Multimedia author
Learning objects	**Composer**
	Graphic designer
Metadata	**Annotator**
Adaptation strategies	Instruction designer
Exercises (Homework)	Student
Assessments	Tutor

Specification of adaptation strategies should be a task for a specialized instruction designer, taking into account especially the student's learning style. The strategies defined in this way should be universal, thus reusable in various courses and the teacher would just choose the most suitable one from the available offer. Usually also students contribute into a course by exercises (homework) and tutors by their assessments. Neither tutors have to be identical with the authors – an author might be a professor and atutor his assistent. To realize this, it is necessary to separate adaptation strategies from the course content, but on the other hand to interconnect them with the used metadata and context attributes.

5 System Architecture

Following the Dexter model, the architecture of hypermedia systems consists of three levels: *Physical level* (storage, shared data, network access), *Logical level* (nodes and links), and *Presentation level* (user interface). Based on our experience [11], we are proposing a technique for specification of adaptation strategies. It enables flexible and context dependent presentation of educational hypermedia. Similarly to AHAM, we suggest that an efficient adaptive hypermedia system contains the following parts:

- *Content management* – maintains the domain model (learning objects with metadata, semantic concept networks / ontologies) and supports the authoring process (separation of content and layout, reusability, semi-automatic annotation)
- *Context management* – includes generic user modelling (preferences, learning styles), enabling reusability and sharing of the model by various applications
- *Adaptation management* – corresponds to presentation specification in the Dexter model; deals with adaptation specifications, including instruction management (learning strategies); adaptation specification is application independent and reusable, it can be specified without programming skills

In [10] we have proposed an adaptation strategy specification technique with the objective to separate the content of educational materials from the knowledge driving the adaptation process, which includes the instruction. In general, each of these parts can have a different author. Similarly, it is good to separate also the declarative and procedural knowledge, as the first one can be specified as attributes and metadata also by authors without the programming skills. From our point of view an adaptation strategy specifies how the individual objects (learning objects or content fragments) should be presented by the system based on their attributes and the current parameters of the learner (user) model, or more generally of the context model.

6 FOSP Method

Our WINDS experience has shown that authors without technical background can specify declarative instructional knowledge. There exist many repositories containing milions of learning objects with metadata (e.g. Cisco, Ariadne). This fact should be taken into account when designing authoring methods and tools. The *Learning Object Metadata* (LOM) standard defines alearning object as any entity, digital or non-digital, that may be used for learning, education or training [13]. Content models identify different kinds of learning objects and their components. Acomparative analysis of six known content models [15] led to the creation of a general model that includes the existing standards and distinguishes between:

- *Content fragments* – learning content elements in their most basic form (text, video), representing individual resources uncombined with any other; instances
- *Content objects* – sets of content fragments; abstract types
- *Learning objects* – aggregate instantiated content objects, add a learning objective

To illustrate our method let us consider the following first. When a teacher wants to teach a learner certain new knowledge or skill, he usually first decides what types of learning resources are suitable for the particular user, e.g. for one learner it can be a definition and an example, for another a demonstration and an exercise. Then he should order the resources, i.e. decide whether to start with the definition or the example. Each learning resource can have alternative representations, so the teacher has to select the most suitable one – narrative explanation, image, animation, video, etc. This illustrates the basic reasoning behind our method, which takes into account also different presentation opportunities of various devices. Note that we are proposing a technique to specify an adaptation strategy, not an adaptation strategy itself. Specification of adaptation strategies is a task for instructional designers.

In our approach an adaptation strategy maps the domain model (learning objects with attributes and metadata) and the context model (including the learner model with learning styles and preferences) onto the course presentation for the learner. To be more concrete we define the following sets:

- *Role* – the pedagogical role of the object (e.g. definition, example, demonstration)
- *Style* – the learner's learning style (e.g. intuitive – sensitive, active – reflective)
- *Media* – the media type (e.g. text, image, audio, video, animation)
- *Context* – the usage context (e.g. multimedia desktop, mobile device)

The proposed adaptive strategy is based on these functions:

- *Weight*: Role × Style ? Integer
- *Sequence*: Role × Style ? Integer
- *Alternative*: Media × Style ? Integer
- *Threshold*: Style ? Integer
- *Granularity*: Context ? Integer

The *Weight* function represents the relevancy of the pedagogical role for the learning style. The *Sequence* function defines the order for the presentation of the role for the learning style. (Note the difference between these two: an introduction does not have to have the highest relevancy, but when selected it should be the first. The selected components do not have to be ordered according to their relevancy.) The *Alternative* function expresses the relevancy of the media type for the learning style. The *Threshold* function sets the threshold for the object display based on the learning style. The *Granularity* function specifies the maximal number of objects presented at once for the context. The proposed adaptation strategy consists of four operations:

- *Filter*: for the current object it selects just those components that have their *Weight* greater than *Threshold*
- *Order*: this sorts the selected components according to the *Sequence* value
- *Select*: from the alternative components it chooses that one with the highest *Alternative* value
- *Present*: it displays the componets taking into account the *Granularity* value

So to define a pedagogic strategy for a certain learning style the instruction designer needs to specify the functional values of *Weight*, *Sequence*, *Alternative*, *Threshold*, and *Granularity* for different types of learning objects (i.e. content objects). But it is not necessary to define all the values. If no value is specified a default one will be applied: 0 for *Weight*, the minimum value for *Threshold* and the maximum one for *Granularity*. The basic operations *Filter*, *Order*, *Select* and *Present* are interpreted by the system. According to their first letters we call this method *FOSP*. To implement this method in practice it is crucial to choose suitable visualization of the functional values and an intuitive user interface to control them.

7 Summary

In this paper, we have proposed a method how to specify adaptation strategies in adaptive hypermedia applications. The key idea of the method is to simplify the complex authoring process for teachers. Collaborative authoring is supported by sharing of partial results between various authors that participate in the development of adaptive hypermedia. This approach is compliant with the established standards and recommendations, including the AHAM reference model. Specification of adaptation strategies by separating the content, declarative and procedural knowledge in adaptive courses is quite natural and similar approaches have been applied in related areas.

References

1. Brusilovsky, P.: Developing Adaptive Educational Hypermedia Systems: from Design Models to Authoring Tools. In: Murray, T., Blessing, S., Ainsworth, S. (eds.): Authoring Tools for Advanced Technology Learning Environments. Kluwer (2003)
2. Carro, R.M., Pulido, E., Rodríguez. P.: TANGOW: a model for Internet Based Learning. IJCEELLL **1-2** (2001)
3. Cristea, A.: Authoring of adaptive and adaptable educational hypermedia: Where are we now and where are we going? Proceedings of the WBE 2004 Conference. IASTED (2004)
4. Cristea, A., Calvi, L.: The three Layers of Adaptation Granularity. Proc. of the International Conference on User Modelling. Springer (2003)
5. Cristea, A., de Mooij, A.: LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. Proc. of WWW 2003 Conference. ACM (2003)
6. De Bra, P., Houben, G. J., Wu, H. (1999). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. Proc. of the ACM Conference on Hypertext and Hypermedia. ACM (1999) 147-156
7. De Bra, P., Ruiter, J.P.: AHA! Adaptive hyermedia for all. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet 2001 Conference. AACE (2001) 262-268
8. Jones, A. et al.: Grand Research Challenges in Information Systems. Computing Research Association (2002)
9. Halasz, F., Schwartz, M.: The Dexter hypertext reference model. Communications of the ACM **2** (1994) 30—39
10. Kravcik, M., Specht, M.: Authoring Adaptive Courses – ALE Approach. Proc. of the Web Based Education 2004 Conference. ACTA Press (2004)
11. Kravcik, M., Specht, M., Oppermann, R.: Evaluation of WINDS Authoring Environment. Proc. of the Adaptive Hypermedia 2004 Conference. Springer (2004)
12. Kravcik, M., Specht, M.: Flexible Navigation Support in the WINDS Learning Environment for Architecture and Design. Proc. of the Adaptive Hypermedia 2004 Conference. Springer (2004)
13. LOM: *IEEE* Standard for Learning Object Metadata. IEEE (2002)
14. Rada, R.: Hypertext: From Text to Expertext. McGraw-Hill (1991)
15. Verbert, K., Duval, E.: Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models. Proceedings of the ED-MEDIA 2004 Conference. AACE (2004)
16. Weber, G., Kuhl, H.-C., Weibelzahl, S.: Developing Adaptive Internet Based Courses with the Authoring System NetCoach. In: Reich, S., Tzagarakis, M.M., De Bra, P. (eds.) Hypermedia: Openness, Structural Awareness, and Adaptivity. Springer (2001) 226-238

Visualizing Adaptivity for Teachers: an Authoring Tool for Designing Educational Adaptive Websites

Jacopo Armani[1]

[1] Institute of Communication Technologies
Università della Svizzera italiana, Via G. Buffi 13, 6900 Lugano, Switzerland.
jacopo.armani@lu.unisi.ch
http://www.istituti.usilu.net/armanij

Abstract. So far educational adaptive technologies have proven their effectiveness only in small-scale lab courses, thus they still wait for being released to the large community of educators. Among the reasons, it is the difficult task of designing and authoring a highly interactive adaptive course, especially for non-technical group of educators. This paper presents the preliminary results of a research on the development of a user-friendly authoring environment for designing web-based adaptive courses. This work is part of a larger research effort, which is aimed at merging the most established adaptive techniques with the state of the art of the Learning Management Systems.

