Just-in-Time Adaptivity through Dynamic Items[*]

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

[1] Shanghai Jiao Tong University, Shanghai
[2] German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, D-66123 Saarbruecken, Germany
ullrich_c@sjtu.edu.cn, Tianxiang.Lu, Melis@dfki.de

Abstract. Adaptive course generation becomes more appropriate for realistic usage scenarios and more flexible if it includes mechanisms deciding just-in-time which content, which exercises, which external resources, and which tools to include for an individual student. We developed such a just-in-time delivery framework (called Dynamic Items) that is used for enhancing the adaptivity of (educational) online material generated by the web-based platform ActiveMath. This paper describes the framework and discusses several new learning opportunities created by Dynamic Items for an individual student.

Keywords: E-learning and intelligent learning environments, Tailoring information presentation to the user, Supporting learning and reflection.

1 Introduction

Course sequencing dynamically selects the most appropriate resource at any moment as defined in [1]. A course is not generated beforehand but step-by-step, hence it can react to the student's current context. However, this local adaptation, with its transitions from resource to resource makes it hard to convey information about the structure of a course [2]. Moreover, it prevents the generation of courses which only differ in places, e.g., for a class in school.

In course generation, the course is generated completely before it is presented to the learner. This 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. However, our experience shows that a fully personalized course material that is automatically adapted to an individual learner at *creation* time does not always satisfy the needs of learners and teacher at *runtime*. Rather, there are parts and activities in a course which need runtime adaptation, e.g.,

(1) in a classroom, a teacher mostly wants to provide the same material for every student (important for communication about the material with and among

[*] The described work was supported by the LeActiveMath project, funded under by the European Community (Contract Nr. IST-2003-507826) and by ALoE funded by DFG (ME 1136/7-1). The authors are solely responsible for its content.

G.-J. Houben et al. (Eds.): UMAP 2009, LNCS 5535, pp. 373–378, 2009.

students) and at the same time wants to take advantage of dynamic adaptation at places (for more or less training and for adjusting the difficulty of problems).

(2) While a student learns his competencies change and, hence, a dynamic selection of learning objects, especially of exercises, makes sense.

(3) For self-regulated learning, a student should be able to include additional learning objects in his personal course on demand.

(4) Similar to an advanced organizer [3], a dynamically generated text (organizer) should prepare the student's mind to what he has to expect and how this is connected to his previous learning.

(5) For access to external tools, the eLearning system should be able to parameterize the call according to the current performance of the student.

To address these issues, we developed the generic framework Dynamic Item for dynamic decisions and implemented it in ActiveMath. It combines adaptive course generation with dynamic features some of which we know from adaptive hypermedia (dynamic selection and sequencing). In the remainder of the article, we describe the Dynamic Item framework and some instances. After the preliminaries, we discuss the principles and the architecture in §3. We also present details of different transformers of Dynamic Items and describe how each serves educational purposes. Finally, conclusions and related work summarize what was achieved and how this differs from other approaches.

2 Preliminaries

ActiveMath [4] is a Web-based intelligent learning platform for mathematics, which has been developed since 2000 at the Saarland University and DFKI. It uses an extension of OMDoc, a knowledge representation for mathematical documents [5] to encode its learning objects. OMDoc consists of different types of elements, such as definition, symbols, example, exercises, text, etc.

The pedagogical knowledge of ActiveMath is implemented in its "tutorial component". Its sub-component "course generator" (CG) [2] generates courses adapted to a particular learner, based on metadata of the learning content as well as information from ActiveMath's student model that is available at generation time. Based on competency values from the student model, the CG searches for appropriate learning objects which satisfy certain constraints. From the search results and depending on learning scenarios it assembles the learning objects and generates a table of contents whose elements are either a predefined learning object or a Dynamic Item.

3 The Dynamic Item Framework

Dynamic Items are abstract learning objects and a parameter specifies their type In contrast to standard OMDoc elements, a Dynamic Item can be dynamically generated by using a context defined by up-to-date pedagogical and user information. A Dynamic Item is always re-generated when executed, i. e., whenever the learner opens a page that contains a Dynamic Item. This property allows to keep the presented information up- to-date.

