TOWARDS AUTOMATIC CONSTRUCTION OF ADAPTABLE COURSEWARE STORYBOARDS

Boyan Bontchev and Dessislava Vassileva
Department of Software Engineering, Sofia University, 5, J. Baurchier blv., Sofia 1164, Bulgaria

Keywords: Automatic Course Construction, Adaptive e-learning, Adaptive Hypermedia Systems, Instructor Tool, Learning Styles.

Abstract: In last twenty years, researchers have conducted intensive research in the area of principal models, software architectures and practical system development of adaptive e-learning platforms. Brains are fascinated by great opportunities for radical improvement of the teaching process by means of applying adaptability at different levels. There are two general issues of adaptive e-learning – enabling different educational content delivered to different individuals or groups and, as well, differently formed sequencing and presentation of that content delivery. This paper presents two approaches for creating and delivering training courses adaptable to learners with different learning styles. The first one is implemented within a platform for building edutainment (education plus entertainment) services called ADOPTA (ADaptive technOlogy-enhanced Platform for eduTAinment). By means of ADOPTA, e-learning courses can be created manually by an instructor as directed storyboard graphs. Another feasible approach is to generate them automatically on-the-fly by the adaptive engine. The article discusses advantages and drawbacks of these two approaches for adaptive e-learning course construction.

1 INTRODUCTION

The hypermedia paradigm is based on usage of hypertext for organising presentation of structured content for Internet access. Naturally, such a paradigm allows introducing models and techniques for adaptive content delivery. Adaptive Hypermedia Systems (AHS) make use of them and represent mainly software applications and platforms for adaptive e-learning, intelligent tutoring, adaptable multimedia delivery and adaptive web games. From the very beginning, AHS try to adapt content in various ways according the user profile. Bearing that to the e-learning area, AHS deliver hypertext and hypermedia content that is consistent with the profile of individual learner or group of learners (Dagger et al, 2005). Such a content delivery requires definition of various pedagogical strategies for a course, mostly supported by appropriate tools for instructional design. Each strategy is supposed be best suited for a particular learner according her/his learning style, knowledge, preferences and goals (Bontchev, Vassileva, 2006). Some e-learning platforms with instructional design tools are as follows:

- InterBook (Brusilovsky et al., 1996) – it provides means for the creation and presentation of adaptive electronic textbooks. Disadvantages of InterBook are that it does not support advanced adaptive methods and there is insufficient suitable interfaces.
- NetCoach (Weber et al., 2001) - knowledge of each training course is presented as a network of concepts. However it does not support learning styles.
- AHA! (De Bra et al., 2006) - learning content is stored in pages, which are represented as XML files. The presentation of the content page is determined at runtime according to predefined conditions. This can lead to confusion and ambiguity among course authors.
- ReCourse (Griffiths et al., 2009) – it is not an e-learning system but rather a tool for creating learning content in accordance with the IMS learning design standard. ReCourse provides rich and user friendly interface, but it supports only IMS LD.

In this paper, we present an instructor tool, which covers disadvantages of above examined tools and is

Bontchev B. and Vassileva D. (2010).
TOWARDS AUTOMATIC CONSTRUCTION OF ADAPTABLE COURSEWARE STORYBOARDS.
In *Proceedings of the 2nd International Conference on Computer Supported Education*, pages 368-373

integrated within ADOPTA (Bontchev and Vassileva, 2009) – an ADaptive technOlogy-enhanced Platform for edutainment, i.e. education plus entertainment. This instructor tool provides rich, comfortable and effective interface for creating courses including various pedagogical strategies. Moreover our module supports learning styles of all kinds and is not bound to a specific standard. It is consistent with our principal adaptability model of adaptive AHS (Vassileva and Bontchev, 2009) as described in the next chapter.

Despite the facilities introduced in the instructor tool, the process of creating adaptive course takes much times and efforts of an instructor. Furthermore, not always existing courses can cover goals of all students. Sometimes a learner with specific objectives need to pass several courses, part of the content of which is already known about her/his. In these cases it is convenient to use automatic generation of an adaptive course. In this area there are various successful development such as PASER (Vrakas et al., 2007), DCG (Vassileva, 1997) and OntAWare (Claus and Holohan, 2009). All of them are based on domain ontologies and construct educational content using links between them and their learning objects. For better efficiency, very important to them are metadata of learning objects that give more information when a particular LO is the most suitable to be used.

