Adaptive Authoring of Adaptive Educational Hypermedia

Alexandra Cristea and Lora Aroyo

Mathematics and Computing Science, Technical University Eindhoven, The Netherlands
{a.i.cristea, l.m.aroyo}@tue.nl
http://wwwis.win.tue.nl/~{alex, laroyo}

Abstract. In this paper we propose a set of development guidelines for an *adaptive authoring environment of adaptive educational hypermedia*. This set consists of relevant and necessary functionalities and architectural features of authoring systems (AS) for adaptive teaching and/or learning environments (LE). We extracted the core functionalities by analysing MyEnglishTeacher (MyET) and AIMS - two independently designed and built AS for adaptable/ adaptive LE. The extended core led us to a *concept-based layered approach* with respect to *concept*- and *attribute-level adaptation*, as well as *lesson* and *presentation adaptation*. We believe that providing adaptive authoring support for adaptive hypermedia will have a strong impact on the authors' motivation and efficiency in performing their tasks and consequently will increase the popularity of adaptive hypermedia.

1 Introduction

To keep up with the high demands in educational software and LE, it is already widely accepted by the educational community that the future of such environments lies in *adaptability* and *adaptivity* [11,13,18,19]. These high demands reflect also on the authoring environments, general purpose or application oriented. At present, AS for adaptive LE are almost non-existent, mainly because of the field novelty and the growing demands– making any authoring system for LE outdated before it is used. However, with the ripening of the field, *standardization of adaptive and adaptable techniques and methods* [10,15,16,22] is starting to preoccupy the research community, and AS become a serious need. Still, it is not reasonable to expect teachers to skilfully apply adaptation to their new on-line courses. Transition from linear, book-like courseware is not easy. Teachers are confronted with the heavy task of designing contents alternatives, adaptation techniques and ultimately, the whole user-interaction mechanism. It is clear that for adaptive courseware the authoring tool has to offer tuneable complexity and automatically perform many of the authoring tasks.

In this paper, we build upon our previous research by shortly reviewing two AS for adaptable/ adaptive LE, MyET and AIMS, which we designed and built independently. From our experience with these two systems we are extracting a set of functionalities that are relevant and necessary for AS for adaptive teaching and/or LE. Given the complexity of the authoring task such environments [14], we independently came to the conclusion that the authoring tool has to be also adaptive to the teacher. Therefore, we extend our previous research to an *adaptive authoring framework* based on the above-mentioned necessary functionalities.

2 Two methods for authoring support of adaptive courseware

In this section we are presenting the two independently designed, developed and tested systems for courseware authoring, with special focus on adaptive courseware: (a) MyET [9], developed at the University of Electro-Communications in Japan, and (b) AIMS [2], developed at Twente University in The Netherlands. In both MyET and AIMS concept mapping paradigm [4,7] is used as a main structure to organise the subject domain terminology and to link to course items (lessons, exercises, documents). We outline the common features shared by these systems with respect to course content organization, maintenance and presentation, with the purpose of *reusability* and *student adaptation*. Our evaluation framework is based on the: (a) *general course structure*; (b) *text presentation and structure*; (c) *lesson composition*; (d) *authoring views and (e) student adaptation facilitation.* Due to lack of space, we have skipped the analysis of tests and exercises creation, structuring and linking.

2.1 General course structure

The information exchange from tutor to system means input of lessons, texts, links, but also asking for help in editing, etc. The data from the tutor is stored in a structured way, as shown in Figure 1. Input can be *audio, video, graphic* or *text.*

Teacher-name
AUDIO FILMS PICTURES LESSON
113-1 115-1 119-1 123-1 190-1 196-1 65-1
114-1 118-1 119-2 123-2 190-2 197-1 67-1
EXERCISES conclusion keywords
title TEXTS explanation pattern
TEXT1 TEXT2
exercise1 keywords pattern

(a)

