A New Framework for Dynamic Adaptations and Actions*

Carsten Ullrich[1], Tianxiang Lu[2], and Erica Melis[2]

[1] Shanghai Jiao Tong University, Haoran Building, 6/F
ullrich_c@sjtu.edu.cn
[2] German Research Center for Artificial Intelligence
Tianxiang.lu,Melis@dfki.de

Abstract. Adaptive course generation is more flexible if it includes mechanisms deciding just-in-time which exercises, which external resources, and which tools to include for an individual student. We developed such a novel delivery framework (called Dynamic Items) that is used by the web-based platform ActiveMath. We describe the framework and discuss several new applications of Dynamic Items for an individual student.

1 Introduction

ActiveMath [9, 10] is a Web-based intelligent learning environment for mathematicswhose course generator, Paigos, uses pedagogical knowledge to generate a sequence of learning objects that is adapted to the learners' competencies and other variables such as learning goals [11]. It uses metadata of the learning content as well as information from ActiveMath's student model that is available at generation time.

In course generation, the course is generated completely before it is presented to the learner. This early generation has the advantage that the course and its structure can be visualized to the learner. In addition, the student can navigate freely through the course. In a generated course, the structure and order of the learning objects does not change, which avoids confusion of the learner as reported in [4].

This differs from course sequencing, which dynamically selects the most appropriate resource at any moment, i.e., step by step. The benefit of this approach is that it can react to the student's progress. However, this local approach,makes it hard to convey information about the structure of a course and the learning sequence can not be presented to the learner. Moreover, it prevents the generation of essentially equal courses which only differ in places, e.g., for students in one classroom.

In this article, we describe the Dynamic Item framework and applications that we developed to combine the advantages of course sequencing and the early

* This work was supported by LeActiveMath, funded under FP 6t of the European Community (Contract Nr. IST-2003-507826) and by the DFG project ALoE.

generation of a complete course. It starts with a brief introduction of the framework and then presents the implemented instances of Dynamic Item and their educational purposes. Finally, related work and conclusions summarize how this differs from other approaches and what was achieved.

2 Dynamic Item Framework's Brief Introduction

In the current course generation, the result (a TOC) contains both learning object references and Dynamic Items. Dynamic Items are abstract learning objects which will be instantiated at presentation time by a component in Dynamic Item Framework. The Dynamic Item Framework consists of three stages: *generation stage*, *adaptation stage* and *presentation stage*. Either fetched from a persistent pre-authored content repository or generated by different learning services introduced in §3, the Dynamic Item elements are provided as middle products before adaptation. Triggered by user actions (e.g., opening a page containing Dynamic Item), the Dynamic Item Transformer renders the Dynamic Items to ordinary learning objects, taking into account up-to-date user information. The resulting learning objects are then transformed into the presentation format selected by the user e.g., HTML.

3 Applications of Dynamic Item

In this section, we illustrate the applications of Dynamic Items in ActiveMath.

3.1 Dynamic Tasks

The most frequent application of Dynamic Item is the dynamic course generation based on dynamic task expansion. Course generation can stop at a level that (abstractly) specifies what kind of learning objects should be selected, *dynamic task*, but does not specify which ones.

Later, at presentation time, when the learner first visits a page that contains a dynamic task t this is passed to the course generator that then assembles the sequence of resources that achieve t. The resulting identifiers of learning objects replace the dynamic task in the course structure with those learning objects. Hence, when the page is revisited, the elements do not change. This means a course is partly static, partly dynamic and, thus, the goal of presenting the complete course to the learner while preserving dynamic adaptivity.

One advantage of dynamic tasks is that they can be used by authors as well. They can manually compose courses, where parts of the course are predefined and others dynamically computed. In this way, an author can use the best of both worlds: she can compose parts of the course by hand and at the same time profit from the adaptive capabilities of the course generator. This also addresses situations in a classroom, where a teacher mostly wants to provide the same material (e. g., definitions, examples) for every student (important for

communication about the material with and among students) and at the same time wants to take advantage of individually selected exercise sequences at places (for more or less training as well as for adjusting the difficulty of problems). This is something a teacher can hardly manage for 20–30 students in parallel, but is easily realized with Dynamic Item.

3.2 Learning Services

Within advanced learning environments such as ActiveMath, the student is able to use external services embedded in the course. When involving the external tools, the system should be able to parameterize the call according to the current performance of the student. This is achieved by Dynamic Item in the following way.