The first manual approach for course creating is implemented as a part of the ADOPTA platform for building edutainment (education plus entertainment) services (Vassileva et al., 2009). The second one is in the process of discussion and planning as future functionality. ADOPTA is a modular system and includes: authoring tool for establishing the e-learning course content, instructor application, and software engine, which is responsible for adaptable content delivery to every individual learner.

2 CONCEPTUAL MODEL OF AHS

The ADOPTA platform is based on a newly proposed hierarchical principle model which tries to improve the traditional AHAM reference model (De Bra P. at al, 1999). Table 1 describes the essence of this model of AHS and provides explanation of its most important characteristics (Bontchev, Vassileva, 2006).

Table 1: Tabular presentation of the structure of the conceptual model.

Learner Model - provides description of the learner character as a triple of sub-models, namely *Goals and Preferences, Learning Style* and *Knowledge and Performance*.	***Goals and Preferences***
	Learning Style
	Knowledge and Performance
Domain Model - includes description of the learning content structure. The content is granulized in LOs, interconnected in a ontology of the knowledge domain. LOs and ontology are described by metadata (*Content Metadata* sub-model) according IEEE LOM specification and Ontology Metadata Vocabulary OMV proposal.	***Ontology graph***
	Learning objects
	Content Metadata
Adaptation Model - is responsible for presentation of each course storyboard as a directed graph (*Narrative Storyboard* sub-model), metadata (link annotations and assessment thresholds) of each storyboard graph (*Narrative Metadata* sub-model) and logic rules for passing over particular graph (*Storyboard Rules* sub-model).	***Narrative Metadata***
	Narrative Storyboard
	Storyboard Rules

The model splits the hierarchical structure into two levels. At first level, the model assures a clear distinction between Learner, Domain and Adaptation sub-models. At second level, each one of these models is divided into three others sub-models. As shown in table 1, the Learner model describes profile of each learner such as her/his goals and preferences, knowledge and performance and learning styles. For each individual learner character, the model defines learning style such as activist, theorist, reflector, or pragmatist or, most often, as a mix of them. Thus, the learning style can be polymorphic, as far as the learner usually is not fixed to a concrete style but rather possesses several ones, at different level.

The domain model contains structured learning content. It contains also three sub-models: learning content as LOs packaged according the SCORM standard (Díaz, Sicilia, Aedo, 2002), metadata about LOs and semantic ontologies organizing the content. The model allows various types of LOs to be used - narrative content, course tasks, essays, assessment questions, games, etc. Each one of them could be associated with one or more narrative content LOs. The content LOs are created by the author and, next, they are placed on course pages by the course instructor.

The adaptation model (AM) takes a central place in that structure. It contains information about courses content, semantics of the pedagogical strategy employed by them and course organization. Courses are presented by so called narrative storyboard graphs.

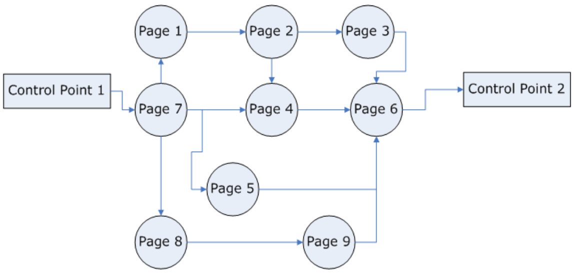


Figure 1: A sample narrative storyboard graph.

Fig. 1 presents a sample for narrative storyboard course graph. Nodes of a storyboard course graph are either narrative pages (such as Page 1, Page 2) or control pages (CP) (such as Control Point 1 and Control Point 2). Between any two CPs there are so called work paths (WP) of narrative content pages. Each one of these content pages is composed of one or several LOs. For each of these LOs the instructor can assign a parameter that specifies conditions a LO to be visible (for example, one such condition may be test results of a learner in a CP to be over a certain percentage). Information on these parameters' value is used by the adaptation engine in adaptive content delivery. Moreover the instructor may define a weight of a WP for each learning style. Therefore a particular WP may be suitable for one or several learning styles. The adaptation engine determines which WP is most appropriate for a particular learner based on these weight and data from the Learner model. The control pages are used for assessment of current knowledge and performance for a learner, by automatic test generation. This test is composed of questions corresponding to the LOs in the pages, which the learner is visited. The obtained assessment result is used for update of WP weights.