COURSE: <name> <description>
INSTRUCTOR: <name>
DOMAIN: <name>
TOPIC-1: <name> <description>
MAIN-TEXT: <link>
TASK-1-1: <name> <keywords> → TEXT(s)
TASK-1-N: <name> <keywords> → TEXT(s)

(b)

Fig. 1. Data structure in MyET (a) ; Course structure in AIMS Course Editor (b)

a. In MyET, the first three inputs have also a text version in the lesson and with automatically generated index. Each lesson object is subdivided into *exercises, conclusion, keywords, title, explanation, pattern to learn* and *text objects*. These are further subdivided into *exercise, keywords, pattern, title* and actual *main text*.

b. A related structure appears in Fig. 1b. The teacher in AIMS creates the course structure as part of an information model, defining the *subject domain* (course terminology/concepts), *library* (collection of texts), *course* (lessons and exercises) and *user profile* objects and expressing all the links between them and their components [3]. Each course *lesson* is divided into *exercises*, which are directly related to subject domain concepts and this way also to related documents (texts). This domain-specific structuring and concept-based linking are provided [17].

2.2 Text presentation and structure

The smallest block / object in the course structure is a TEXT.

In MyET, each text also has (next to text body), some obligatory attributes: a *short title, keywords, explanation, patterns to learn, conclusion* and *exercises*. This way, not only titles and keywords but also explanation and conclusion files are used for search and retrieval. Moreover, the corresponding text to any video/audio recording allows any non-text resource to be retrieved via a text-based search.

The authors in AIMS are offered the choice to select *keywords from the domain terminology* they have created as a concept map. These keywords also make the link directly to the lessons (course structure of course topics and exercises). The AIMS library editor provides the authors with presentation and instructional formats to map each course related material in such a way that it is both *task*- and *use-oriented* [2].

2.3 Lesson composition

One or more texts (with multimedia or not) build a LESSON object. In MyET, each lesson also has (beside of texts, etc.) the following attributes: *title, keywords, explanation, conclusion, combined exercises* (generated automatically or not). This structure is very similar to the text object structure.

A text or lesson is generically called ‘SUBJECT’ in MyET or ‘TOPIC’ consisting of course ‘TASKS’ in AIMS.

2.4 Student Adaptation issues

In the MyET environment, adaptation to the students’ needs meant interpreting the concept maps and the links created by the teacher. A *global agent* would copy the map to create a global student model that would serve as a guide for all students. Moreover, a *private agent* would make its own copy and alter it with respect to the student’s needs and based on the interaction with the student. The system agents work based on the embedded rule/knowledge systems. They act as learning objects, which can adaptively change their representation of the subject space [10].

The adaptation to the students’ needs in AIMS is performed with a team of collaborating agents with respect to (a) *user-oriented presentation of the information*, based on the general work progress of the student on the course, and (b) *students’ search activities.* This is realised by refining the student search query according to the current course task context, by adjusting the search result presentation with relevance to the course task and by providing students with alternative view of the results and thus alternative ways to build a conceptual structure of the course knowledge [2].

2.5 Courseware Views for Authors

Once a map has about 50 subjects representing a number of concepts with their links, it is impossible to get a sensible display in one map [1]. To manage the complexity we designed extra courseware views showing different sub-sections: “views” of whole graph, with reduced information ("*bird's eye views*"); one concept and its "star"-links (all concepts currently linked to it) ("f*isheye views*" [9,11]); non-linked concepts: "*floating*"-concepts [10]; one concept and its *"star"-linked documents* (texts) [3]; all *link types* currently in use [2]; all *concepts ordered alphabetically* by name, by description and category; the *place in the concept map* of every newly defined concept.