1 Introduction

Adaptive technologies in the field of education have proven so far their effectiveness only in small lab experiments, thus they are still waiting for being presented to the large community of educators. First of all, as pointed out by some recent studies [4], adaptive educational hypermedia systems are difficult to design, set-up, and implement, due to the high technical competencies they require to master them. In particular, all of the (few) existing general purpose adaptive educational systems have a steep learning curve, that forbids a non technical teacher to autonomously create a course.
More generally speaking, the main issues that hinder the spread of many of the available adaptive systems in community of educators are:

1. High technical competencies to set up an adaptive course (i.e. writing of XML descriptors, textual configuration files, …);
2. Difficulty in specifying in the system language the interactions that must occur between the user and the system (i.e. definition of concept networks, condition statements, resource indexing, …);
3. Lack of ready-to-use patterns that exploit frequent adaptive teaching strategies.

Despite this situation, different researchers point out the importance of adaptivity in the definition of effective learning scenarios. For example, studies from the Instructional Design field [13] show how much the adaptive instruction paradigm has been, and still is, a common trait in every day instructional situation: a teacher in a classroom naturally adapts his/her learning goal, presentation style, instructional strategy, and language to match the needs of his/her class, thus, why this can not happen online too?

2 Literature Review

In the last decade, several domain-independent Educational Adaptive Hypermedia Systems (EAHS), with different degrees of authoring capabilities, have been proposed. Some important examples are Interbook [6], Net-Coach [14], and AHA! [7]. Yet, all of them have been used quite always within the research group who developed them (with very rare exceptions, actually the Author of this paper has experimented with AHA! to design a set of adaptive courses to assess the usability of the tool, cf. [1]).

Moreover, the task of building authoring interfaces for adaptive systems has been recognized as one of the main reason for the small diffusion of EAHS so far [4]. In this direction, some researchers [3] are trying to develop intelligent authoring interfaces that can help the instructors in their tasks of designing an AEHS, by providing them with adaptive and intelligent support. On the other hand, many EAHS rely on interfaces that are still textual or, in the best cases, form based.

The idea of design patterns for adaptivity is not new (cf. [15], [16]). For instance, the SCORM consortium suggests over twenty different kind of learning patterns that have, at different degrees, some kind of adaptivity: from simple personalized learning environment, to advanced trial-and-error environments. Surprisingly the practise of design pattern has not even began yet, even though it is doubtless that it would be extremely beneficial for educators and, in the end, for learners. The main reason is because, until now, educators lack of right tools to do that. By the way, according to past studies [9] this feature seems to be very relevant for supporting educators unleashing their creativity indeed.

3 Introduction to ADLEGO

ADLEGO is a low-level adaptive rule-based engine. In ADLEGO an educational website is organized in resources. Each resource provides access to content which can be pulled from an external site or a local directory on the server. Moreover a resource comes with a set of presentation and tracking rules. The former are inspected by ADLEGO to rendering the resource itself and the links from one to another resource. The latter are used for sending to the user model information about the user's behavior within the resource.

The system supports basic actions that can be performed on different properties of a resource. One can grant or deny access to the resource, or perform operations on the incoming or outgoing links of it. Those operations include: hiding, disabling, adding an icon or a text note, setting a color or a style. Despite this basic set of actions that are available, ADLEGO allows indeed to set up many of the established adaptive techniques (for the full taxonomy refer to[5]), namely adaptive link annotation, adaptive link hiding, adaptation of modality, adaptive text and multimedia presentation, and map adaptation.

4 The Visual Authoring Environment

The Authoring Tool for ADLEGO is a visual multi-window interface that presents in a integrated view the different facets of an adaptive website: the hypertext structure, the content, the adaptive interaction model, the user model.

The left panel gives an overview of the resources. ADLEGO support a traditional tree-like structure as many commercial Learning Management Systems. Though, it is possible to develop a more complex hypertext structure of resources, by connecting them with hyperlinks. For this reason a complementary graph view of the course is provided. Within this interface the educator can add new resources and put them inside the tree with a simple drag & drop mechanism. Moreover some visual cues inform about the nature of content of the resource (internal /external content or content to be provided).

Clicking on a resource in the left panel will open the corresponding resource detail in the top right panel (see top right of Fig.1a). In this panel a preview of the current page is presented, displaying its title and content. Moreover, the rules that have been specified for that resource are listed on the left and right sides. As anticipated in the former chapter, they can be either presentation or tracking rules (respectively in the left and right column).

Each rule is displayed with an icon that represents the action that is performed when the rule condition applies. For link rules an additional cue informs whether the rule applies to incoming or outgoing links.

Clicking an icon brings up a contextual menu to sorting, editing, or deleting the corresponding rule. To edit a rule a visual Rule Editor is provided (see Figure 1b). The Rule Editor is split in two versions: a basic and an advanced one. The former interface has been designed to help the teacher in the task of writing simple rules, in this way reducing the likelihood of errors during the formulation of complex Boolean conditions. For this reason the basic interface allows to define an array of conditions which are connected by *and* clauses only. This approach stems from past studies on the difficulties encountered by non-programmers in the statement of Boolean conditions [11][12], and it aims to cope with the reported common misunderstanding of *not* and *or* Boolean clauses by non technical people. Moreover only a subset of actions is available, along several macro rules, thus fostering the instructor on reasoning on how to present the resource itself, without being distracted by technicalities.

Conversely, the advanced interface supports the full expressive power of the ADLEGO rule engine, yet providing a less error-prone environment for writing complex rules. This result is accomplished by a set of panels, which allow to quickly build a condition without knowing exactly the underlying syntax of each terms. The instructor simply clicks on a variable icon and the interface fills the condition window with the corresponding statement according to the right syntax.

Fig. 1. (a) - Interface of the Visual Authoring Tool; (b) - Rule Editor: advanced panel

Finally, the expert user can also type a condition using the ADLEGO syntax and test it with the syntax checker.

A similar interface is provided for writing tracking rules, but with a different set of actions that can be performed: setting, incrementing, decrementing the value of a variable, etc.

Since the authoring tool is still under development, some features have not been implemented yet. In the near future, the tool will allow to store and retrieve sub-sets of resources, along with their rule sets, to foster reusability. This feature will pave the road to the development of design patterns, which an educator could easily store and reuse in his/her different projects.

5 Discussion

According to results from HCI field [10], four minimalism principles should be taken into account for the design of an authoring tool:

1. Choose an action-oriented approach;
2. Anchor the tool in the task domain
3. Support error recognition and recovery
4. Support ready to do, study, and locate

Taking them in account means basically asking ourselves a question: what should be the border between flexibility and usability of the interface? There is no an easy answer to that. In our project, we have narrowed down the spectrum of the targeted end users of our tool to be able to come to a conclusion. Our research framework is that the design process of an adaptive educational website is under the responsibility of an instructional designer with no programming background. This assumption leads to reduce as much as possible the richness of the interface, and therefore of the design blueprint that can be produced with it. On the other hand, it is evident that we can not impoverish over a certain threshold the set of tools which an educator should be able to use. This issue still requires more debate, and we suggest that interesting results could be gained from the empirical evaluation of usability of both adaptive courses and their authoring tools as well.

Another insight we had is that making use of strong visual metaphors for helping the educator understanding all the aspects of an adaptive system, would improve the usability of the authoring tool in the end. Yet, what is the best metaphor to achieve this? The current project makes use of the well known “desktop” metaphor to build up on the existing conventions that we all share by the nature of modern window based operative systems. Other metaphors could suit the case as well.

Moreover, since the design of an user-system interaction requires a time consuming task for testing that all designed behaviors actually occur, some kind of simulation or debugging tools should be provided to asses it in a controlled and organized way.

The current implementation of the authoring tool has been designed to suit the needs of the specific adaptive engine, namely ADLEGO, however a more general approach could be explored as well. The next generation of authoring tools could be able to produce a more application-independent model, and could provide compiling tools to translate it into the major adaptive engines (i.e. the same authoring interface could compile the same model of adaptive course in either the Interbook , AHA, or ADLEGO syntax).

Finally a more complex scenario arises when we think in the perspective of web services. Imagine a network of: adaptive servers (executing a different adaptive engine each), user models, and content and service providers. In this context the authoring and design needs cannot be easily integrated in a holistic interface anymore. But still, the designer of the learner experience (our educator), needs to have access to all of them, and perhaps even modify parts of them.

6 Conclusions

A visual authoring tool to design an adaptive website seems to be the missing key to unfold the hidden potentiality of adaptive technologies for education. The importance of visualizing all of the different aspects of an educational adaptive systems is even more evident when we think at the community of instructional designers and teachers as our main target audience. Those members do not usually have the required technical skills to unleash the power from an adaptive engine, even though it has been designed with simple features (i.e. conditional fragments, adaptive link annotation only).

A prototype of a visual authoring tool for instructors has been discussed in this paper. The authoring tool allows a teacher to structure the website in a network or a tree of resources, connect them with the actual content (both locally stored or coming from an external website), and define the necessary rules to present it in an adaptive form to the learners, just using simple drag & drop mechanisms. Moreover every part of the adaptive course being developed is shown to the author in a visual form, in order to support them thinking and manipulating the usually abstract objects that constitute an adaptive website (concept networks, user model variables, etc.). We are currently setting up an evaluation test for the presented authoring tool in order to assess its impact[2].