3.1 Workflow of the Dynamic Item Framework

Fig. 1 shows the workflow of the Dynamic Item framework. It 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 (see §3.3), an OJDynamicItem element is input to the Dynamic Item transformer. Information about pedagogical goals and constraints processed during course generation are stored within this element. The Transformer takes the Dynamic Item element whenever the student first looks at the page that contains the element and renders the Dynamic Item representation to ordinary OMDoc, taking into account up-to-date user information. The resulting OMDoc elements are then transformed into the presentation format selected by the user e. g., HTML.

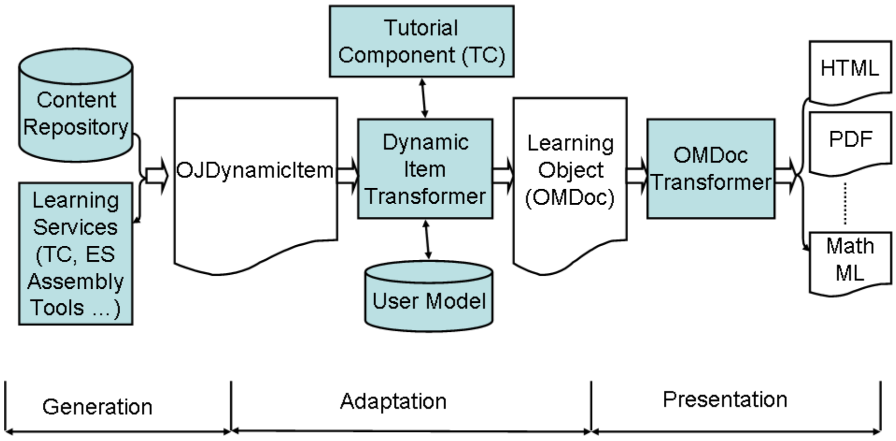

Fig. 1. Workflow of the Dynamic Item Framework

3.2 The Generic Dynamic Item Representation

Dynamic Items are represented by an extension of OMDoc which requires more complex processing than a common OMDoc. It looks as follows.

```
<dynamic-item type="type" servicename="serviceName"
              queryname="queryName">
   <ref xref="r1" /> ...<ref xref="rn" />
   <queryparam attribute="p1" value="v1" />
     ...
   <queryparam attribute="pm" value="vm" />
   <OMOBJ> ...</OMOBJ><OMOBJ> ...</OMOBJ>
</dynamic-item>
```

The *type* specifies the type of a Dynamic Item. Currently, ActiveMath supports the following three types: dynamic tasks, calls to a learning-support service and symbolic representations for text generation. The attributes servicename and queryname allow for further differentiation, for instance, starting a concept mapping service in exercise or example mode.

The optional children of a Dynamic Item specify information about the context: relevant learning objects (referenced using the ref element), mathematical terms in OpenMath format (OMOBJ), and additional parameters given as attribute-value pairs (queryparam).

3.3 Examples of Dynamic Item Transformers

We implemented various transformers of Dynamic Item in ActiveMath.

(1) *Dynamic Tasks*: these are place holders for a single or a sequence of learning objects that achieves a task. Tasks stand for pedagogical activities, such as training or illustrating a concept. During course generation, the CG selects the appropriate learning objects that help a learner to perform this activity. Using dynamic tasks, course generation can stop at a level that specifies what kind of learning object should be selected but does not specify which ones. When the learner first visits a page that contains a *dynamic task*, the task is passed to the course generator. Then, the course generator assembles the sequence of resources that achieve the task for that student. The student model is queried to provide the current competencies of the learner.

(2) Learning Services: these are place holders for services such as concept mapping tool, an Open Learner Model, a Computer Algebra System, and an Exercise Sequencer (ES) to be called dynamically. For instance, the ES can be driven by different strategies, such as based on competency levels [6] that provides exercises until the student has reached a goal level. ES also provides general feedback depending on the learner's problem solving success (e.g., whether the student has reached a higher competency level).

(3) Dynamic Text Generation (Narrative Bridges): are place holders for dynamically generated template-based narrative bridges. These bridging texts serve the following purposes: (i) they explain the purpose of a course or a section at a level of abstraction higher than the level of single learning object; (ii) they link consecutive sections and provide coherence that a mere sequence of educational resources might lack. The student's profile is used to automatically select the proper language (ActiveMath is a multi-lingual platform) and templates adequate for specific learning context (e.g., middle school or university).