3 Our concept mapping layered approach to adaptive authoring

From the analysis and extraction of common proprieties of the two AS, a more intuitive division seems therefore to be to separate the course material into concepts [2,10], as derived from the *concept-mapping* paradigm [7]. A low level concept should represent an *atomic piece of content/ information* that has an independent semantics. This atomic unit can be labelled with one concept. Collections of concepts can, of course, build *composite concepts*, generating a *concept hierarchy*. Concepts can be related to each other at any level of the hierarchy. The creation of these building bricks is the role of the course designer [3,9]. The division of the content into concepts only gives us the primitive building blocks of the courseware. Putting these building blocks together with different sequences generates different presentations or *lessons*. This can be done by a course designer or by a teacher. In a more advanced environment it can be automatically generated by a system [10].

At this level, we would only speak of *adaptive navigation support*. That is because *adaptive presentation* is (normally) at a lower level than the concept level and binds actually parts (fractions) of concepts with each other. Clearly, it makes no sense to just transform parts of concepts into sub-concepts, as it is possible that they make no sense but in the context of the atomic concept (so have no independent semantics attached).

- *Example:* Consider, for instance, the case of the introduction to some text. This is a construct that appears very often and that can be dropped in later versions of the browsing (or used together with other introductory fragments in an introductory chapter). However, this construct usually has no independent meaning.

The solution is quite obvious [9]: the concept can be sub-divided into its *attributes*. These can be anything from a concept name to alternative contents or fragments.

By dividing/mapping the course content into a concept hierarchy, and the concepts into a set of attributes, the adaptation has only to deal with *concept-level adaptation* and *attribute adaptation*. The advantage is that it can all be performed (and viewed) from a high level and does not need separate consideration of different conditions written within the text, which are more difficult to re-use by other authors. Basically, the adaptation becomes only a matter of combining concept attributes into *pages* (pieces of information that can be show at a time), *chapters* and *subchapters*. This way, the adaptation is only at a content level (equivalent to adaptive presentation), while the navigation is only dependent on the presentation format. We will return to these issues in Section 3.1 (e.g., short pages will mean that the "next" button within the same lesson appears more often, but the content of the page is shorter [20]).

3.1 Lesson map

A lesson map is, in the simplest case, the lesson sequence that tells the student how the lesson should proceed (according to the teaching style, learning style or learning goals). In a more general case (Figure 2), the lesson map is a directed (not necessarily a-cyclic) graph with at least one beginning (START) and at least one ending state (GOAL). Circles here represent either whole concepts, or concept attributes.

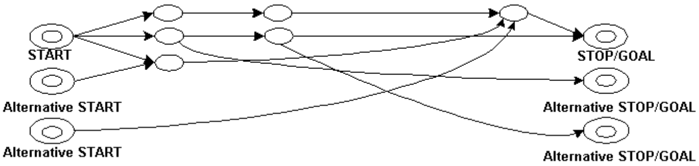

Fig. 2. Alternatives ‘start’ and ‘stop’ positions for the lesson map

The START and STOP positions do not have to be unique. The user model (UM) based adaptation engine determines the selection between alternatives.

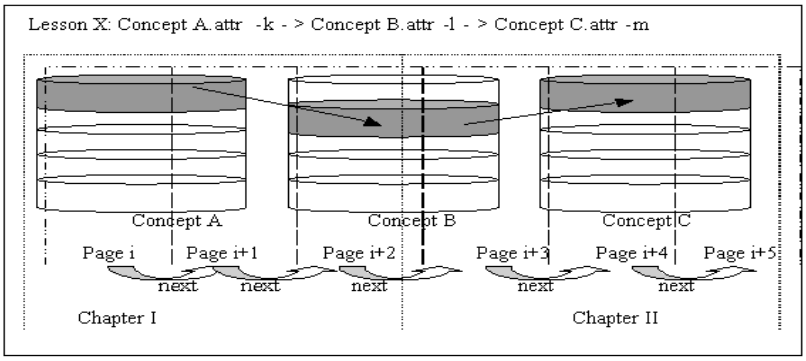

Fig. 3. Lesson adaptation: *lesson* X (straight line square), composed of, e.g., 3 *concepts* A, B and C (cylinders), with *attributes* (darkened cylinder); presentation order shown by directed connections between attributes; lesson has 2 *chapters*, that contain (parts of) concepts (dotted line squares). In chapters information is presented (in browser window, e.g.) in *pages* (point-dotted line squares). “Next” buttons at page level are navigation support for presentation only.

