Tutorial planning : Adapting course generation to today's needs.

Carsten Ullrich

Tutorial Planning: Adapting Course Generation to Today's Needs [1]

Carsten Ullrich
German Research Center for Artificial Intelligence, DFKI GmbH

Abstract. Most of today's course generation does not allow an in-depth, generic representation of pedagogical knowledge. However, supporting individual learners with different goals requires an elaborate representation of pedagogical expertise. In this paper, I describe a framework that adapts existing approaches for representing and using pedagogical knowledge to meet today's needs. Furthermore, I will show how in this framework several of today's problems are solved, such as the integration of distributed content, dynamic adaptivity of a generated course, new forms of interaction, and offering course generation as a service.

Keywords. Course generation, pedagogical knowledge, learning objects, adaptivity, Web services

1. Introduction

Course generation (CG) automatically assembles learning objects (LOs) to a greater unit that supports the learner to reach a given learning goal. Today's CG uses rather simplified pedagogical knowledge, e.g., the typical learning time of a LO [11]. However, to generate a course adapted to the individual learner's goals and needs requires more elaborate expertise. Representing pedagogical expertise is not a new idea [14,20], but it seems to have been neglected in the last years. In this paper, I argue to look back and adapt these ideas to suit today's needs. I will show how the introduction of *pedagogical tasks* and *methods* provides a framework that solves several of today's challenges. These challenges include: the integration of distributed content, dynamic adaptivity of a generated course, new forms of interaction, and offering course generation as a service.

Section 2 starts by summarizing the advantages of declarative pedagogical knowledge and provides the basics of my approach. Section 3 then describes in detail what is new in my approach and how existing challenges are solved. The paper closes with a comparison to related work and a description of the current state of the implementation (Section 4).

[1]The author wishes to thank all members of the ActiveMath group for fruitful discussions. Special thanks goes to Kris van Marcke and the YRT reviewers for their comments and to my supervisor Erica Melis for her valuable help. This publication was generated in the LeActiveMath project, funded under FP6, Cntr. 507826. The author is solely responsible for its content.

```
(:method (teachConcept ?c)
         ((problem ?p (for ?c))) ;; this is the precondition
         (;; the following are the sub-tasks:
          (!startSection bookTitle) (provideSituation ?p)
          (teachSubConcepts ?p) (provideResult ?p) (!endSection)))
```

Figure 1. A method for a problem-based pedagogical strategy

2. Representing and Applying Pedagogical Knowledge

A *declarative*, *generic* representation of knowledge about *tasks and methods* offers several well-known advantages [18]. *Declarativity* eases authoring and change; being *generic* allows reusability; and to distinguish between *task and methods*, i.e., between what to achieve and how to achieve it, clearly separates different kinds of knowledge. Using partly different collections of tasks and methods, various problem solving *strategies* can be realized.

Declarative knowledge can be executed in a number of ways. In my work, I decided to use hierarchical task network planning [6] (HTN), among others because it is a very efficient planning technique and offers a relatively straight-forward way for representing human expert knowledge. It incorporates heuristic knowledge in the form of the decomposition rules: A planning problem is represented by sets of tasks (*task networks*); *methods* decompose non-primitive tasks into sub-tasks until a level of primitive tasks is reached, which can be solved by *operators*.

Together with the University of Augsburg, we are developing a problem-based pedagogical strategy that relies on the "Programme for International Student Assessment" (PISA) framework, [15]. Figure 1 contains an example of one of our methods. This method is applicable in case there exists a LO that represents a "problem" for concept c. Then, the complex task of teaching concept c is decomposed into three non-primitive sub-tasks: The first one describes the context the problem; the second one teaches the sub-concepts necessary for solving p, and the last one discusses the results of solving the problem using c (e.g., possible shortcomings). One by one, these tasks are further decomposed. "!startSection" and "!endSection" are primitive tasks, which insert and close a section.

Figure 2 provides an example of a method that encodes the pedagogical knowledge of selecting an exercise that is just within the current capabilities of the learner. Even this simplified method shows that even such a very basic task can require sophisticated knowledge: If the learner has low knowledge and additionally is unmotivated, then an appropriate exercise needs to be easier than in case he is highly motivated.

The result of the planning is a sequence of LOs called *course structure*. Similar to the "organization" element of an IMS Content Package [9], it consists of nested sections with the leaves being pointers to LOs. Additionally, leaves can consists of tasks (see Section 3.2). Because in my approach tasks represent a vast range of pedagogical issues, the size of the generated courses ranges from a single element to a complete curriculum.

```
(:method (insertAppropriateExercise ?c)
         ;;preconditon
         ((learnerProperty competencyLevel ?c low)
          (learnerProperty motivation low))
         ;;sub-task
         ((insertExercise ?c easy))
         ;;preconditon
         ((learnerProperty competencyLevel ?c low)
          (learnerProperty motivation high))
         ;;sub-task
         ((insertExercise ?c medium))
         ;;more
         ...)
```

Figure 2. A method for selecting an exercise

3. Tutorial Planning

In this section, I describe my contributions to CG and how several of today's challenges are solved. Because the resulting framework offers a broader approach to the sequencing of LOs as CG, I will refer to it as *tutorial planning.*

3.1. Integration of Distributed Content by Tutorial Planning

Today, content is distributed over the Internet and made available via Web-servers or LO repositories. Integrating this content by CG involves several difficulties.