References

1. Armani J., Botturi L.: Adaptive Hypermedia System Design: a Method from Practice. Proceedings of ICWI 2003, Algarve, Portugal, (2003), 843–848.
2. Armani J.: Shaping Learning Adaptive Technologies for Teachers: a Proposal for an Adaptive Learning Management System. Submitted to ICALT '04, (to appear)
3. Aroyo L., Dicheva D., and Cristea A.: Ontological Support for Web Courseware Authoring. Proceedings of ICALT '02, Las Vegas, US, (2002).
4. Brusilovsky P.: Developing Adaptive Educational Hypermedia Systems: from Design Models to Authoring Tools. In: Murray T., Blessing S., & Ainsworth S. (eds.): Authoring Tools for Advanced Technology Learning Environment, Dordrecht: Kluwer Academic Publishers, (2003), 377–410.
5. Brusilovsky P.: Adaptive hypermedia. In: Kobsa A. (ed.): User Modeling and User Adapted Interaction, Ten Year Anniversary Issue 11 (1/2), (2001), 87–110.
6. Brusilovsky P., Eklund J., and Schwarz E.: Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, 30 (1-7), (1998), 291–300.
7. De Bra P., and Calvi L.: AHA! An Open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, 4, Taylor Graham Publishers, (1998), 115–139.
8. EduTools: Providing decision-making tools for the E-D-U community, Url: http://www.edutools.info/course/, (last visited on Feb 20th, 2004).
9. Finzer W. F., and Gould L.: Rehearsal World: Programming by Rehearsal. In Cypher A. (ed.): Watch What I Do – Programming by Demonstration. The MIT Press Cambridge, Massachusetts, London, England, (1993), 79–100.
10. Hackos, J.T.: An Application of the Principles of Minimalism to the Design of Human-Computer Interfaces. Common Ground 9, (1999), 17–22.
11. Pane J.F., Ratanamahatana C.A., and Myers B.A.: Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems. International Journal of Human-Computer Studies, Vol. 54:2, (2001), 237–264.
12. Pane J.F., and Myers B.A.: Tabular and Textual Methods for Selecting Objects from a Group. Proceedings of VL 2000: IEEE International Symposium on Visual Languages, Seattle, WA: IEEE Computer Society, (2000), 157–164.
13. Park O., and Lee J.: Adaptive Instructional Systems. In: Jonassen, D.H. (ed.): Handbook of Research on Educational Communications and Technology, 2nd ed., AECT, (2004).
14. Weber G., Kuhl H.-C, and Weibelzahl S.: Developing adaptive internet based courses with the authoring system NetCoach. In: De Bra P., Brusilovsky P., and Kobsa A. (eds.): Proceedings of Third workshop on Adaptive Hypertext and Hypermedia, July 14, 2001. Sonthofen, Germany, Technical University Eindhoven, (2001), 35–48.
15. Avgeriou, D. Vogiatzis, A. Tzanavari, S. Retalis: Design Patterns In Adaptive Web-Based Educational Systems: An Overview. Proceedings of Web Education Conference 2004, 11-14 February 2004, Innsbruck, Austria, (2004).
16. SCORM. Best Practices Guide for Content Developers, 1st Edition, (2004). Available online at: http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/

[2] The present work has been carried on under the supervision of Prof. Brusilovsky (School of Information Sciences, University of Pittsburgh, PA, USA).

Authoring processes for Advanced Learning Strategies

Jana Trnková[1], Wolfgang Theilmann[2]

[1] Telecooperation Research Group, TU Darmstadt, Germany
jana@tk.informatik.tu-darmstadt.de
[2] SAP Research, CEC Karlsruhe, Germany
wolfgang.theilmann@sap.com

Abstract. The paper provides an overview on the authoring process in the adaptive learning environment with support of advanced leaning strategies. We explain the concept of the metadata model and adaptation process implemented in the L^3 learning environment. We provide a summary of experiences from different projects using L^3, point out the most common difficulties during the authoring process and describe our future plans to support authors.

1 Introduction

Authoring process in e-Learning is a complex problematic, which involves also specific didactical, design related and technological issues. Modern approaches as adaptability and reusability make this process even more complicated. A good e-Learning course is typically a result of team cooperation. Authoring environment has an important role to make this process smoother and help to bridge the gaps between the different areas of expertise. Therefore, recently many learning environments are coming up with integrated pedagogical concept. However, very often the authors seem to be only more confused and pushed to use a certain didactical approach. We believe that support of the authors should include these two aspects: the direct support of authoring process (e.g., improving of user interface and integration of templates) and adaptability of pedagogical concept to the specific requirements of the author.

In our paper we describe the authoring process based on the concept of advanced learning strategies. This concept had been implemented in L^3 learning environment (later adopted by SAP Learning Solution) and tested in few projects. We provide the overview on the experiences with this approach and identify the main challenges of authoring processes. Finally we introduce our ideas for improving the current authoring environment and the areas of our future research[3].

2 Basic Authoring Process

This section describes the underlying basic approach for structuring and authoring learning content. Starting with the course model we explain how the concepts of learning strategies influence the authoring process.

[3] The research in this project is financed and supported by SAP AG.

2.1 Course Model

One of the major goals in L^3 is to provide the methodology and the tools to structure learning material in a way that allows for both, reusability and adaptive delivery.

Content Aggregation Model. In L^3, content is aggregated in four distinct structural levels where each higher level may contain instances of all lower levels. The lowest level of granularity is formed by *knowledge items* which represent the smallest indivisible element in a course. Each knowledge item shall contain material that illustrates, explains, practices or tests a certain aspect in one thematic area and thus refers to actual learning content. Several related knowledge items are typically assembled into one *learning unit*, which is the logical representation of such a distinct, thematically coherent unit. Learning units are still considered small in terms of "size" (i.e. duration) and are further grouped into larger structural units, so-called *sub courses*. Sub courses may also be used to build an arbitrarily deep nested structure by including other sub courses. At the highest structural level sub courses, learning units and knowledge items are contained in a *course*.

Meta-data. Besides structural composition, course material in L^3 can be tagged with additional meta-data that further improves the support for adaptive delivery, reusability, and search and retrieval of existing material.

The meta-data set used in L^3 can be divided into four categories:

1. *Instructional meta-data*. The L^3 authoring tool allows authors to attach the full Learning Object Meta-data (LOM) set to individual course elements.
2. *Knowledge types*. Receptive knowledge items can be categorized using a didactical ontology defined in [1]. At the topmost level, it distinguishes between orientation knowledge ("know what a topic is about"), action knowledge ("know how"), explanation knowledge ("know why something is the way it is") and reference knowledge ("know where to find additional information"). These four basic types are further sub-divided into a fine grained ontology. Furthermore, knowledge items can represent also a (performance) test which may have implications on the competencies a learner has mastered (see below).
3. *Relations.* While assembling the higher level building blocks of an L^3 course, an author may specify relations between elements. Matter of facts relations describe the dependencies strictly on a subject level (e.g., "part of"). Didactic relationships describe restrictions for the delivery to the learner (e.g. "prerequisite"). Again, both types of relationships are sub-divided into a fine grained ontology [1].
4. *Competencies.* Performance evaluation is an integral part of the L^3 learning platform. Course authors can assign competencies (or skills) to learning material and can provide test procedures to evaluate the individual learner's performance.

Strategies and Navigation. Sequencing is deliberately omitted from our content aggregation model, thus allowing different sequencing rules to be applied to the same course material: One strategy might start at the bottom, i.e. the specifics, moving up to the more general concepts, which resembles an *inductive strategy*. At the opposite end, another strategy may lead the learner from the general concepts to the specifics, thus implementing a *deductive strategy*.

The computation of a learner's path through the material is divided into two steps:

To navigate between higher-level elements (sub courses and learning units), a strategy used focuses on the matter of facts relations defined by the author. One can think of this strategy as moving along the different topic areas. In other words, it operates on a macro level, thus the term *macro strategy* is used in the rest of this paper.

Opposed to that, a different strategy is used when entering a specific topic enclosed in a learning unit. Here are no matter of facts relations between the knowledge items, but the items are tagged with different knowledge types. The strategy determines a didactical approach taken to present the topic specific knowledge to the learner. E.g., an "action oriented" strategy may present any action items, before it moves on to the other items, whereas an "overview only" will present orientation knowledge while ignoring all other items. This part is termed *micro strategy*.

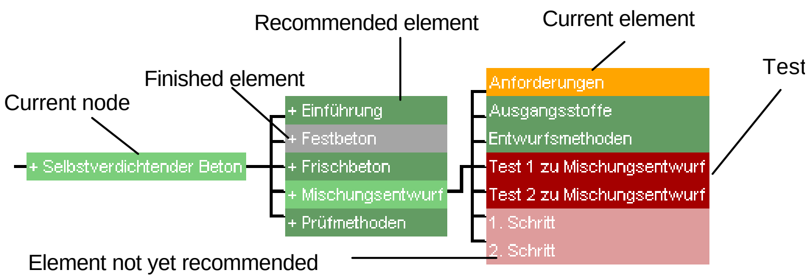

Fig. 1. Example of navigation path generated by L^3

At the beginning of learning session learner chooses micro and macro strategy. Based on this choice and pedagogical metadata set by author, the recommended order of learning elements is suggested. This can be followed by clicking on the navigation button *Forward*, or it is possible to display a navigation path in the bottom part of the window and simply click on the desired element. Navigation path also gives an orientation about visited and recommended elements as it is shown in Fig. 1.

2.2 Authoring Process

For the author is a course represented as a set of graphs. A *node* represents a structural unit of a course and node attributes carry the meta-data attached to the corresponding unit. An *edge*, in turn, represents a relation between two structural components.