In Figure 3 we present a sketch of the concept mapping layered approach to lesson adaptation. A lesson X is composed of some concept fragments (attributes) that are grouped into chapters (here, 2). At presentation, there is another separation into pages (here, 6 pages). The latter is presentation means dependent (e.g., laptop screen browser, hand-held device, etc.).

It is interesting to note that, although the fragments themselves have no semantics (and therefore could not have been subdivided into concepts), they can be addressed via the concepts they belong to.

3.2 Authoring adaptive courseware with concepts

Next we present two types of concept editing processes. This way we show that the transition from classical course editing - to pure concept editing style is an easy one.

3.2.1 Concept –based authoring in traditional order

In order to make the transition from traditional editing to concept editing easier a concept-based authoring tool should be able to allow, in extreme cases, pure traditional editing. In this section we present a five-step procedure for concept editing in traditional order, illustrated in Figure 4.

1. First, the author writes sequentially the text of the course/lesson together with the respective multimedia. The instructor can stop the process here or at any of the next steps and let the system automatically perform the rest of the steps and show the results again to the teacher for approval, or to another author for reusing.
2. The content is divided and organised into a *concept structure*. First, the concept hierarchy of concepts, sub-concepts and atomic concepts is created, where the atomic concepts are the smallest semantically indecomposable building blocks within this structure. Further, the main attributes of the concepts are filled in, such as concept name and content and possibly any other related attributes [9].
3. Adaptive features are added by writing a separate rule-base [22] or setting importance coefficients and weights [10] and generating at least one *lesson map* (sequence). In some cases it may be necessary here to return to STEP 2 in order to refine the division granulation or to STEP 1 to add more concepts/ material.
4. The author can define the different items to display depending on certain conditions (only mentioned in the separate rule-base, started at STEP 3: lesson maps).
   - ➢ At first glance, we have here *alternatives* (such as alternative texts in different languages for the same concept/ lesson/ etc.) or *conditionals* (such as additional information that is only presented if “conditions are right” – classical adaptive presentation). However, from the adaptive engine point of view, this distinction cannot be meaningful, as conditionals can be seen as alternatives, where, if the condition is not satisfied, nothing is displayed (instead of an alternative). Here we don’t discuss the cases where empty information pages result due to, e.g., bad authoring. This is a matter of presentation means adaptation and not user-modelling adaptation.

   After new attributes have been generated the author can return to STEP 3 to refine the rules. If necessary, s/he can jump to STEP 2 to refine granulation of division or to STEP 1 to add more concepts/ material.
5. The author creates the lesson maps following the procedure from Section 3.1.