The conceptual model sketched over proposes many advantages, especially in assuring strong independence between learner profile, author content and pedagogical strategy (Vassileva, Bontchev, 2009). Moreover, it provides support of different families of learning styles, content metadata, and adaptive rule metadata.

3 TRADITIONAL STAGES IN ADAPTIVE COURSEWARE DESIGN AND DELIVERY

The traditional workflow of adaptive courseware design and delivery includes three main stages as shown in fig. 2:

- Authoring of courseware LOs (usually organized in domain ontology);
- Instructional design of an adaptive course;
- Adaptive courseware delivery – done in various ways with different delivery parameters controlling the adaptation engine.

As far as each of these three phases supplies results for the next one, it is very important to plan the work of authors, instructors and supervisors in a coherent way. In that sense, authors should designed many domain LOs being of different complexity level and of various types suitable for any of the learning styles. Content authors are supposed to do it in order to provide instructors with e-learning courseware for constructing various working paths appropriate for different learner's characters. As well, instructors should set appropriate metadata and parameters for the course pages in order to control courseware delivery with effective adaptation towards learning styles and assessment results. As far as this is very difficult to be obtained in a pure sequential workflow, transitions from each one of the phases to another should be allowed.

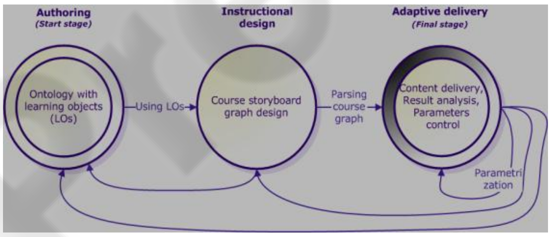

Authoring
(Start stage)
Instructional design
Adaptive delivery
(Final stage)
Ontology with learning objects (LOs)
Using LOs
Course storyboard graph design
Parsing course graph
Content delivery, Result analysis, Parameters control
Parametrization

Figure 2: Stages in traditional adaptive courseware design and delivery.

3.1 Content Authoring

The authoring process involves content author as a creator of LOs for a given domain. In many other approaches, authors of domain LOs have to design a great number of LOs of various types such as formal theory, informal LOs, examples, tasks, essay topics, quests, quizzes, mazes, etc. They have to do that in order to feed instructors with e-learning courseware material sufficient for construction of various working paths for different combinations of learning styles.

Fig. 3 represents a distribution of LOs types in a two dimensional space in accordance with their appropriateness to several learning styles. The plane is formed by the four learning styles according Honey and Mumford (Bontchev, Vassileva, 2006). Within this family of styles, the activist is a complimentary style to the theorist and, also, the pragmatist is the opposite style to the reflector. We have disposed various types of learning objects over the plane according their suitability for a learner being dominated by a given learning style or by a combination of two learning styles (the most easy case). For sure, given learning character may be composed by all the four styles – in this case, various types of LOs may be proposed to the learner as far as they are suitable for any of the styles.

The distribution shown in fig. 3 is a fruit of our practical experience and does not pretend to be punctual or validated according instructional theory. In other words, we would like just to attract readers' attention to typification of LOs according their appropriateness to learning styles. Thus, LOs produced during the authoring phase are used within instruction design on different working paths within the narrative storyboard graph, in order to satisfy learner expectations.