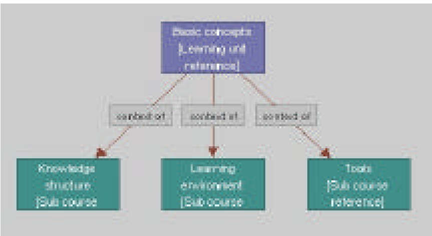

Fig. 2. Example course: "Authoring in L^3"

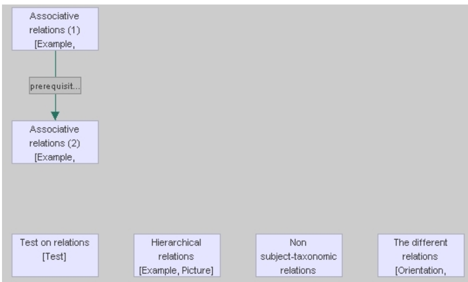

Fig. 3. Learning Unit "Relations"

The following example may illustrate this: The author has divided a course about "Course authoring in L^3" into three sub courses and one learning unit. She also decided that the concepts explained in the learning unit provide the context for the concepts covered in the three sub courses (see Fig. 2).

The learning unit "Relations" introduces the different relation types, gives examples, provides further explanations on "non-subject taxonomic" relations, and contains a "Test on relations". In this unit, the author has decided to declare the first example about "Associative relations" as a prerequisite of the second example (see Fig. 3).

3 Experiences within Pilot and Customer Projects

This section reports experiences made with the authoring and using learning strategies within various project contexts – from research pilots to a commercial product.

3.1 Research Pilot L^3

The approach of authoring learning content and applying learning strategies on that content at runtime was first used within the German lighthouse research project L^3. A consortium of 20 companies headed by the SAP Research was developing and establishing a national backbone for advanced education and training. L^3 aims to make lifelong learning possible by implementing an organizational and technical infrastructure that can be used by everyone, for professional and private education.

Within L^3 the authoring environment was used and evaluated by professional content developers. The experiences from the authors showed that the structuring and tagging of instructional units is a highly complex task and that content creators need support to introduce this to their organization and to deal with the methodology. A two days introduction with hands-on sessions has been developed and results in a reasonable learning curve so authors have been able to create and/or reengineer first domain fragments. More specific they have to abandon the "traditional" way of "hard-wiring" courses, but design self-contained learning units to allow flexible assignment and reusability. The overhead introduced through the initial creating process has been measured to 5-10% depending on the experience of the author. Some authors still tend to create "hard-wired" courses by using many prerequisite relations. However, their content got lower ratings from the students because of the lack of redundancy. Self-contained learning units that cannot assume that other units have been already consumed automatically lead to a higher redundancy level.

3.2 WiBA-Net

WiBA-Net Project [3] was a German e-learning project for architects and civil engineers, supported by the German Ministry of Education and Research and SAP AG. It was a multi disciplinary multi-site project involving six Universities in Germany, headed by the domain expert group of Prof. Grübl (Civil Engineering Dept., Darmstadt University of Technology). The project has a web-based interface for students and educators, a WiBA-Net Portal. In the background, the portal consists of few mostly independent components, which are even located on different computers. Since we were trying to create a compact and easy-to-use environment, this modularity remains hidden to the end-user. Since Pedagogical Department of TU Darmstadt (Germany) has been one of the project partners, we had been receiving feedback right from the beginning of the project. First suggestion came up from the students, which were missing an overview of visited materials and materials which still need to be seen. Actually this information was available from the content overview but this has been shown not to be sufficient. Students want an overview of their progress continuously, without any clicking. Therefore we have implemented displaying a number of the visited pages and all the pages together (see Fig. 4).

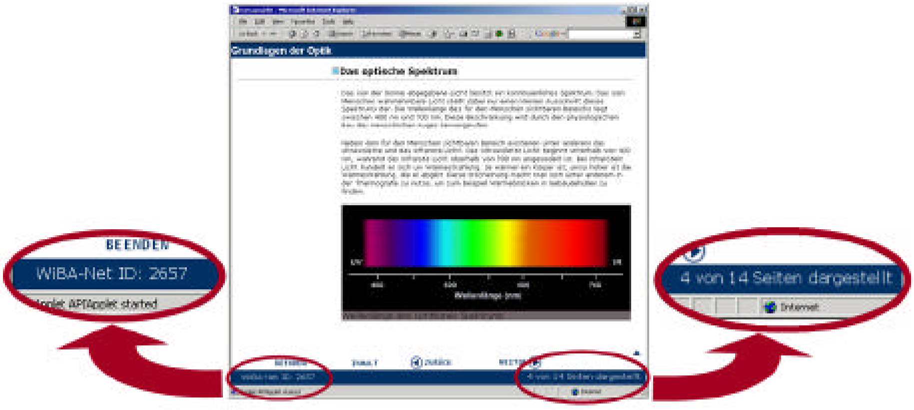

Fig. 4. WiBA Net Project: Extensions of original L^3 environment

Another requirement came from teachers, which were willing to receive a feedback on content from learners. In order to achieve a comment on a particular content page, the WiBA-Net ID number (from database of knowledge network) must be provided. Thus we extracted the ID number and displayed it in the course interface (see Fig. 4).

3.3 Commercial Product SAP Learning Solution (SAP LSO)

The SAP LSO [5] is a commercial product build upon the concepts of L^3. It realizes a comprehensive solution for blended learning tightly integrated to an Enterprise Resource Planning (ERP) system. Learning activities and results can be correlated with the ERP module Human Resource (HR). The SAP LSO is primary targeting corporate learning scenarios. The content used by customers of the SAP LSO mainly stems from 2 sources. General purpose content (e.g., courses about office products) is typically bought from external professional content developers. Specific content about a company's core business is typically produced within the company. External content is mainly offered by using the SCORM packaging format [4]. Such content can be imported and converted into the internal format.

4 Evaluation and Future Research

Previous projects showed that adaptability and learning strategies are a very complex topic. The big advantage of flexibility for the learners is becoming a problem for the authors. We have recognized following topics as the main obstructions during the authoring process:

- Creating of self-contained reusable learning units
- Predicting the behaviour of the course under different learning strategies
- Understanding of different pedagogical concepts and creating a reasonable structure for the course – adaptive to the specific learning needs of student

Templates. The pedagogical power of L^3 allows the authors to create a very sophisticated courses based on modern didactical approaches. On the other hand, this requires an experienced user with a strong pedagogical background. Less experienced users are advised to use the templates. We would like to improve current (relatively simple) template manager and implement an advanced tool, which will allow, besides creating a template of a course structure, also to define own learning strategy. The open question is how to keep the plurality of learning process and adaptability together with support of specific learning strategies defined by authors. We are considering incorporating into the template editor also the choice to allow certain strategies to use with the template. Another open issue is how to assist authors in enhancing their course material to be applicable for several strategies.

Strategy Visualization. One of the lacks of L^3 is insufficient transparency of the final design of the course under the different macro and micro strategies. The authors have difficulties to predict the behavior of the course under the different strategies. We plan to integrate an improved tool for the strategy visualization. It should:

- allow switching between the different strategies,
- provide an overall view on structure of the course,
- be able to simulate viewing the course (visibility of learning elements to student, whether is learning element recommended at certain stage).

Standard Conformance. The most relevant standard for e-Learning lately becomes SCORM [4]. Last version (SCORM 2004) introduced a new concept of Sequencing and Navigation. The Sequencing and Navigation is, of course, an important step for support of adaptability and learner-centred approach. Nevertheless, the implementation of sequencing rules is very closely connected to the structure of the particular course – it doesn't allow definition of general rules related to the metadata, but only the rules tightly connected to the concrete learning units. The sequencing means are very much on programming level instead of at pedagogical level. This approach is rather different from ours.

References

1. Meder, N. (1999) Didaktische Ontologien. In Proc. 6. Tagung der Deutschen Sektion der Internationalen Gesellschaft für Wissensorganisation (Hamburg, Germany, September 23-25, 1999), pp. 401-416. (in German)
2. Rößling, G., Trnková, J., Langendorf, U., Tillack, K., Mühlhäuser, M. (2002): Learner-Centric Online Teaching for Non-Computer Science Students. Proceedings of ED-MEDIA 2002. Denver, Colorado. AACE Press, Charlottesville, VA. pp. 1645-1650.
3. SCORM definition and resources. Online: http://www.adlnet.org/index.cfm?fuseaction=scormabt (seen May 12, 2004)
4. SAP Learning Solution. Online: http://service.sap.com/learningsolution (seen May 12, 2004)

Adaptive Hypermedia Authoring: From Adaptive Navigation to Adaptive Learning Support

Pythagoras Karampiperis [1, 2] and Demetrios Sampson[1, 2]

[1] Department of Technology Education and Digital Systems,
University of Piraeus,
150, Androutsou Street, Piraeus, GR-18534, Greece
[2] Advanced e-Services for the Knowledge Society Research Unit,
Informatics and Telematics Institute,
Centre for Research and Technology Hellas,
42, Arkadias Street, Athens, GR-15234 Greece
{pythk, sampson}@iti.gr

Abstract. Educational hypermedia systems seek to provide adaptive navigation, whereas intelligent web-based learning systems seek to provide adaptive courseware generation. The design of powerful authoring frameworks by merging the authoring approaches used in the above mentioned systems is recognized as one of the most interesting questions in adaptive web-based educational systems. In this paper we address adaptive hypermedia authoring proposing an authoring framework that combines the approach of automatic courseware generation with the paradigm of educational hypermedia systems based on the use of ontologies and learning object metadata.

1 Introduction

Educational hypermedia systems seek to provide adaptive navigation, whereas intelligent web-based learning systems seek to provide adaptive courseware generation. Adaptive navigation seeks to present the content associated with an on-line course in an optimized order, where the optimization criteria takes into consideration the learner's background and performance on related knowledge domain [1], whereas adaptive courseware generation is defined as the process that selects learning objects from a digital repository and sequence it in a way which is appropriate for the targeted learning community or individuals [2]. The need for gradual merge between the authoring approach of adaptive educational systems and the authoring approach of adaptive hypermedia systems has been already identified in literature [3].

