Course Generation as a Hierarchical Task Network Planning Problem

Carsten Ullrich

Course generation automatically assembles courseware adapted to the learner's competencies and learning goals. Previous course generators offered only a limited representation a pedagogical knowledge and did not take full advantage of service-oriented architectures. This dissertation proposes a course generation framework (called Paigos) that enables the formalization and application of complex and realistic pedagogical knowledge. Compared to previous course generation, this approach generates structured courses that are adapted to a variety of different learning goals and to the learners' competencies. Additionally, Paigos is service-oriented, thus allowing the integration of learning-support services into generated courses in a generic and pedagogically sensible way and enabling third-party learning environments access to its functionality using a Web-service interface. Paigos was evaluated by technical, formative and summative evaluations. The technical evaluation primarily investigated the performance to Paigos; the formative and summative evaluations targeted the users' acceptance of Paigos and of the generated courses.

1 Introduction

Course generation uses information about educational resources, the learner and his learning goals to generate an adapted sequence of resources that supports the learner in achieving his goals.

However, previous course generators cannot handle *complex* learning goals. In most course generators, the learning goal consists of the target concepts the learner wants to learn about. But during learning, a user will have different objectives: when the content is unknown to him, he requires detailed, comprehensive information. Later, he might want to rehearse the content, which requires a different course. When preparing for an exam, he wants to use a workbook, which is yet another type of course.

Such complex learning goals require sophisticated pedagogical knowledge. While early, pre-Web work emphasizes this requirement [8], it is marginalized in recent work on course generation [2, 1]. In addition, the quality and coverage of the pedagogical knowledge in most of the recent work cannot be judged due to insufficient information. Schulmeister's [5] criticism on adaptive systems in general applies to course generation as well: a large percentage of existing work neither describes the characteristics of the learner used for adaptivity nor the methods and dimensions of adaptivity that are aimed at.

In addition, none of the previous course generators has a service-oriented architecture. They cannot perform federated search (i. e., combine courses from resources found in different repositories), nor make their functionality available as a service to other systems.

This thesis [7] uses many of the possibilities offered by today's (Semantic) Web, Artificial Intelligence and technology-enhanced learning techniques to overcome these and further problems. The work contributes to service-oriented course generation and modeling of pedagogical knowledge. Several evaluations served to assess the practical value of Paigos.

2 Modeling of Pedagogical Knowledge

Paigos implements realistic pedagogical knowledge developed jointly with pedagogical experts. The implementation is based on Hierarchical Task Network planning (HTN [4]). In HTN-planning, the goal of the planner is to achieve a list of tasks, where each task is a symbolic representation of an activity to be performed. The planner formulates a plan by using methods to decompose these top tasks into smaller subtasks until primitive tasks are reached that can be carried out directly using operators.

The formalized pedagogical knowledge encodes how to generate courses that help the learner to achieve his learning goals. Paigos's domain knowledge realizes a large set of learning goals, ranging from selecting single resources such as examples and exercises to complete courses.

The basic pedagogical axioms, operators and methods developed in this thesis are pedagogically neutral. Pedagogical neutrality is important since pedagogical researchers as well as practitioners disagree on the question which pedagogical principles to use for teaching. Hence, if a course generator aims at wide-spread applicability, it should not impose any specific learning theory.

The basic knowledge encompasses the following: 1) inserting resources in a course; 2) structuring courses by opening and closing sections and by inserting dynamically generated texts that inform the learner about the purpose of sections; 3) interacting with knowledge sources, i. e., querying content repositories and the learner model; 4) combining information from knowledge sources, e. g., removing all known educational resources from a given list; 5) inserting links to learning-support services.

These primitives are employed to build up complex collections of pedagogical knowledge, so-called *scenarios*. The scenarios are *discover* new content, *rehearse* weak points, establish *connections* between concepts, *train intensively*, *train competencies* and *exam simulation*. A scenario determines the basic structure of a course, for instance, by prescribing that the presentation of an example should follow the presentation of a definition. The courses are adaptive, that is, different resources are selected depending on information from the learner model, such as his success in solving previous exercises. While the above scenarios implement a novel competency-based pedagogical approach, an additional scenario realizes a more traditional approach based on instructional design guidelines.

The courses that result from applying the formalized knowledge are structured according to pedagogical principles. This structure is made explicit by the nested sections of the table of contents and by bridging texts that are created during course generation. This structure and the bridging texts convey to the learner additional information about the learning process that he can later use to structure his own learning.

The knowledge is generic, that is, independent from the actual content. This makes the knowledge reusable and applicable to other domains than it was originally developed for.