(3) Dynamic External Resources: are place holders for dynamically chosen hyperlinks. Students can easily add an external resources (e.g. entries in Wikipedia) they found and add it to the current course. The assembly tool of ActiveMath [7] uses this functionality to add user-selected content to an assembled course.

(4) Gap Detection: is an application with which authors can request a course generation that characterizes learning objects, which are not yet available but would fit best for a specific set of students. Based on our approach, the gap detection application generates text characterizing the missing learning object for authors instead of inserting the actual learning object. This helps authors to fill gaps in the content.

4 Related Work

Our work is different from Adaptive Hypermedia systems such as AHA! [8]. There, the focus lies on adapting the individual hypertext document. Whenever the user accesses a concept, a set of rules adapts the resulting document. In our approach, the focus lies on the 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 enables to have part of the course completely generated and parts of it dynamically. Furthermore, unlike in AHA!, a page can contain several concepts. Selector [9] 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 our CG to interrupt the planning process and select the specific learning objects at a later. In the following section, we discuss the significant performance improvements made possible by dynamic tasks. KnowledgeTree [10] 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. The Medea framework [11] allows integrating different learning environments. Compared to our approach, Medea is more general but also carries some overhead. For instance, to integrate an intelligent service, such a service requires a student model, a set of domain concepts and a set of services implemented as web services.

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 a novel approach for just-in-time adaptivity integrated with course generation. It is more general than the common dynamic selection and sequencing of adaptive hypermedia. At generation time only the type and constraints of a Dynamic Item are determined. Dynamic Items enable persistent storage of information about pedagogical goals and constraints processed during course generation. The implementation of the Dynamic Item framework in ActiveMath enables novel features such as dynamic tasks, learning services and dynamic text and link generation.

Due to the very general design of Dynamic Item and its many usages, it is difficult to make a general evaluation of the framework. However, it significantly enhances the performance of ActiveMath, especially the longer the generated courses. Without dynamic tasks, generating a short course (six pages, 37 learning objects) takes less than half a second–an acceptable delay. But generating a long course (80 pages, 365 resources) takes five seconds. In contrast, using dynamic tasks, generating the same long course takes again only half a second. Such consideration are especially important for an actively used system like ActiveMath, which is used by hundreds of students in several schools.

References

1. Brusilovsky, P., Vassileva, J.: Course sequencing techniques for large-scale web-based education. International Journal of Continuing Engineering Education and Lifelong Learning 13(1/2), 75–94 (2003)
2. Ullrich, C. (ed.): Courseware Generation for Web-Based Learning. LNCS (LNAI), vol. 5260. Springer, Heidelberg (2008)
3. Ausubel, D.: The Psychology of Meaningful Verbal Learning. Grune & Stratton, New York (1963)
4. Melis, E., Goguadze, G., Homik, M., Libbrecht, P., Ullrich, C., Winterstein, S.: Semantic-aware components and services of ActiveMath. British Journal of Educational Technology 37(3), 405–423 (2006)
5. Kohlhase, M.: OMDoc – An Open Markup Format for Mathematical Documents. Springer, Heidelberg (2006)
6. Klieme, E., Avenarius, H., Blum, W., Dbrich, P., Gruber, H., Prenzel, M., Reiss, K., Riquarts, K., Rost, J., Tenorth, H., Vollmer, H.J.: The development of national educational standards - an expertise. Technical report, German Federal Ministry of Education and Research (2004)
7. Homik, M.: Assembly Tool. Deliverable D37, LeActiveMath Consortium (June 2006)
8. Bra, P.D., Smits, D., Stash, N.: Creating and delivering adaptive courses with aha! In: Nejdl, W., Tochtermann, K. (eds.) EC-TEL 2006. LNCS, vol. 4227, pp. 21–33. Springer, Heidelberg (2006)
9. Keeffe, I.O., Brady, A., Conlan, O., Wade, V.: Just-in-time generation of pedagogically sound, context sensitive personalized learning experiences. International Journal on E-Learning 5(1), 113–127 (2006)
10. Brusilovsky, P.: Knowledgetree: a distributed architecture for adaptive e-learning. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, pp. 104–113. ACM Press, New York (2004)
11. Trella, M., Carmona, C., Conejo, R.: MEDEA: an open service-based learning platform for developing intelligent educational systems for the web. In: Proceedings of Workshop on Adaptive Systems for Web-based Education at 12th International Conference on Artificial Intelligence in Education, pp. 27–34 (2005)