In this paper we address adaptive hypermedia authoring proposing an authoring framework that combines the approach of automatic courseware generation with the paradigm of educational hypermedia systems. The paper is structured as follows. Initially, we survey the adaptive techniques used in educational hypermedia systems, classifying them in two main classes namely adaptive presentation and adaptive navigation techniques. The second part discusses the main steps in the adaptive educational hypermedia design process and presents the abstraction layers of adaptive hypermedia authoring process proposing an authoring framework that enables the definition of learning objectives and automatic authoring of adaptive activities. This framework is based on the use of pedagogical templates which include the rules for adaptive navigation and can be processed by an adaptive content selection mechanism in order to serve adaptive web-based courses based on a diverse set of pedagogical strategies. The selection of learning path takes into consideration learner's cognitive characteristics and preferences.

2 Adaptivity in Educational Hypermedia Systems

In the literature there are several adaptive techniques employed in educational hypermedia systems that can be classified in two main classes, namely:

- Adaptive Presentation. The goal of the adaptive presentation techniques is to adapt the web-based content to the user's goals, knowledge and other information stored in the user model [3].
- Adaptive Navigation. Adaptive navigation seeks to present the learning objects associated with an on-line course in an optimized order, where the optimization criteria takes into consideration the learner's background and performance on related learning objects [4].

Adaptive Content Selection is the first step to adaptive navigation and adaptive presentation and is based on a set of teaching rules according to the cognitive style or learning preferences of the learners [5]. Adaptive Content Selection, Adaptive Navigation and Presentation are recognized as among the most interesting research questions in intelligent web-based education [6].

3 Adaptive Educational Hypermedia Authoring

The information structure of an adaptive hypermedia system can be considered as two interconnected networks or "spaces" [3]:

- a network of concepts (knowledge space) and
- a network of educational material (hyperspace or media space).

Accordingly, the design of an adaptive hypermedia system involves three key steps:

- structuring the knowledge
- structuring the media space
- connecting the knowledge space and the media space.

3.1 Authoring Abstraction Layers

The process of Adaptive Hypermedia Authoring can be represented by the use of five abstraction layers as shown in figure 1. In the literature several authoring frameworks have been proposed e.g. the LAOS [7], but those frameworks are focusing more on the concept and the adaptation logic layers. In our approach we propose an additional layer called *Pedagogical Strategy Layer* that focuses more on the pedagogical characteristics rather than the concepts covered by the educational resources. At this layer a pedagogical template based on the IMS Learning Design specification is introduced that is responsible for filtering the learning paths according to the selected pedagogical strategy. The proposed authoring abstraction layers are the following:

- *Learning Objectives Layer*. In this layer the author (or the learner if the educational hypermedia system includes an automatic courseware generator) can define a "learning goal". The learning goal is a node in a concept hierarchy graph that corresponds to the desirable by the learner knowledge.
- *Conceptual Layer*. In this layer related to the learning goal concepts are selected based on the structure of the knowledge space. The use of educational ontologies can significantly assist the structuring of the knowledge space.

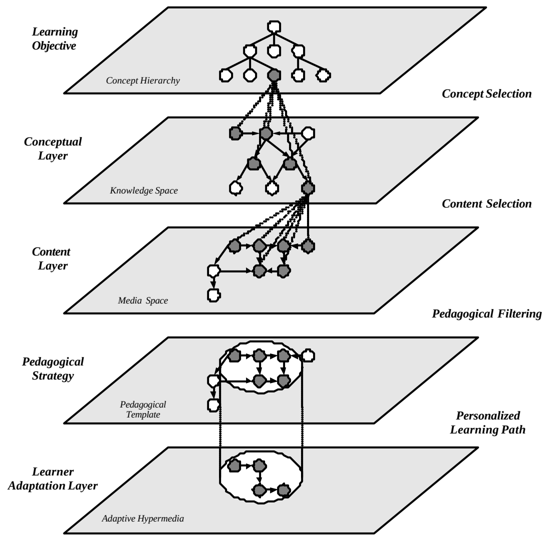

Figure 1. Abstraction Layers of Adaptive Hypermedia Authoring

- *Content Layer*. In this layer the learning resources that are related to the previously selected concepts are selected based on the connection of the knowledge space with the media space. The result of the selection is a directed acyclic graph (DAG) of learning objects inheriting relations from both spaces. This graph contains all possible learning paths in order for a learner to achieve the specified learning goal.
- *Pedagogical Strategy Layer*. This layer filters the media graph and produces a sub-graph which:
    1. is also a directed acyclic graph (DAG)
    2. includes all possible learning paths in order for a learner to achieve the specified learning goal, according to a specific pedagogical strategy.

The "pedagogical filtering" is based on the use of reusable templates allowing the definition of generic learning activities. These templates include the rules for adaptive navigation and can be processed by an adaptive content selection mechanism (previous abstraction layer) in order to serve adaptive web-based courses based on a diverse set of pedagogical strategies.

Figure 2 presents examples of such templates following the “diagnose-and-remedy” pedagogical strategy and an alternative “behavioral” approach.

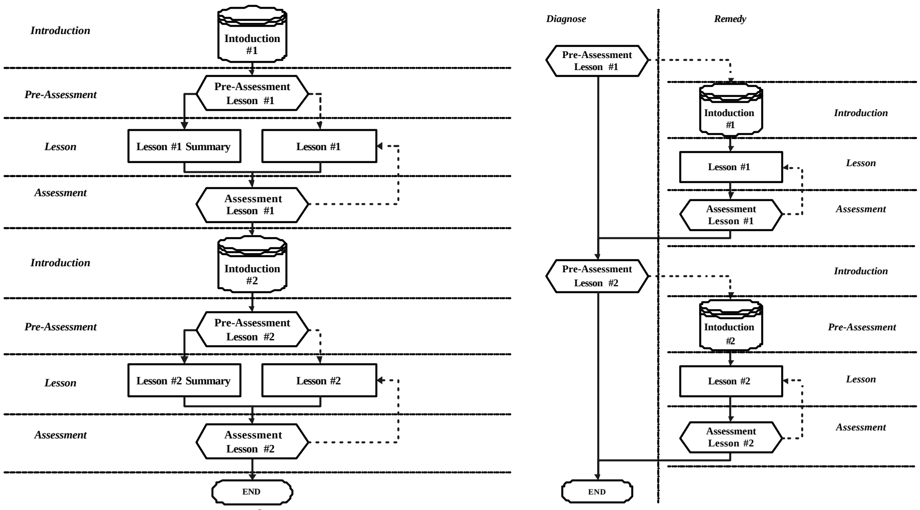

Figure 2. Example of Behavioral Template (a) and Diagnose-and-Remedy Template (b)

- *Learner Adaptation Layer*. This layer includes the process of adaptive learning path selection in order to produce a personalized learning path. The selection process takes into consideration educational characteristics of learning objects, learner cognitive characteristics as well as learner preference-related information stored in the learner profile. In our case, we use learning object characteristics derived from the IEEE Learning Object Metadata (LOM) standard [8] and learner characteristics derived from the IMS Learner Information Package (LIP) specification [9].

3.2 Discovering Optimum Learning Path

In order to extract from the resulting directed acyclic graph of learning resources the “optimum” learning path, we need to weight each connection of the DAG. The weighting process consists of two phases:

- *Selection of Criteria*. In Table 1 and 2 we have identified the LOM and LIP characteristics respectively, that can be used as criteria for the selection of the learning path.

Table 1. Learning Object characteristics for Learning Path Selection

Criteria	IEEE LOM	Explanation
General	General/Structure	Underlying organizational structure of a Learning Object
	General/Aggregation	The functional granularity (level of aggregation) of a

	Level	Learning Object.
Educational	Educational/Interactivity Type	Predominant mode of learning supported by a Learning Object
	Educational/ Interactivity Level	The degree to which a learner can influence the aspect or behavior of a Learning Object.
	Educational/Semantic Density	The degree of conciseness of a Learning Object, estimated in terms of its size, span or duration.
	LOM/Educational/Typical Age Range	Age of the typical intended user. This element refers to developmental age and not chronological age.
	LOM/Educational/Difficulty	How hard it is to work with or through a Learning Object for the typical intended target audience.
	LOM/Educational/Intended End User Role	Principal user(s) for which a Learning Object was designed, most dominant first.
	LOM/Educational/Context	The principal environment within which the learning and use of a LO is intended to take place.
	LOM/Educational/Typical Learning Time	Typical time it takes to work with or through a LO for the typical intended target audience.
	LOM/Educational/Learning Resource Type	Specific kind of Learning Object. The most dominant kind shall be first.

Table 2. Learner characteristics for Learning Path Selection

Criteria	IMS LIP	Explanation
Accessibility	Accessibility/Preference/typename	The type of cognitive preference
	Accessibility/Preference/prefcode	The coding assigned to the preference
	Accessibility/Eligibility/typename	The type of eligibility being defined
	Accessibility/Disability/typename	The type of disability being defined
Qualifications Certifications Licences	QCL/Level	The level/grade of the QCL
Activity	Activity/Evaluation/noofattempts	The number of attempts made on the evaluation.
	Activity/Evaluation/result/score	The scoring data itself.

