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
Carsten.Ullrich, Tianxiang.Lu, Melis@dfki.de

Abstract. We developed a just-in-time delivery framework (Dynamic Items), which adds a concept to adaptive course generation which enables features similar to but more general than those of adaptive hypermedia (adaptive selection and sequencing). It offers a generic technique for dynamic decisions in e-Learning systems. This paper describes the framework and discusses several instances of Dynamic Items in the web-based platform ActiveMath.

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

1 Introduction

Course sequencing as defined in [2] dynamically selects the most appropriate resource at any moment. 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 [11]. 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 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).

* The described work was supported by the LeActiveMath project, funded under the 6th Framework Programme of 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.

(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 advance organizer [1], 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.

Therefore, 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 some preliminaries, we discuss the principles and the architecture in §3. Then, we present the implemented instances of Dynamic Items and describe how each serves educational purposes. The implementation is evaluated in §5 and, finally, conclusions and related work summarize what was achieved and how this differs from other approaches.

2 Preliminaries

ActiveMath [9] is a Web-based intelligent learning platform for mathematics, which has been developed since 2000 at the Saarland University and at DFKI.

It uses an extension of OMDoc to encode its learning object. OMDoc is semantic knowledge representation for mathematical documents [7], represented in xml. OMDoc has different types of elements/learning object, such as definition, symbols, illustration, example, exercises, text etc.

The pedagogical knowledge is implemented in a separate module called "tutorial component". Its sub-component "course generator" (CG) [11] generates courses adapted to a particular learner, using LOM metadata of the learning content [11] 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 predefined learning object, or Dynamic Item.

3 Description of the Dynamic Item Framework

Dynamic Items are abstract learning objects and a parameter specifies their type (currently, task, learningService, text). In contrast to standard OMDoc elements, a Dynamic Item can be dynamically generated. Dynamic Items store the pedagogical and user information and this provides a context for each learning object. A Dynamic Item is always re-generated when executed. This takes place 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 Dynamic Item Framework

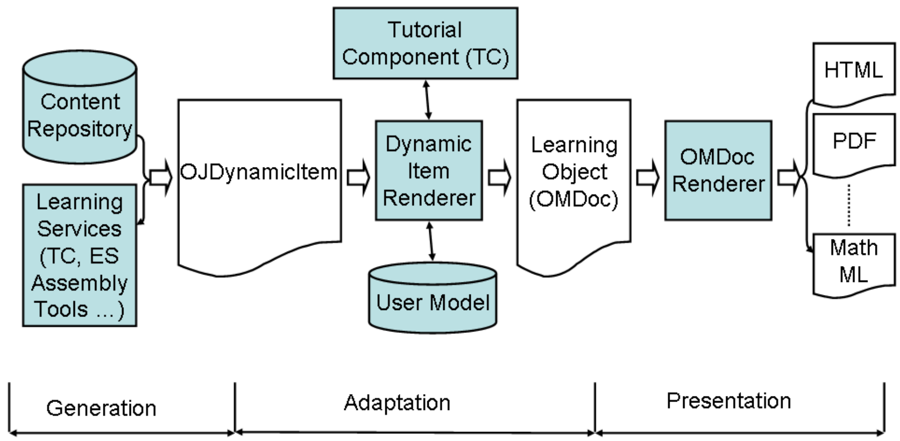

Fig. 1. Workflow of the Dynamic Item Framework

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 introduced in §4, an OJDynamicItem element is an input to the Dynamic Item Renderer. Information about pedagogical goals and constraints processed during course generation are stored within this element. Based on the Dynamic Item element, the Dynamic Item Renderer will execute the Dynamic Item whenever the student first looks at the page that contains the element or requests it. **Ednote: 'Renderer' ist hier eigentlich falsch. Besser vielleicht Transformer. Muss dann auch im Diagramm gendert werden** The Dynamic Item Renderer transforms the Dynamic Item representation to ordinary OMDoc. The resulting OMDoc elements are rendered in the presentation format selected by the user, e. g., HTML, XHTML +MathML, and PDF. This process is executed by the OMDoc Renderer [12].

Ednote: die Reviewer wollten mehr wissen ber die Architektur des 'Renderer'. Eher nicht als UML sondern allgemeiner.

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. The generic representation of a Dynamic Item 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 three types mentioned above. The attributes `servicename` and `queryname` allow for further differentiationFor instance, ActiveMath's concept mapping service [8] can be started in exercise or example mode.

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

4 Applications of Dynamic Item

We implemented various instances of Dynamic Item in ActiveMath.

(1) *Dynamic Tasks*: are represented as `dynamic-item type="dynamicTask" servicename="tutorialControl`. These are place holders for a (sequence of) learning object that achieves a task. 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: are represented by `dynamic-item type="learningService" servicename="serviceName`. These are place holders for services such as concept mapping tool, an Open Learner Model, a CAS, and Exercise Sequencer (ES) to be called dynamically. The ES can be driven by different strategies. For instance, one selection algorithm is based on competency levels [6] and 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 represented by `<dynamic-item type="text" servicename="NLG" queryname="phrase name t">`. These are place holders for dynamically generated template-based narrative bridges. These bridging texts serve the following purposes: (1) they explain the purpose of a course or a section at a level of abstraction higher than the level of single learning object. (2) 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 (remember, ActiveMath is a multi-lingual platform) and templates adequate for specific learning context (e.g., middle school or university).