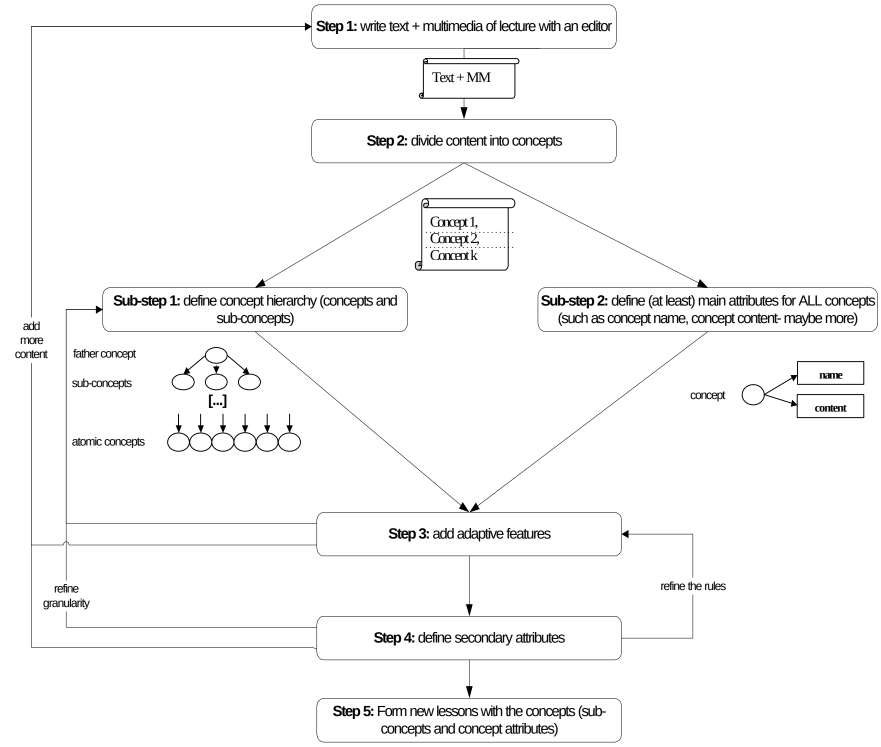
Step 1: write text + multimedia of lecture with an editor
Text + MM
Step 2: divide content into concepts
Concept 1,
Concept 2,
Concept k
Sub-step 1: define concept hierarchy (concepts and sub-concepts)
Sub-step 2: define (at least) main attributes for ALL concepts (such as concept name, concept content- maybe more)
add more content
father concept
sub-concepts
[...]
atomic concepts
concept
name
content
Step 3: add adaptive features
refine the rules
refine granularity
Step 4: define secondary attributes
Step 5: Form new lessons with the concepts (sub-concepts and concept attributes)

Fig. 4. Concept-based editing: traditional order

The procedure above allows the author to be as precise and detailed as s/he wants but at the same time, it allows him/her to do as little authoring as possible. This is realized by the clear division between the authoring stages. An author/ teacher can be just content creator – or the creator of non-adaptive hypermedia (STEP 1). Other authors nevertheless can reuse and refine the created content, by performing the steps starting from STEP 2. In this way, we have accomplished several goals:

- Simple authoring, yet
- Complex results; moreover
- Collaborative authoring.

The latter is made possible by the semantic structure, based on a flexible, growing common ontology based on well-annotated concepts (as concept attributes can also take over the role of concept annotations).

Moreover, the transition from STEP 1 to STEP 2 is not unique and loops to refine the granulation of the concepts are possible – so, subdivisions of higher level into lower level concepts – as long as the units obtained still have an independent semantics. This structuring is in the sense of the Semantic Web, while the division of contents into concepts, with their attributes and links is similar to the RDF W3C standard recommendations (of separating resources, literals and properties).

3.2.2 Non-traditional authoring: direct concept editing

The previously presented method is only to make the bridge between traditional (classical) editing and concept-based editing. In a purely concept oriented editor, an author would follow the following steps:

STEP 1: write concepts + concept hierarchy
STEP 2: define concept attributes (define main and extra attributes)
STEP 3: fill concept attributes (write contents)
STEP 4: add content related adaptive features regarding UM (des. alternatives, conditions)
STEP 5: define format (presentation means-related; define chapters)
STEP 6: add adaptive features regarding presentation means (define pages).

3.3 Resulting layers, author adaptation and automatic processing per layer

Concluding from this procedural explanation, it is clear that the courseware resulting will have a *layered structure* as illustrated in Figure 5.