- *Weight Calculation*. After identifying the set of characteristics/criteria that will be used, we define a weighting function that corresponds to the inverse suitability of a learning resource based on the profile of the target learner or group of learners. Let us consider a set of learning objects which is valued by a set of criteria $g = (g_1, g_2, \ldots, g_n)$. The assessment model of the inverse suitability of each learning object for a specific learner, leads to the aggregation of all criteria into a unique criterion that we call a weighting function and is defined as an additive function of the form: $W(g) = \sum_{i=1}^{n} w_i \times g_i \in [0,1]$ with the following additional notation:
  - g_i :the value of the ith selection criterion in the range [0,1] with 1 the less suitable value and 0 the most suitable value,
  - w_i : the inverse suitability weight factor of the ith selection criterion

Higher weighting value, means that a learning resource is less suitable, thus the link in the DAG that leads to that resource has less possibility to be included in the learning path.

After weighting the DAG with the use of the weighting function, we need to find the optimum (shortest) path by the use of a shortest path algorithm.

4 Conclusions

In this paper we address adaptive hypermedia authoring proposing an authoring framework that combines the approach of automatic courseware generation with the paradigm of educational hypermedia systems based on the use of ontologies and learning object metadata. The main advantage of this framework is the use of pedagogical templates which can be processed by an adaptive content selection mechanism in order to create adaptive web-based courses based on a diverse set of pedagogical strategies.

Acknowledgements

The work presented in this paper is partially supported by the European Community under the Information Society Technologies (IST) programme of the 6th FP for RTD - project ICLASS contract IST-507922 and the Greek Ministry of Development - General Secretariat for Research and Technology, project “eLand-An Integrated Virtual Environment for Supporting on-line Learning Communities” contract 2823/4-3-03.

References

1. Brusilovsky, P.: Adaptive and intelligent technologies for web-based education. Künstliche Intelligenz, Vol. 4 (1999) 19-25
2. Knolmayer, G.F.: Decision Support Models for composing and navigating through e learning objects, In Proc. of the IEEE International Conference on System Sciences, Next Generation Learning Platforms minitrack (2003)
3. Brusilovsky, P.: Developing adaptive educational hypermedia systems: From design models to authoring tools. In: Murray, T., Blessing, S., Ainsworth, S. (eds.): Authoring Tools for Advanced Technology Learning Environment. Dordrecht: Kluwer Academic Publishers (2003)
4. Brusilovsky, P., Eklund, J., and Schwarz, E.: Adaptive Navigation Support in Educational Hypermedia on the World Wide Web. In: Howard, S., Hammond J., Lindgaard, G. (eds.): Human-Computer Interaction. New York: Chapman & Hall (1997)
5. Brusilovsky, P., Vassileva, J.: Course sequencing techniques for large-scale Web-based education. International Journal of Continuing Engineering Education and Life-long Learning, Vol.13 (2003)
6. Devedžic, V.B.: Key Issues in Next-Generation Web-Based Education. IEEE Transactions on Systems, Man and Cybernetics Part C, vol. 33(3) (2003) 339-349
7. Cristea, A.I.: Evaluating Adaptive Hypermedia Authoring while Teaching Adaptive Systems. In Proc. of the ACM Symposium on Applied Computing SAC'04 (2004)
8. IEEE Draft Standard for Learning Object Metadata, IEEE P1484.12.1/d6.4 (2002)
9. IMS Global Learning Consortium Inc., Learner Information Package (LIP) Final Specification v1.0 (2001)

Authoring Web Content in ActiveMath: From Developer Tools and Further

Paul Libbrecht

DFKI and Universität des Saarlandes, Saarbrücken, Germany
paul@activemath.org

Abstract: In the ActiveMath learning environment, authoring is currently based on source editing following a *build*-practice similar to software development cycles. In this article we explain the characteristics of a build-cycle that are important to the creation of content for an adaptive educational hypermedia such as ActiveMath. The trust of an author into the behaviour of the system is one of the important goals that is aimed at. We then describe what, we expect, would further help the author's trust, namely, usable tools to input and play simulations of the learning process.

1 ActiveMath as an adaptive educational hypermedia

ActiveMath is an intelligent web-based learning environment. It presents content and lets learners perform interactive exercises.

Content, in ActiveMath is organized in knowledge units called *items* which roughly correspond to paragraphs in a text-book. The content of items is made of text and mathematical formulæ encoded semantically using OPENMATH [2]. Items are given a type (for example, a *definition* or *exercise*), and are complemented with metadata, the latter is made of slot-values (e.g., the *difficulty* or *learning-context*) as well as relations (e.g., the statement that a proof *proves* a given assertion, or for a concept to *depends-on* another concept). A subset of items, named *concepts*, along with their metadata make the *domain-model*. Please see [6] for a more detailed coverage of ActiveMath knowledge handling.

The adaptivity in ActiveMath is based on an overlay user-model based on this domain-model. The user-model is permanently updated using information such as reading of items and exercise results. Based on this user-model the following adaptivity features are provided in ActiveMath:

1. Using preferences, the presentation is given an appearance or *theme*
2. In ActiveMath the learner mostly accesses content through the familiar metaphor of books which are presented as a sequence of pages with a table-of-contents. The latter is presented with visual hints (little red, yellow, or green bullets) indicating which page refers to concepts with low, medium, or high user-model values. This can be seen as an elementary link-annotation adaptivity.
3. Books can be dynamically generated in ActiveMath: they are generated according to the goals of a learner and to a pedagogical scenario. Course

generation makes use of the items types, slot metadata, and relations as well as the user-model values so as to select the concepts that should be read as well as the supporting texts and exercises. This course generation can be viewed as a form of adaptive navigation support. While staying with the metaphor of a book, this course generation will be enhanced into a reactive component where the learner will be presented, in the usage of the book, with modifications to her book considered to be appropriate for her learning (see [7]).

4. Interactive exercises are currently mostly non-adaptive in ActiveMath: they are, for example, multiple-choice-questions. Exercises, within the LeActiveMath European project are planned to become adaptive using the same source of information as the course generation: it should make it possible for an interactive exercise to propose actions and provide feedback depending on the knowledge (and understanding or applications capabilities) of the learner for the given concept.

2 Authoring in ActiveMath: Tools Available

This section reviews briefly the authoring tools available in order to write content for ActiveMath.

2.1 Writing by Hand

Content in ActiveMath is encoded as XML documents in the OMDOC syntax[4]. As such, it can, in principle, be written by hand. This was done by some developers in our group while building the system. Experience has shown that doing so was a very confusing task especially with respect to OPENMATH formulæ which are encoded in very deep sub-trees.

An alternative was to write QMath documents, offering a compact syntax for both formulæ and content structure. The QMath processor then converts them to OMDOC. Results were disappointing as QMath was not complete enough for the needs of ActiveMath's OMDOCs. The treatment of mathematical formulæ in QMath was however kept: a wrapper was written, called OQMath, letting QMath process the formulæ. This allows the rest of the document be a valid XML document.

Experience has proven that the encoding for OQMath documents offers a reasonable readability while still allowing support of XML-based editing tools. An editor was chosen as reference tool to edit documents, the open-source editor jEdit. It offers, among others, templates, suggestion of possible children at insertion point and automated XML-validation (with a feedback similar to spell-checking).

The package is distributed under the name jEditOQMath and comes along with build-scripts to let authors reload content easily into a running ActiveMath, thus allowing them to cycle between content-preview and content-editing. The tool has been used successfully by non-developers, even without prior knowledge of HTML or XML.

2.2 On the Importance of Validation and Error-Reporting

As we see, the developer-tools approach is still quite present. But developer-tools are not only present by the fact that a source is being edited, but also by the fact that a *build-script* is being used along with error-reporting presented in the source. We describe here the validation tasks and error-reporting performed in the current jEditOQMath and highlight why they are important:

- XML-validation: in jEditOQMath, it happens simply at each save and makes sure the content elements are appropriately nested. The lack of such validation has been observed several times to create awkward presentation problems which are hard to detect.
- reference checking: reference between items are both inside the items and in the metadata relations. For a long time, tools to report these errors were not present and the database storing the content was very tolerant. Reference errors were frequent and had such misbehaviours as the lack of content in course-generation, the crash of the latter, or wrong user-model updating.

Actually, the needs for error-reporting are independent of the fact that a source is being edited. A visual tool should, as well, provide such feedback, including an error-list taking the user to the place where the error is to be corrected.

2.3 Verifying the Content

The need to verify (or *test*) an installed content can also be deduced from authors observed thus far: they spend almost more time to proof-read the content presentation than actually writing content. They do so as a *demo-user* (sometimes several) using their local web-browser.

One reason for this fact is certainly that the current presentation is from an XML source with extensible presentation of mathematical symbols for which it is easy to loose the overview. Moreover, this presentation provides access to many possible interactions (e.g. the navigation along a path of relations in the dictionary). It has been observed that such aspects of the presentations are, indeed, manually checked by authors.

All the adaptive behaviours of ActiveMath are also verified manually: the appearance under different *themes*, the results of course-generation, and the user-model updates. Such tests are, however, very long to perform: currently, course-generation takes several minutes and relies on a user-model state which the author should be clear with. Moreover, verifying the user-model update can only be done following a navigation path which takes a long sequence of clicks and may need much resources to be computed.

As a result, such verifications, although wished, are very rarely performed.

3 Scope of the Authoring Task

Before going further with the verifications task, allow us to try to answer the following question: *how much information should be affected by an authoring tool?*

The candid answer to this question would be that authoring tools should edit the *content*. Authors tend, however, to wish more than just providing content items, along with their metadata. Here are a few examples of possible modifications of the default ActiveMath behaviour that authors have wished thus far:

- the presentation of ActiveMath functions may be changed in order to make the system more accessible to target users (for example removing some links-generation for school pupils)
- adapting the pedagogical rules used in the course-generation scenarios to particular usages or particular content
- performing elementary changes in the appearance such as the introduction of the institution's logo or the change of some colour choices
- the base user-model that new registrants will be endowed with should be adapted to the expected newcomers (or their groups)

In order to be able to provide the freedom to perform and distribute such information, the scope of authoring should encompass the whole data of ActiveMath, including configuration, rules, and menu-templates. A notion of *project* should thus to be defined, extending the notion of content-collection. We shall try to satisfy these requirements: projects will be defined which will have *deployment* routines to install on a fresh ActiveMath. As much as possible, these projects will be encoded using ontology-based tools so as to allow a declarative knowledge representation of such modifications.