When the learner visits a page that contains a Dynamic Item for a learning-support service, the presentation system converts the it into a link and displays it. The link is generated based on the information enclosed in the Dynamic Item element and on values obtained from the student model.

The following describes how three services were integrated using the service Dynamic Item: an Exercise Sequencer, a Concept Mapping tool and an Open Learner Model.

Exercise Sequencing The Exercise Sequencer presents to the learner a dynamically selected sequence of exercises whose selection strategy is parametrized, e.g., as mastery learning that leads the student towards a higher competency level.

This functionality differs from the exercise selection of Paigos because the Exercise Sequencer selects an exercise at the time of first visit, presents it to the learner in a separate window, provides information about the learner's problem-solving progress and terminates or selects a new exercise for a new cycle. The selection algorithm is based on competency levels [7].

Within this interactive sub-environment, the learner can interact with a dynamically selected sequence of exercises until he/she reaches a set learning goal.

External Learning Tools External learning tools are similarly integrated into the courses generated by ActiveMath using Dynamic Item. An example is the interactive Concept Mapping Tool, iCMap [8], that helps the learner to reflect on his mathematical knowledge. It supports the visualization and construction of structures for a mathematical domain. It supports the learning process by verifying the concept map constructed by a student and by suggesting reasonable changes to the created map. It is called with an instantiated, parametrized exercise that is chosen dynamically by Paigos.

Another service that can be included is the Open Learner Model (olm), which provides learners with a possibility to inspect and modify the beliefs that the learner model holds about the mastery or competencies of the student.

3.3 Dynamic Text Generation

Narrative Bridges. Similar to an advanced organizer [1], a dynamically generated text should prepare the student's mind to what he has to expect and how this is connected to his previous study. These texts cannot be authored manually, since it is practically impossible to cater for all possible choices and histories of students. Therefore, ActiveMath includes a template-based dynamic generation of bridging texts, which (1) explain the purpose of a course or a section at an abstract level. They make the intent of sections and the structure of a course explicit, they provide cues that the learners can use to remember and set the stage learning processes. (2) By linking different sections they provide coherence that a mere sequence of educational resources might lack.

In the case of Dynamic Items of type `text`, the service uses parameters to determine the adequate text template t and returns an OMDoc element whose text body consists of t. If a template is available in several languages, a specific text body is generated for each language (in case that the user changes his/her language profile any time later). Based on a template of a bridging text a controller responsible for the presentation calls the service specified in the Dynamic Item and passes the remaining attributes and sub-elements as parameters.

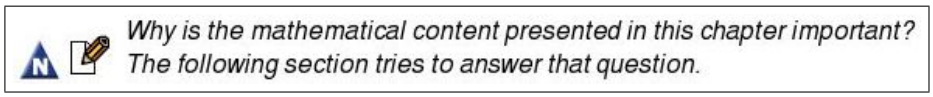

Fig. 1. A transformed bridging text

Fig. 1 shows a type of bridging texts after HTML transforming. The text is highlighted by a frame box in order to convey to the learner that the text is on a different level of abstraction than the remaining content displayed on the page.

External Resources. In order to provide an opportunity of self-regulated learning, a student should be able to include additional learning objects on demand in his personal course .

Since Dynamic Items can provide automatically generated text according to given parameters including hyperlinks, Dynamic Item can also be used to include external learning resources referenced by a link.

A student can easily add an external resources (e.g. entries in Wikipedia) she found and add it to the current course. ActiveMath's assembly tool [5] uses this functionality to add user-selected content. This includes not only texts but also multimedia content (e.g., videos) dragged link from internet. The technological means for this functionality are Dynamic Items for generating text including hyperlinks.

4 Related Work

Previous course sequencing such as the Dynamic Courseware Generator [12] selects the next page dynamically at the time the student requests it. While this allows better reactivity, the learner cannot see and use the structure of the complete course for learning.

Our work is different from Adaptive Hypermedia systems such as AHA! [2] which focuses on adapting an individual hypertext document. Whenever the user accesses a concept, a set of rules adapts the resulting document. Our approach uses a book metaphor: a complete course is generated and navigation is unrestricted so that the user can visit each page of the course any time. In such a setting, our mechanism can add parts dynamically to a previously generated course.