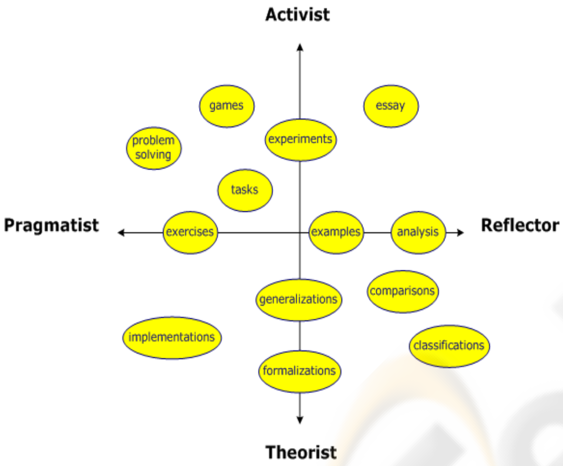


Figure 3: Distribution of LOs types according their suitability for learning styles (Bontchev, Vassileva, 2009).

3.2 Course Storyboard Design

The instructor tool is a Flex application for creation via Internet of courses adaptable to different users with specific learning styles. The courses are composed in terms of interconnected pages represented as nodes of a narrative storyboard. The narrative storyboard graph is to be processed by the adaptation engine (AE) in order to choose the best working path for a particular user. Content pages can be easily modified by drag and drop of available learning objects. Fig. 4 shows instructors drag action from learning objects browser where they are organized in an ontology graph as defined by the author. In the course graph, there is one terminal vertex that represents a control page, i.e. course exam. A course exam is generated automatically by choosing some of the questions related to the learning objects shown on pages of the work path leading to that CP (as far as questions are designed by the course author and linked to correspondent LO within the ontology tree). Thus, the instructor is not responsible for construction of assessment tests. To tune the course feedback, he/she can adjust CP thresholds values, i.e. level of assessment results for passed exam.

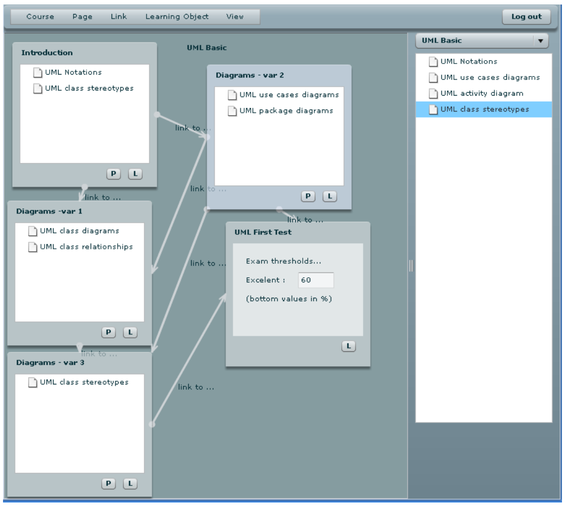
Instructor has also the responsibility to annotate page links and to set page weight parameters for each of the learning objects for given page. These page parameters are used for controlling the adaptive content selection and, therefore, are very important for tuning the system. The supervisor of AE may match parameters value to assessment result and, thus, he/she is able to control appearance of LOs for any particular learner. If the parameter of a LO within the page has high value and the learner has shown high performance at the last CP, this LO should be viewed to such a learner. Thus, when learner asks for the next page, adaptive engine may hide some objects that are not important for this user. Links annotation labels can be added also by instructor to influent user's decision when a particular user is choosing among several links. If a learner abandons the work path determined by AE (by clicking on a link leading to another page outside of the path), the AE continues tracking pages the user has passed through giving the user ability to return back to the path by adding the link "Return back to the proposed path" to each of the pages.

The instructor uses a Web based client application developed in Adobe FLEX 3, as a rich internet application while the server-side of the application is developed in Java EE. Instructors may perform any action concerning creation and update of narrative storyboard including creating courses, creating pages, filling pages with learning objects, interconnecting pages, adjusting learning objects characteristics, setting link annotations, adjusting exam thresholds, and checking user feedback.

While editing narrative storyboard, the instructor has the responsibility to annotate page links and to set page weight parameters for each of the learning objects population the page. These page parameters are used for controlling the adaptive content selection.

The instructor can parameterize the level of difficulty of a particular learning object. This parameter provides information to the adaptive engine whether or not to show a given learning object to a particular student with shown knowledge level.

Thus, given work paths created by the instructor are appropriate for students with pronounced learning style. For example students, who can be determined mainly as theorists, will receive content materials only for this learning style such as formalizations, generalizations, etc.