Offering a freedom as important as impacting the whole ActiveMath is, however, dangerous: it is easy to break a system by changing its configuration or changing a set of rules and a presentation can be made unusable because of wrong colour choices. The need to verify the *installation* of a project is thus made even more important. In the next section, we sketch how such verification could be helped by simulations which we propose to implement.

4 Simulations to Check the Learning Path

In [3], Hayashi, Ikeda, Seta, Kakusho, and Mizoguchi present an approach to author learning content using ontological engineering. They propose to model *conceptual simulations* which are high-level specifications of the expected behaviour of the content-playing-in-the-software. This approach seems to be the right path to take in the long term, where the systems' behaviour is sufficiently transparent, it seems not to provide an answer to nowadays verification wishes on existing adaptive systems and requires a very abstract representation for the simulations to be entered.

We claim that such a simulation can be made much more visual and concrete, in fact, close to the current practice of authors checking the content with his browser but providing capability to input quickly such a simulation (including the time spent to read items), store it and replay it, with a summarized view (e.g. thumbnail) and, most probably, with automated tests on such values as the learner-model entries or the existence of a link.Making these simulations of the size of a thumbnail allows an author to have several simulations under the eyes, thereby being to envision several target users. Being able to replay these simulations often with the content evolving allows the author to use these as a form of integrated tests: a quick view on the sequence of screens obtained in such a simulation provides a glimpse at the expected views the target user will experience, the measure of success of the tests is a measure of achievement of the content, a practice similar to the practice of unit-testing in software development.

Such simulations are probably of more general applicability. In particular, they apply for both content that is re-used and content that is freshly authored.

5 Conclusion

We have been describing the authoring tools realized thus far for the ActiveMath learning environment. The experience gained has proven the importance of the validation tools and verification possibilities so that authors' trust in the presentation and adaptive features can be gained.

In order to raise the trust we propose to provide to authors efficient and usable tools to input and play simulations of the learning process using visual approaches but still allowing computable verifications.

It should be observed that the visual simulations we are proposing could be interpreted as an application of the WYSIWYG paradigm (*What You See Is What You Get*). This is misleading, however, as the simulations are intended to provide multiple views on the content whereas a WYSIWYG approach is based on a single view.

5.1 Related Work

The problem of putting the control of intelligent *agents*, such as adaptive systems, at work under the hands of a user is not new, see for example [5]. However, we have found little done towards offering control of an adaptive systems to a user. These simulations would like to provide ways to experiment with an adaptive systems more efficiently and in a comprehensive fashion.

Our investigation in the literature for finding related work has not been fruitful. The high-level simulation formulated in [3] seems to be the only such approach and is fundamentally different from ours. We anticipate, however, that the needs for such a concrete simulations will grow as the serving world evolves in adaptivity functions.

5.2 Future Work

The development of the source-editing facilities that we have presented will be stabilized and the authoring tools in ActiveMath will migrate to edition within visual tools. Following the *view to the future* of [1], we are currently evaluating considering graph-based input of the domain-model along with the items' content using the Protégé ontology editor. The wealth of the verification tools will be kept and be complemented with visual simulations and their associated tests.

Acknowledgements

The author would like to thank the members of the ActiveMath group under the direction of Erica Melis for the feedback provided and the patience for their realization. He would like to thank the authors of ActiveMath content having taken the time to dive into the proposed tools even with little help or documentation.

This work has been partially supported by projects funded by the German Ministry for Education and Research (BMBF).

References

1 Peter Brusilovsky. Developing adaptive educational hypermedia systems: From design models to authoring tools. In Tom Murray, Stephen Blessing, and Sharon Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environment*. Kluwer Academic Publishers, Dordrecht, 2003. See http://www2.sis.pitt.edu/~peterb/papers/KluwerAuthBook.pdf.

2 O. Caprotti and A. M. Cohen. Draft of the open math standard. Open Math Consortium, http://www.nag.co.uk/projects/OpenMath/omstd/, 1998.

3 Yusuke Hayashi, Mitsuru Ikeda, Kazuhisa Seta, Osamu Kakusho, and Riichiro Mizoguchi. Is what you write what you get?: An operational model of training scenario. In Gilles Gauthier, Claude Frasson, and Kurt VanLehn, editors, *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems (ITS 2000)*, volume 1839 of *LNCS*, pages 192-201. Springer, June 19-23 2000. See also http://www.ei.sanken.osaka-u.ac.jp/pub/hayashi/hayashi-ITS2000.pdf.

4 Michael Kohlhase. OMDOC: Towards an internet standard for mathematical knowledge. In Eugenio Roanes Lozano, editor, *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2000*, LNAI. Springer Verlag, 2001. See also http://www.mathweb.org/omdoc.

5 Pattie Maes, Agents that reduce work and information overload. *Commun. ACM*, 37(7):30-40, 1994.

6 E. Melis, J. Büdenbender E. Andrès, A. Frischauf, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. Knowledge representation and management in ACTIVEMATH. *Annals of Mathematics and Artificial Intelligence, Special Issue on Management of Mathematical Knowledge*, 38(1-3):47 - 64, 2003. Volume is accessible from http://monet.nag.co.uk/mkm/amai/index.html.

7 C. Ullrich., An instructional component for dynamic course generation and delivery. In R. Tolksdorf and R. Eckstein, editors, *Proceedings of Berliner XML Tage 2003*, pages 467-473, 2003.

Educational Adaptive Hypermedia meets Computer Assisted Assessment*

Enrique Alfonseca, Rosa M. Carro, Manuel Freire, Álvaro Ortigosa, Diana Pérez, and Pilar Rodríguez

Computer Science Department, Universidad Autonoma de Madrid,
Carretera de Colmenar Viejo, km. 14,5,
28043 Madrid, Spain
{Enrique.Alfonseca,Rosa.Carro,Manuel.Freire,Alvaro.Ortigosa,
Diana.Perez,Pilar.Rodriguez}@ii.uam.es

Abstract. In this paper we explore the many possibilities that arise when we combine adaptive web-based courses with computer-assisted assessment. We argue that this integration has several advantages, such as the feasibility of getting a better model of the student's progress, which will be used with adaptation purposes, and the possibility of proposing and evaluating open-ended questions in the way that is judged more suitable for each student.

1 Introduction

Adaptive hypermedia has been widely used for the development of adaptive Web-based courses, in which each student is individually guided during the learning process [1]. Most of these systems obtain feedback from the student from two sources: their behaviour browsing the course (e.g. pages visited, time spent in each or navigational path) and test questions (e.g. true-false, multiple-choice or fill-in-the-blank questions). Some authors have expressed their concern that this limited way of assessment may not be really measuring the depth of the student learning [2]. This fact has been the motivation of the field known as Computer-Assisted Assessment (CAA) of student essays. CAA of student essays is a long-standing problem that has received the attention of the Natural Language Processing research community. There are many possible ways to approach this problem, including a study of the organization, sentence structure and content of the student essay [3, E-rater], pattern-matching techniques [4, IEMS], or Latent Semantic Analysis [5, IEA]. Valenti et al. [6] describe the state-of-art of CAA systems.

In order to support adaptive distance teaching and learning, we have developed the TANGOW system, which supports the description of adaptive web-based courses and their dynamic generation, so that their components are tailored to each student at runtime [7,8]. We have also developed, independently, a CAA

* This work has been sponsored by CICYT, project number TIC2001-0685-C02-01.

system called Atenea [9] which is based on n-gram co-occurrence metrics [10]. In this paper we describe the ongoing integration of TANGOW and Atenea and the benefits of this integration.

The paper is structured as follows: in Section 2 we describe separately Atenea and TANGOW, next in Section 3 we describe their integration and finally Section 4 will highlight the conclusions and future work.

2 Atenea and TANGOW

Atenea [9,11] is a Computer-Assisted Assessment system for automatically scoring students' short answers. It relies on the combination of shallow natural language processing modules [12] and statistically-based evaluation procedures. The recall is calculated by studying the percentage of those references that is covered by the student's answer and the precision of the student's answer is obtained by calculating the BLEU score [10].

The BLEU algorithm relies on the idea that a text is better when it is closer to a model text written by an expert. It was originally devised for evaluating Machine Translation systems, with the core idea that a Machine Translation is better when it is closer to the human translation. Hence, it looks for coincident n-grams (usually from unigrams to trigrams) between the human translator and the machine translations. In past work [9] we proved that BLEU can also be applied for evaluating students' answers, since the core idea remains: *the more similar a student's answer is to the teacher's reference answer, the better it is.* BLEU has been implemented, with minor modifications, as a module called ERB (Evaluating Responses with BLEU).

The system has already been tested with ten short questions, which are grouped in three collections: Operating Systems, Advanced Operating Systems and Object-Oriented Programming.

The scores provided by ERB are values between 0 and 1, where the upper and the lower bounds depend on the particular question. Nonetheless, we can perform a linear regression to find a correspondence between the interval of possible Atenea's scores and a fixed rank of scores, for instance, between 0 and 10. As expected, the quality of the assessment is very influenced by the kind of question and the references written. When scoring definitions, the correlation between Atenea's scores and the marks assigned manually can reach 0.80 [9]. On the other hand, questions that ask the student to make a reasoned argument or to compare several topics are more difficult to evaluate since they require a deeper linguistics processing.