3 Adaptivity in Generated Courses

Course generation faces a dilemma: on one hand it makes sense to generate a complete course immediately after receiving the learner's request instead of selecting and presenting one resource after another: in one-shot generation, the learner sees the how the content is structured and he can freely navigate. On the other hand, if a long time-span separates between the generation and viewing of a page, assumptions about the learner made during course generation may have become invalid, resulting in an inadequate course. Hence, if possible, the course generating should be dynamic in the sense to use the most up-to-date information about the learner that is available.

The solution realized in PAIGOS is based on dynamic subtask expansion: course generation may stop at a level that specifies what kind of educational resources should be selected but does not specify which ones. The specific resources are selected as late as possible, that is, only at the time when the learner wants to see them. An important aspect of dynamic subtask expansion is that this technique can be used by human "course generators", i. e., authors that manually compose courses: an author can define a course where parts of the course are predefined and others dynamically computed, taking the learner model into account. In this way, an author can profit from the best of both worlds: she can compose parts of the course by hand and at the same time profit from the adaptive features of the course generator.

4 Service-Oriented Course Generation

PAIGOS views all software systems that are involved in course generation as *services*. This includes the course generator, learning object repositories and additional tools that support the user during learning (called *learning-support services*).

A Web-service wrapper of PAIGOS allows accessing its course generation functionality by well-defined Web-service interfaces. This way, if a learning management system wants to offer course generation, it can reuse the functionalities made available by PAIGOS and is not required to implement the pedagogical knowledge itself. Several learning environments were connected in this manner.

Learning object repositories are treated as services that can register at PAIGOS and make their educational resources available for course generation. However, a difficulty is that the representation of resources often varies simply because different database schemas may be used in the repositories. In addition, despite standardization efforts such as LOM [6] almost every repository available uses its own description of learning objects (or at least a variant of LOM). PAIGOS uses a mediator architecture to overcome these problems. The registration process requires the repository to provide a mapping between the knowledge representation used in the course generator and the representation used in the repository. The knowledge representation used in the course generator is based on an ontology of instructional objects. Its classes allow categorizing learning objects according to their pedagogical purpose in a more precise way than existing learning object metadata specifications and it can be used for intelligent pedagogical functionalities other than course generation.

Last but not least, PAIGOS views tools that support the learning process as services, too. PAIGOS integrate these services, not arbitrarily but in a pedagogically sensible way: during the learning process, at times the usage of a tool can be more beneficial than at some other time. For instance, reflecting upon the learned concepts may be most effective at the end of a lesson because attention is not split between cognitive and meta-cognitive (reflective) activity.

5 Evaluation

Evaluations were an integral part of this thesis. A technical evaluation investigated the performance of PAIGOS under various conditions. Under optimal conditions, it takes about half a second to generate courses with an approximated reading time of about 11 hours. The results also illustrate the drawbacks of service-oriented architectures: the above figures were obtained when the latency of the services accessed during course generation was minimized. In real life, performance decreases, due to the large amount of queries send over the Web to the repositories and learner model.

PAIGOS was subject to Europe-wide formative and summative evaluations involving first and second year university students. The evaluations were performed by partners at the University of Edinburgh, University of Malaga, Universität Augsburg and Ludwig-Maximilians-Universität München. The summative evaluation shows that a majority of students prefers generated courses over manually authored courses, mostly due to the generated courses being tailored to the student's learning goals. The users also rate the selected content (in particular the exercises) as being well adapted to their competencies.

6 Example and Conclusion

The screenshot in Figure 1 shows a course generated for the scenario *discover* and the goal concepts "definition of derivative", "definition of the derivative function" and the theorem "sum rule". The page displayed on the right hand side of the figure is the second page of the course. It contains the first items of the prerequisites page: the text generated to describe the purpose of the section and the prerequisite concepts (only one is