(3) Dynamic External Resources**Ednote: wieso war das herausgestirchen?**: are represented by <dynamic-item type="text" servicename="???" queryname="???">. These are place holders for dynamically chosen hyperlinks. A student can easily add an external resources (e.g. entries in Wikipedia) she found and add it to the current course. The assembly tool of ActiveMath [5] uses this functionality to add user-selected content (dragged link from internet) to an assembled course.

Ednote: wieso wurde gap detection gestrichen?

5 Technical Evaluation

Due to the very general design of Dynamic Item and its many usages, it is difficult to make a general evaluation of the framework. Hence Table **??** presents (only) a technical evaluation of how dynamic task effect the overall performance of ActiveMath.

Number of concepts	1	4	8	12	16	20
Expanded	429	1204	1875	2562	3360	4834
Dynamic tasks	205	288	415	446	617	617

Table 1. Required time of course generation for increasing number of fundamentals

For increasing numbers of concepts Table 1 shows the time required for course generation (in milliseconds). In the table, the condition Expanded provides the time required for a completely expanded course, i. e., all dynamic tasks are directly executed. The generation of the smallest course (for a single concept, six pages and 37 learning object in total) takes less than half a second. The generation of a course for 20 concepts takes less than five seconds, an obvious delay in a Web-based environment. For Dynamic Tasks the comparison illustrates that not planning the complete plan t once can result in a significant performance improvement: a course for 20 concepts is generated in slightly more than half a second. This is important for the usability of course generation.

Similarly, the generation of a completely expanded course takes 30 second on average. Using dynamic tasks, it takes about 8 seconds.

6 Related Work

Adaptive runtime selection and sequencing has been realized previously, e.g., in AHAv3 and APeLS. **Ednote: das kann man nur schreiben, wenn man die refs einfgt. Es wre zu frech den Reviewer zu kopieren...** Our Dynamic Item approach is more abstract and general and includes more activities included into courses. Moreover, Dynamic Item combines course planning with dynamic selection and sequencing and, hence, the student also benefits from the guiding structure provided by course generation.

[10] provides adaptivity through an encapsulation of learning information and user profile and trace information in a Sharable Content Object (SCO). User profile information can be accessed by *getValue* from the API of SCORM RTE. The trace of student work can be stored in the *intersection* field of a SCO. **Ednote: Tianxiang, I dont understand the adaptivity here??** T-MAESTRO [10], is a system for personalized learning experiences. It uses self-adaptive SCO to store meta-level learning information just like the Dynamic Item representation **Ednote: verstehe ich nicht. Und welche meta-information gibt es in Dynamic Item??**. It seems that the function of a server in the architecture of T-MAESTRO is similar to our Dynamic Item Renderer **Ednote: oben schon gesagt: schlechter Name. Und das Ziel von SCO ist doch ein anderes, nmlich Informationen ber viele Benutzer zu tragen nicht nur ber einen. Oder habe ich das falsch in Erinnerung???**.

7 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 /di 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.

ActiveMath including its Dynamic Item has been used by hundreds of students so far. A technical evaluation has shown that Dynamic Item makes the course generation more efficient.

Ednote: why does Latex start a new page here?

Bibliography

[1] D. Ausubel. *The Psychology of Meaningful Verbal Learning.* Grune & Stratton, New York, 1963.

[2] P. Brusilovsky and J. Vassileva. Course sequencing techniques for large-scale webbased education. *International Journal of Continuing Engineering Education and Lifelong Learning*, 13(1/2):75–94, 2003.

[3] P. De Bra. Pros and cons of adaptive hypermedia in web-based education. *Journal on CyberPsychology and Behavior*, 3(1):71–77, 2000.

[4] E. Freeman, E. Freeman, B. Bates, and K. Sierra. *Head First Design Patterns.* O'Reilly, October 2004.

[5] M. Homik. Assembly Tool. Deliverable D37, LeActiveMath Consortium, jun 2006.

[6] 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.

[7] M. Kohlhase. *OMDoc – An Open Markup Format for Mathematical Documents.* Springer Verlag, 2006.

[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] E. Melis, G. Goguadze, M. Homik, P. Libbrecht, C. Ullrich, and S. Winterstein. Semantic-aware components and services of ActiveMath. *British Journal of Educational Technology*, 37(3):405–423, 2006.

[10] M. Rey-Lopez, P. Brusilovsky, M. Mecawy, R. Diaz-Redondo, A. Fernandez-Vilas, and H. Ashman. Resolving the problem of intelligent learning content in learning management systems. *International Journal on E–Learning*, 7(3), 2008.

[11] C. 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] C. Ullrich, P. Libbrecht, S. Winterstein, and M. Mühlenbrock. A flexible and efficient presentation-architecture for adaptive hypermedia: Description and technical evaluation. In Kinshuk, C. Looi, E. Sutinen, D. Sampson, I. Aedo, L. Uden, and E. Kähkönen, editors, *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004), Joensuu, Finland*, pages 21–25, 2004.

[13] J. Vassileva and R. Deters. Dynamic courseware generation on the WWW. *British Journal of Educational Technology*, 29(1):5–14, 1998.