The feedback that the students get from the system is a numerical score and a copy of their answer where, with color codes, they can observe which were the coincident n-grams and which words did not appear in any reference. From that output they can easily know which are the portions of their answers that are correct and have contributed in incrementing their score. Figure 1 shows an example answer page. In the user profile, students may also indicate whether they just want the score and are not interested in receiving this feedback.

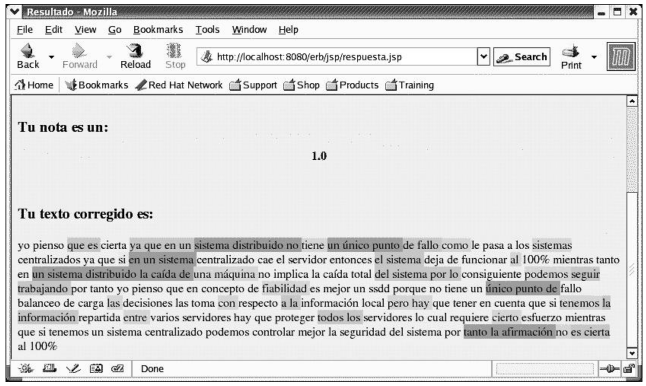

Fig. 1. Feedback that a student gets after answering the question *"Discuss whether distributed systems are more robust than monolithic systems"*, in Spanish. The darker the background, the longer the coincident n-gram.

Although Atenea is currently underpinned by ERB, it is not only limited to it. In fact, more NLP modules are currently being added to the already existing ones, including syntactic analyzers and word-sense disambiguation. A web-based wizard has also been developed to facilitate the task of introducing new data sets of questions and new questions.

The TANGOW system delivers adaptive web-based courses, and has evolved significantly since [7]. Courses delivered by TANGOW are composed of several tasks, that can be accomplished by the students. A task can correspond to either a theoretical explanation, an example, an exercise to be done individually or an activity to be performed collaboratively (problems to be solved, discussions, etc). The set of available tasks is constantly regenerated, tracking changes in the student's profile (static features and dynamic actions). Once a task is chosen, the system generates the corresponding web pages by selecting, among the content fragments and the set of available collaborative tools, those that provide the best possible fit to the current profile.

A rule-based formalism has been developed in order to facilitate the specification of alternative structures for the same course, and to support different teaching strategies, navigational guidance variations and collaboration workspaces for each type of student [8].

3 The integration of Atenea and TANGOW

The integration of Atenea and TANGOW will support the inclusion of CAA exercises inside adaptive courses, as a new type of TANGOW task. The process of integrating both systems was expected to be quite easy: Atenea would be launched from TANGOW and, after asking the students and automatically evaluating their answers, it would return the results to TANGOW so that this information could be used to update the user model and continue with the adaptation process. Atenea is currently configured to show different questions depending on the student's language and experience. The assessment is adapted to the used model, e.g. by showing easier questions to novice students. The information stored in user model also affect the set of reference answers that is chosen. The integration is not complete yet, but our ongoing work on implementation indicates that it will be simple to finish the links between both systems.

An initial step in the integration process was to decide which features from the current TANGOW user model would be used in Atenea in this first experience. We chose to use the *student name* as the login input in order to address the student by his or her name; *age*, because questions should be formulated in a simpler fashion for children than for adults, and different writing styles are expected from them; *language*, because we plan to extend Atenea with multilingual capabilities; *experience*, because the assessing process should be different for advanced students than for novice ones; and *feedback type*, because when formative assessment is used, the feedback should be more detailed than for summative assessment (where the score is the most relevant result).

Concerning the order of the questions, it is possible to take into account the student experience so that advanced students are not asked questions that they have already solved or that are too easy for their level. Moreover, the higher the level of experience, the stricter the system should be when assessing student answers.

The protocol for connecting TANGOW and Atenea is the following:

- TANGOW gathers information about the student's profile and sends it to Atenea, along with the identification of the task the student is going to perform, as well as the type of feedback desired.
- Atenea randomly chooses a question from the dataset corresponding to this task, that has not already been solved by the student (that is, not yet graded or graded less than half of the maximum score). The question is chosen taking into consideration the student profile. The answers submitted by the students are then evaluated by Atenea, and the resulting score and feedback is presented to the student. This process is repeated until the student has answered the required number of exercises. Finally, once the stop condition is satisfied, Atenea returns a holistic student score for the task to TANGOW.

A first consequence of the integration will be a richer set of activities, which can contribute to a more engaging learning process. Secondly, the use of the TANGOW formalism allows course authors to specify different teaching strategies by incorporating CAA activities at different points of the course, depending on each student's evolution. It will be possible for authors to choose the types of users to whom CAA activities will be presented; the places in the course where these exercises will appear; the requirements for a CAA activity to be proposed; and the grading criteria to determine the degree of success of each activity. Each of these adaptations can be made in different ways depending on the user's model. Finally, the formalism also supports the adaptation of CAA activities

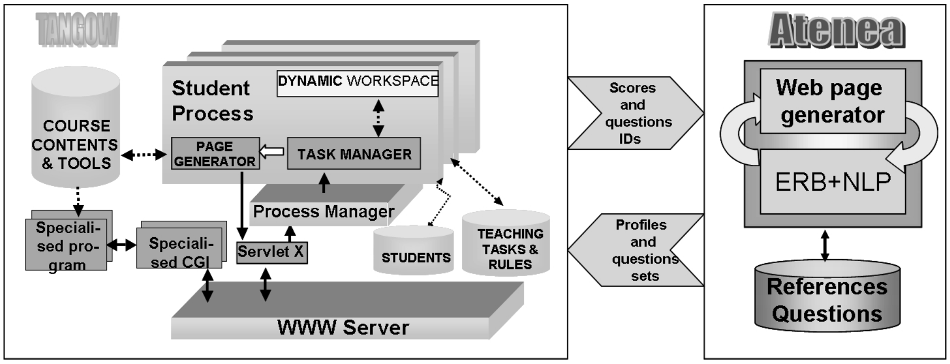

Fig. 2. Architecture of the integration between TANGOW and Atenea

themselves: the questions to be asked and the reference answers can be chosen according to each student's profile.

4 Conclusions and future work

We are implementing the integration of the adaptive hypermedia educational system TANGOW with Atenea, a program for automatic assessment of student answers. Atenea attains a good correlation with respect to teachers' marks, particularly when evaluating definitions and short descriptions [9]. The current implementation allows the students to try out their knowledge, and its complete integration with TANGOW, whose feasibility has been proved, will support the following:

- Atenea will use the description of the user profiles maintained by TANGOW, so it will accept variable profiles.
- The adaptation engine from TANGOW will decide at which time each student should be assessed, depending on his/her profile, knowledge, and actions, and Atenea will choose the most adequate set of questions for this student, resulting in a fairer evaluation.
- TANGOW will benefit not only from the possibility of automatically evaluating free-text answers, but also from the feedback from those questions, which can be used to guide the students during the rest of the course.
- It will be possible to obtain a dataset of student answers related to their profile and performance in the course, which we shall use in further studies to analyze how the adaptation can improve CAA activities.

The interaction protocol between Atenea and TANGOW has already been designed and is being currently implemented. Current work comprises a complete integration of Atenea and TANGOW in the direction described above and the evaluation of the integrated system with real students.

References

[1] Brusilovsky, P.: Adaptive hypermedia. User Modelling and User-Adaptive Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) **11** (2001) 87–110

[2] Whittington, D., Hunt, H.: Approaches to the computerized assessment of free text responses. In: M. Danson (Ed.), Proceedings of the Sixth International Computer Assisted Assessment Conference, Loughborough, UK (1999)

[3] Burstein, J., Leacock, C., Swartz, R.: Automated evaluation of essay and short answers. In: M. Danson (Ed.), Proceedings of the Sixth International Computer Assisted Assessment Conference, Loughborough, UK (2001)

[4] Ming, Y., Mikhailov, A., Kuan, T.L.: Intelligent essay marking system. In: C. Cheers (ed.), Learners Together, Singapore, NgeeANN Polytechnic (2000)

[5] Laham, D.: Automated content assessment of text using Latent Semantic Analysis to simulate human cognition. Ph.D. Dissertation, University of Colorado, Boulder (2000)

[6] Valenti, S., Neri, F., Cucchiarelli, A.: An overview of current research on automated essay grading. Journal of Information Technology Education **2** (2003) 319–330

[7] Carro, R., Pulido, E., Rodríguez, P.: Dynamic generation of adaptive internet-based courses. Journal of Network and Computer Applications (1999) 249–257

[8] Carro, R., Ortigosa, A., Schlichter, J.: A rule-based formalism for describing collaborative adaptive courses. In: Knowledge-Based Intelligent Information and Engineering Systems. LNAI 2774, Springer-Verlag (2003) 252–259

[9] Pérez, D., Alfonseca, E., Rodríguez, P.: Application of the Bleu method for evaluating free-text answers in an e-learning environment. In: Proceedings of the Language Resources and Evaluation Conference (LREC-2004). (2004)

[10] Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation (2001)

[11] Pérez, D., Alfonseca, E., Rodríguez, P.: Upper bounds of the bleu algorithm applied to assessing student essays. In: Proceedings of the 30th International Association for Educational Assessment (IAEA) Conference. (2004)

[12] Alfonseca, E., Perez, D.: Automatic assessment of short questions with a bleu-inspired algorithm and a shallow semantic representation. Accepted in the 4th International Conference Spain for Natural Language Processing (ESTAL) (2004)