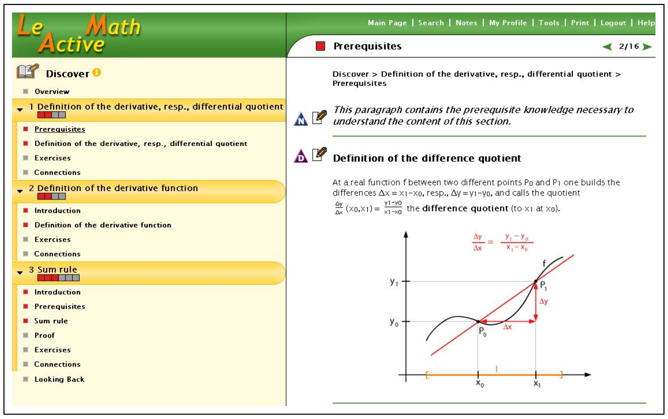

Figure 1: An example of a course of type *discover*

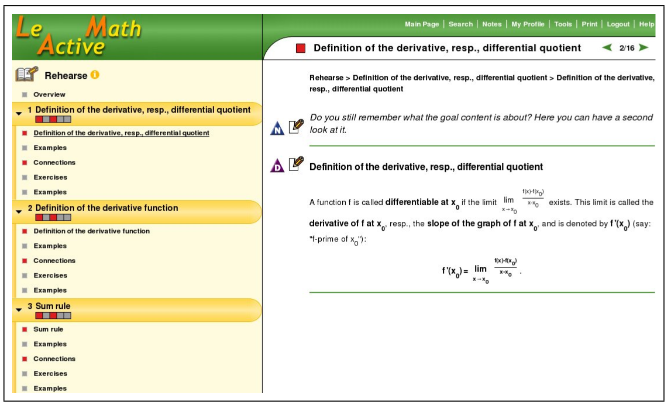

Figure 2: An example of a course of type *rehearse*

visible in the screenshot). The sections displayed in the table of contents (left) vary in the pages they contain. For instance, the first section does not contain an introduction page. The reason is that no elements could be found to be displayed in this page and therefore, the page was skipped.

Figure 2 show a course generated using the same concepts as the previous example but for scenario *rehearse*. Note that the table of contents reflects the focus of the scenario on examples and exercises.

PAIGOS is a novel approach to course generation that allows to represent complex pedagogical knowledge and uses the potential of service-oriented architectures. It is not tied to a specific learning environment and due to Web-service interfaces its functionality is publicly available.

The pedagogical knowledge used for course generation is specified as basic building blocks. From these, seven different scenarios were formalized based on learning theories, such as moderate constructivism or instructional design. This illustrates the flexibility and pedagogical neutrality of PAIGOS.

In total, PAIGOS encompasses about 300 methods. PAIGOS is integrated in the learning environment ACTIVEMATH [3] and several other environments. A demo of ACTIVEMATH is available at `http://www.activemath.org/demo.php`.

References

[1] P. Karampiperis and D. Sampson. Automatic learning object selection and sequencing in web-based intelligent learning systems. In Z. Ma, editor, *Web-Based Intelligent e-Learning Systems: Technologies and Applications*, chapter III, pages 56–71. Information Science Publishing, 2005.

[2] K. Keenoy, A. Poulovassilis, V. Christophides, P. Rigaux, G. Papamarkos, A. Magkanaraki, M. Stratakis, N. Spyratos, and P. Wood. Personalisation services for self e-learning networks. In N. Koch, P. Fraternali, and M. Wirsing, editors, *Proc. of 4th International Conference on Web Engineering, Munich, Germany*, volume 3140 of *Lecture Notes in Computer Science*, pages 215–219, Heidelberg, Germany, 2004. Springer.

[3] 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.

[4] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.

[5] R. Schulmeister. *eLearning: Einsichten und Aussichten*. Oldenbourg, München, Germany, 2006.

[6] IEEE Learning Technology Standards Committee. 1484.12.1-2002 IEEE standard for Learning Object Metadata, 2002.

[7] C. Ullrich. *Course Generation as a Hierarchical Task Network Planning Problem*. PhD thesis, Computer Science Department, Saarland University, Saarbrücken, 2007.

[8] K. Van Marcke. GTE: An epistemological approach to instructional modelling. *Instructional Science*, 26:147–191, 1998.

Contact

Dr. Carsten Ullrich
E-Learning Lab of Shanghai Jiaotong University
Shanghai Jiaotong University
1954 Hua Shan Road
200030 Shanghai
China
Email: cullrich@activemath.org
Blog: `http://bloggingullrich.blogspot.com/`
Homepage: `http://ullrich.gigacities.net/`

Bild

Carsten Ullrich is a PostDoc at the e-learning lab of Shanghai Jiaotong University. He received his Dr.-Ing. from Saarland University in September 2007 where it was awarded magna cum laude. His research interests include technology-supported learning, especially by AI techniques, Web 2.0, Semantic Web and mobile learning. During his time at the German Research Center for Artificial Intelligence (DFKI GmbH) he developed a tutorial component based on PAIGOS, which is one of the core modules of the the learning environment ActiveMath. Currently, at Shanghai Jiaotong University he is focusing on fostering active learning by combining Web 2.0 and Artificial Intelligence. He served as a reviewer for a number of conferences and journals and has contributed to several national as well as European projects.