Selector [6] first determines the skills/concepts to be taught and then selects or constructs the required learning object. This is very similar to our approach, with the exception of dynamic tasks which allows PAIGOS to interrupt the planning process and select the specific learning objects at a later.

KnowledgeTree [3] and its extension ADAPT^2 is a distributed architecture for adaptive e-learning that integrates different learning services. A teacher can author a course and add references to static and dynamic learning objects (service calls). Our framework allows the automatic generation of courses, including the selection of such services. Automatic generation in KnowledgeTree might be possible, too, but to our knowledge has not been investigated.

Compared to existing work, our approach focuses on an abstract representation of service invocation that is easily authorable and that can be created manually by human authors and automatically during course generation.

5 Conclusion

This paper presents instances of Dynamic Item to provide just-in-time adaptivity to a student in a technology-enhanced learning environment. The idea is to separate the generation of appropriate constraints from the inclusion of the actual learning material.

The course generator decides where a Dynamic Item should be added and what kind of Dynamic Item should be added, its type and further constraints. Dynamic Items enable a persistent storage of information about pedagogical goals and constraints processed during course generation.

Usually, pedagogic information is available at generation time only and lost afterward. Dynamic Items can store this information, which provides a context for each learning object eventually presented in the course.

The implemented Dynamic Item framework applies this idea and integrates several features into the ACTIVEMATH system, such as dynamic tasks, learning services and text generation. ACTIVEMATH including its Dynamic Item has been used by hundreds of students so far.

References

1. David Ausubel. *The Psychology of Meaningful Verbal Learning*. Grune & Stratton, New York, 1963.
2. Paul De Bra, David Smits, and Natalia Stash. Creating and delivering adaptive courses with aha! In Wolfgang Nejdl and Klaus Tochtermann, editors, *Innovative Approaches for Learning and Knowledge Sharing, First European Conference on Technology Enhanced Learning, EC-TEL 2006, Crete, Greece, October 1-4, 2006, Proceedings*, volume 4227 of *Lecture Notes in Computer Science*, pages 21–33. Springer, 2006.
3. Peter Brusilovsky. Knowledgetree: a distributed architecture for adaptive e-learning. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 104–113, New York, NY, USA, 2004. ACM Press.
4. P. De Bra. Pros and cons of adaptive hypermedia in web-based education. *Journal on CyberPsychology and Behavior*, 3(1):71–77, 2000.
5. Martin Homik. Assembly Tool. Deliverable D37, LeActiveMath Consortium, jun 2006.
6. Ian O. Keeffe, Aoife Brady, Owen Conlan, and Vincent Wade. Just-in-time generation of pedagogically sound, context sensitive personalized learning experiences. *International Journal on E-Learning*, 5(1):113–127, 2006.
7. E. Klieme, H. Avenarius, W. Blum, P. Dbrich, H. Gruber, M. Prenzel, K. Reiss, K. Riquarts, J. Rost, H. Tenorth, and H. J. Vollmer. The development of national educational standards - an expertise. Technical report, Bundesministerium fr Bildung und Forschung / German Federal Ministry of Education and Research, 2004.
8. E. Melis, P. Kärger, and M. Homik. Interactive Concept Mapping in ActiveMath (iCMap). In Jörg M. Haake, Ulrich Lucke, and Djamshid Tavangarian, editors, *Delfi 2005: 3. Deutsche eLearning Fachtagung Informatik*, volume 66 of *LNI*, pages 247–258, Rostock, Germany, sep 2005. Gesellschaft für Informatik e.V. (GI).
9. Erica Melis, Eric Andrès, Jochen Büdenbender, Adrian Frischauf, George Goguadze, Paul Libbrecht, Martin Pollet, and Carsten Ullrich. Activemath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education*, 12(4):385–407, 2001.
10. Erica Melis, Giorgi Goguadze, Martin Homik, Paul Libbrecht, Carsten Ullrich, and Stefan Winterstein. Semantic-aware components and services of ActiveMath. *British Journal of Educational Technology*, 37(3):405–423, 2006.
11. Carsten Ullrich. *Pedagogically Founded Courseware Generation for Web-Based Learning – An HTN-Planning-Based Approach Implemented in PAIGOS*. Number 5260 in Lecture Notes in Artificial Intelligence. Springer, September 2008. ISBN 978-3-540-88213-8.
12. Julita Vassileva and Ralph Deters. Dynamic courseware generation on the WWW. *British Journal of Educational Technology*, 29(1):5–14, 1998.