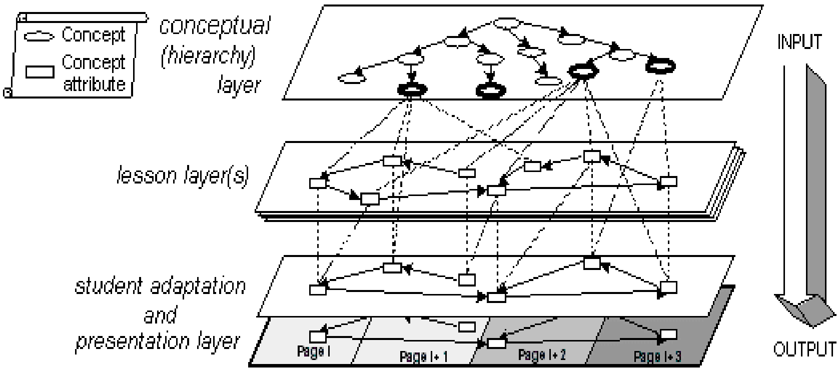

Fig.5. Authoring adaptation layers

1) The first layer is the conceptual layer, with
   a) a first sub-layer of *atomic concepts*, which cannot be changed anymore.
   Adaptation and automatization at this level can mean search of related (possibly identical) concepts, in order to warn the author that the new concept s/he wants to define possibly already exists. Also, this is the moment to automatically establish connections between concepts (computed by system via heuristics [9] and approved by the author).
   b) a second sub-layer of composite concepts, which are sets of concepts of atomic granulation or larger, and which have a *hierarchical structure*. Course designers and/or system can change this hierarchy automatically. In this layer, connections as *relatedness connections* [10] also appear.
   The system can perform concept – sub-concept adaptation by helping in suggesting relevant higher concept classes, etc.
2) The second layer contains *lessons* (the hierarchy of chapters, sub-chapters and the directed graph called lesson – Section 3.1). This represents the way and the order in which the concepts should be taught. Note, that the lesson layer deals with con-

cept attributes and not necessarily directly with concepts. This allows having both adaptive navigation support and adaptive presentation within the same layer and guarded by the same mechanism. The system can search for alternative existing orders and make the author attentive to other possibilities or inquire in a dialog with the author for the possibility that the order is not compulsory.

3) The third layer contains

   a) a UM-based sub-layer, an *adaptation engine* that specifies what material should be presented and when (under which conditions or equivalent sets of conditions – Section 3.4).

   From this point, user adaptation means student adaptation.

   b) a sub-layer, formed by the *presentation means based adaptation*. This adaptation part is concerned with the formatting so that the information appears nicely in the page, with questions such as the ideal page length – so where chapters should be cut to form pages, how and where multimedia presentations should appear, colours, fonts, etc. These matters should, in the simplest authoring version, be generated fully automatically in order to simplify the task of authors. Of course, such issues should also be designable by authors, if necessary [20] Here, rhetorical structure adaptation should be overlayed.

5 Conclusions

Providing adaptive authoring for adaptive hypermedia is quite a crucial task as the authoring process involves, beside of what was mentioned in this paper, also a number of other complicated tasks, such link-checking (e.g., issues of *termination* and *confluence* [22]) which become almost impossible for a human to keep track of. Therefore, next to the adaptation support with respect to the content organization and presentation, an important issue is also the provision of support *tools to analyse and monitor the information input by the author*. The editing environment must provide support for a number of editorial tasks, such as information search and retrieval, information visualisation, selecting, restructuring, annotating information with metadata, generation of adaptive user feedback and user preferences information. In order to make the course related content and knowledge to be most efficiently maintainable it is of a vital need for the authors to be provided with facilities to view the content from different perspectives and to perform various analyses and statistics on it.

Note that the few authoring systems which allow auhoring of adaptive hypermedia are quite restrictive. The well-known InterBook shares the concept-based approach, but relies only on the simple overlay model, and is based on a strict prerequisite structure [6]. A more advanced system, Tangow [8], has a top-down approach (whereas here we describe a bottom-up one), and requires a predefined set of concept (here, task) attributes – whereas here we allow more flexibility.

We obtained from the analyses and tests of two separate, independent systems (MyET and AIMS) a framework of a concept-based, layered architecture and guidelines for adaptive hypermedia, in concordance with the stratified hypermedia structure for information disclosure [5]. This way we set the basis towards standardization-based authoring [15, 16, 21].