Exam thresholds...

Excelent : 60

(bottom values in %)

Figure 4: View of the instructor tool.

3.3 Adaptive Content Delivery

Adaptive content delivery is controlled by a software engine assuring adaptability of courseware content. Line other approaches (Weber, Hans-Christian, Weibelzahl, 2001), adaptation takes place mainly on two levels - adaptive content selection and adaptive navigation:

- Workflow controlling adaptive content selection – by means of the administration module, it is possible to configure start/stop of content adaptation or of navigation adaptation, how many questions to generate on a CP, and which LOs to be visible at a given page for learner with given assessment results. As well, supervisors can use the module for monitoring to track the effectiveness of adaptation.
- Workflow controlling adapting navigation – first at the beginning of a new WP (here the engine chooses the path of greatest weight (computed by the engine itself); next, at the end of the current WP - involving updates of the weights of the traversed path and determining whether the student can continue forward or to return to the start of the path.

4 AUTOMATIC CONSTRUCTION OF STORYBOARD GRAPHS

The opposite approach of constructing storyboard graphs by instructors is that one of automatic sequence construction in a dynamic way. In fact, this approach excludes the instructional design as an intermediate stage of adaptable courseware production (fig. 2).

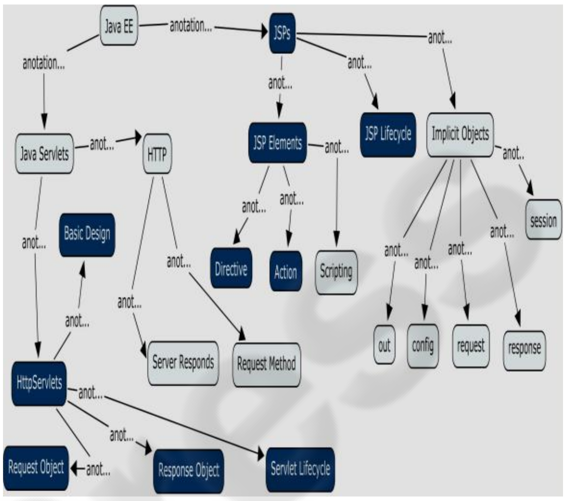


Figure 5: Selection of LOs within an ontology.

For this goal, the learner is supposed to select within the ontology all the sub-domains he/she likes to learn. The next figure presents a part of our ontology of Java EE LOs, where a learner has selected some of them (shown in dark ovals) simply by pressing the mouse. A mouse click over a LO selects it and all its sub-type LOs below so if the learner would like to select only part of them he or she has to click over the rest of LOs. E.g., the LO named “JSPs” is selected but its sub-type LOs “Scripting” and “Implicit Objects” are not (fig. 5).

After selection of desired LOs, the automatically generated storyboard for the particular learner will include the selected LOs from top to down and from left to right. In fact, main narrative LOs will be shown to the learner but LOs of other types will be present only if they are appropriate for this learner character. The adaptation engine will track again the shown LOs in terms to generate the final assessment; however, learners are free to ask the engine for intermediate tests at any moment.

5 DISCUSSIONS

The paper (supported by the ADOPTA project funded by the Bulgarian National Science Fund under agreement no. D002/155) has presented two, orthogonally opposite approaches for construction of courseware content adaptable to learner styles of individual learners. The fist one requires availability of course instructor, who uses an instructor tool for constructing storyboard graph of the course. In particular, the ADOPTA instructor tool allows instructors to create within the storyboard different work paths for different learner's characters, i.e. characters pertaining to different learning styles. In such a way, learners who are predominantly activists, theorists, pragmatists or reflectors, will receive partially different courseware content adapted to their personal learning style. This is achieved by means of adaptive navigation through the storyboard graph which is controlled by the ADOPTA adaptation engine. In the same time, LOs on the pages shown to different learners may vary according their complexity and achieved individual results. Thus, there are two important issues to be pointed out here:

1. The instructor is responsible for setting the control points and the WP leading from one CP to another;
2. The instructor selects LOs allocated on pages of given WP according their type (suitable for given learner character) and their complexity.