References

1. Ackerman, F., Eden, C., Cropper, S.: Cognitive Mapping – a user guide, http://www.banxia.com/depapsum.html (1993)
2. Aroyo, L., Dicheva, D.A.: Concept-based approach to support learning in a Web-based support environment, In: Moore, J. (eds.) Proc. of AIED’01, Leipzig: IOS Press (2001) 1-12
3. Aroyo, L., Dicheva, D., Velev, I.: Conceptual visualisation in a task-based information support system, In: Bourdeau, J., Heller, R. (eds.): Proc. of EdMedia’00 Conf. (2000) 125-130
4. Beyerbach, B.: Developing a technical vocabulary on teacher planning: preserves teachers’ concept maps, Teaching and Teacher Education, Vol. 4 (4) (1988) 339-347
5. Bruza, P. D., Van der Weide, T. P.: Stratified Hypermedia Structures for Information Disclosure. The Computer Journal, Vol. 35 (3) (1992) 208-220
6. Brusilovsky, P., Schwarz, E., Weber, G.: A Tool for Developing Adaptive Electronic Textbooks on WWW, WebNet’96
7. Buzan, T., Buzan B.: The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential. New York: Plume (1996)
8. Carro, R. M., Pulido, E., Rodriguez, P.: Designing Adaptive Web-based Courses with TANGOW, ICCE’99, v. 2, 697-704
9. Cristea, A., Okamoto, T.: MyEnglishTeacher–A WWW System for Academic English Teaching. ICCE’00, Taiwan (2000)
10. Cristea, A. I., Okamoto, T.: Object-oriented Collaborative Course Authoring Environment supported by Concept Mapping in MyEnglishTeacher, Edu. Tech. & Society 4(2) (2001)
11. De Bra, P., Brusilovsky, P., Houben, G.-J.: Adaptive Hypermedia: From Systems to Framework. ACM Computing Surveys, http://wwwis.win.tue.nl/~debra/public.html (1999)
12. Dicheva, D., Aroyo, L.: An approach to intelligent information handling in Web-based learning environments. In: Arabnia, H. R. (ed.), Proc. of ICAI’00 Conf. (2000) 327-333
13. Fink, J., Kobsa, A., Schreck, J.: Personalized Hypermedia Information Provision through Adaptive and Adaptable System Features. Mullery, A. et al (eds.)Springer (1997) 456-467
14. Höök, K., Karlgren, J., Waern, A., Dahlbäck, N., Jansson, C-G., Karlgren, K., Lemaire, B.: A Glass Box Approach to Adaptive Hypermedia, Journal of UMUAI, 6 (1996) 157-184
15. IEEE P1484.6 Course Sequencing Working Group: http://ltsc.ieee.org/wg6/index.html
16. IEEE P1484.2 Learner Model Working Group: http://ltsc.ieee.org/wg2/index.html]
17. Murray, T. et al.: MetaLinks - A Framework and Authoring Tool for Adaptive Hypermedia. S. Lajoie and M. Vivet (eds.): AIED‘99, (1999) 744-746
18. Okamoto, T., Cristea, A., Kayama, M.: Future integrated LE with Multimedia. JCAI, Advanced information technologies for learning in the APR. Vol. 17 (1) (2001) 4-12
19. Opperman, R., Rashev, R., Kinshuk: Adaptability and Adaptivity in Learning Systems, Knowledge Transfer (Volume II) (Ed. A. Behrooz), 1997, pAce, London, UK, pp173-179
20. SMIL, W3C standard, http://www.w3.org/AudioVideo/
21. W3C Recommendation: Authoring Tool Accessibility Guidelines 1.0 (3 February 2000): http://www.w3.org/TR/ATAG10/
22. Wu, H., De Bra, P.: Sufficient Conditions for Well-behaved Adaptive Hypermedia Systems. Proc. of WI Conf. Lecture Notes in AI. Vol. 2198. Springer-Verlag. (2001) 148-152