The second approach of automated generation of storyboards (i.e., automated sequencing) is very promising, as far as it is much cheaper and faster. Moreover, it allows learners to state explicitly their goals by selecting sub-trees on the ontology with desired sub-domain LOs. As well, learners are not supposed to make control assessment tests in predefined control points – instead, they may ask the adaptation engine to generate assessment questions at any page of the sequencing. Thus, the automated generation of storyboards is more promising in terms of adaptation flexibility. On other side, storyboards created by instructors follow a pedagogical strategy and better balance between LOs types and complexity which makes them obsolete for many specific cases.

REFERENCES

Bontchev B., Vassileva D., 2009. Adaptive courseware design based on learner character, *Proc. of Int. Conf. on Interactive Computer Aided Learning (ICL2009)*, 23-25 Sept., 2009, Villach, Austria, pp.724-731.

Brusilovsky, P., Schwarz, E., Weber, G., 1996. A Tool for Developing Adaptive Electronic Textbooks on WWW, *Proc. of WebNet'96 - World Conf. of the Web Society*, October 16-19, 1996, AACE, pp. 64-69.

Claus, P., Holohan, E. 2009. Applications of Semantic Web Technology to Support Learning Content Development, *Interdisciplinary Journal of E-Learning and Learning Objects*, Volume 5, 2009, pp. 1-25.

Dagger, D., Wade, V. & Conlan, O., 2005. Personalization for All: Making Adaptive Course Composition Easy, *Special issue of the Educational Technology and Society Journal*, IEEE IFETS.

De Bra, P., Smits, D., Stash, N., 2006. Creating and Delivering Adaptive Courses with AHA!, *Proc. of EC-TEL*, Springer LNCS 4227, pp. 21-33.

De Bra P. at al, 1999. AHAM: A Dexter-based Reference Model for adaptive Hypermedia, *Proc. of the ACM Conf. on Hypertext and Hypermedia*, pp. 147-156.

Díaz, P., Sicilia, M.A. and Aedo, I., 2002. Evaluation of Hypermedia Educational Systems: Criteria and Imperfect Measures, *Proc. of the Int. Conf. on Computers in Education*, USA, 2002, pp. 621-626.

Griffiths, D., Beauvoir, P., Liber, O., Barrett-Baxendale, M., 2009. From Reload to ReCourse: learning from IMS Learning Design implementations, *Distance Education*, Vol. 30, Issue 2, pp. 201 - 222.

Vassileva D., Bontchev B., 2006. Self adaptive hypermedia navigation based on learner model characters, *Proc. of IADAT-e2006*, Barcelona, Spain, 2006, pp.46-52.

Vassileva D., Bontchev B., Chavkova, B., Mitev V. 2009. Software Construction of an Authoring Tool for Adaptive E-learning Platform, *Proc. of BCI'2009*, Thessaloniki, Greece, pp.187-192.

Vassileva D., Bontchev B., 2009. Adaptation engine construction based on formal rules, *Proc. of CSEDU 2009*, ISBN 978-989-8111-82-1, Vol.1, Lisbon, Portugal, March 23-26, 2009, pp.327-332.

Vassileva, D., B. Bontchev, S. Grigorov, 2009. Mastering Adaptive Hypermedia Courseware, *Acta Electrotechnica et Informatica*, Vol. 9 (1), pp. 57–62.

Vrakas, D., Tsoumakas, G., Kokkoras, F., Bassiliades, N. 2007. PASER: a curricula synthesis system based on automated problem solving, Int. J. of Teaching and Case Studies, ISSN:1749-9151, Vol. 1, pp. 159-170.

Vassileva, J. 1997. Dynamic Course Generation on the WWW, *Proc. of 8th World Conf. on AI in Education (AI-ED97)*, Knowledge and Media in Learning Systems, Kobe, Japan.

Weber, G., Hans-Christian, K., and Weibelzahl, S., 2001, Developing Adaptive Internet Based Courses with the Authoring System NetCoach, *Hypermedia: Openness, Structural Awareness, and Adaptivity*, LNAI 2266, ISBN: 978-3-540-43293-7, pp. 226-238.