First, traditional AI planning requires that a method's preconditions are evaluated against the planner's world state. However, in a distributed environment this would require to mirror all the metadata of the content stored in the repositories. This is simply infeasible. Second, despite standardization efforts content is described in various knowledge representations and metadata formats. Third, LOs can be constructed on the fly using generators [7]. How can the tutorial planner integrate these LOs without a practically impossible explicit modeling of the generators?

To solve these problems, I applied techniques for information integration from different knowledge sources [13]. There, an additional component acts as a *mediator* between the knowledge processing and knowledge storing components. It uses a reformulation engine to transform the incoming queries into queries in the formats of the repositories and sends back the collected results. Based on this approach, I extended the HTN-planning architecture so that queries about LOs in a method's preconditions result in a call to a mediator. The "lingua franca" of the mediator uses the vocabulary of an ontology of instructional objects [19] that defines about 30 different types of learning resources. Currently, however, mappings between the ontology and the knowledge representations of the repository need to be provided by hand.

Additionally, I extended the mediator approach to handle LO generators similar to repositories: the generators check whether they can generate an LO corresponding to the query, and, if so, send back identifiers from which the LO can be reconstructed later when it is presented to the learner. In this way, the tutorial planner abstracts from the LO repositories and generators and can focus on pedagogical knowledge.

3.2. Dynamic Adaptivity of a Generated Course

CG faces the dilemma that early course generation cannot take into account how capabilities of the learner actually change. Still, generating a course as early as possible supports orientation and self-organisation of the learning process. Plan repair [21] is a possible solution, I will present another one, lazy task execution.

I extended the planner such that planning may stop at the level of specially marked tasks (lazy tasks). These tasks are inserted in the content structure like a LO. When the learner first visits a page that contains a lazy task, the task in the content structure is passed to the tutorial planner. The resulting LOs replace the task in the course structure for good (hence, when the page is revisited, the elements do not change, which avoids confusion of the learner reported in [1]). This means a course is partly static, partly dynamic.

Lazy task execution offers new possibilities for authors, too. An author can define a course structure, where parts of her 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 tutorial planner.

The method in Figure 2 is an example of a method that can serve as a lazy task. Because the motivation of the learner may change while working through a course, the lazy execution of the task *insertAppropriateExercise* can take the current state into account.

3.3. New Forms of Interaction between Learner and Content

The explicit representation of tasks and the abstract layer they introduce offers to the learner new ways to access the pedagogical knowledge of the tutorial planner. If the learner requires support while navigating through a course, she can trigger the execution of a tutorial task (e.g., *insertAppropriateExercise*), for instance by selecting them from a drop-down list. Then, the task is processed by the tutorial planner and the resulting LOs are presented to her.

This approach has the advantage that content is retrieved in a more sophisticated manner than search, sparing the learner the exact knowledge of the LO metadata. Hence, she can easily and actively request additional content, and thus, a course less resembles a traditional text book but a dynamic and extensible workbook.

3.4. Tutorial Planning as a Service

Currently, several research groups investigate the integration of e-learning (Web) services and sharing of functionalities (see, e.g., [5]). In my approach, tasks can be interpreted as a description of the functionality that the tutorial planner offers. They can therefore serve as a basis for making the functionality of a tutorial planner accessible as a service to other systems. More specifically, the tutorial planner makes itself accessible as a service by offering a set of public tasks. The tasks range from the generation of a complete course to selecting a not too difficult exercise. Other components can start the tutorial planner on these tasks and receive a content structure as a result.

Again, the method *insertAppropriateExercise* can serve as an example. Several services could make use of this task, e.g., an Open Learner Model (OLM, [4]) that while interacting with the learner diagnoses that the learner has a erroneous belief about his knowledge state. One possible remedy is to offer him an exercise he should be able to solve. Instead of encoding the pedagogical knowledge of selecting an appropriate exercise in the OLM, it can make use of the services provided by the tutorial planner.

4. Related Work and Conclusion

CG has a long history, e.g., [16,14,20,21]. Despite continued research (e.g., [17,12, 11]), to my knowledge no system paralleled the extent of explicit representation of pedagogical knowledge as done by van Marcke. My work extends van Marcke's approach and adapts it to meet today's needs.

A related strain of research is Adaptive Hypermedia (for an overview, see [3]). One of its techniques, Adaptive Presentation, allows to conditionally include text fragments. In systems like AHA [2], conditional rules are included into the hypertext document. Using a technique like lazy task execution allows moving the rules from the document to a dedicated component like the tutorial planner with the advantages described above.

Open corpus hypermedia as described in [8] allows to integrate HTML pages from different sources. Using information about the concepts a page covers (provided by the author), the system generates a trail (course) leading the learner to her learning goal. The generation takes into account the learner's knowledge and the dependency relations between the concepts, but no other explicitly represented pedagogical knowledge.

An alternative to IMS CP [9] as the output of the course generation is IMS Learning Design (LD, [10]). IMS LD describes ordered activities in learning and the roles of the involved parties. However, in our context of web-based elearning, the simpler IMS CP proved to be adequate.

To summarize, in this paper I described tutorial planning, an approach to course generation that is based on a declarative representation of pedagogical knowledge. This knowledge level provides a framework which solves several of today's problems. These include the integration of distributed content and learning supporting tools within a course, dynamic adaptivity of a generated course, new forms of interaction, and offering course generation as a service.

The implementation is currently underway. The HTN-planner JSHOP was extended and connected to a metadata mediator. Parts of the problem-based pedagogical strategy are formalized. After the implementation, several pedagogical scenarios will be formalized and tested. An evaluation will assess and help to improve the system. It is an essential goal of my research not only to theoretically but also practically advance technology supported learning.

References

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

[2] P. D. Bra, A. Aerts, D. Smits, and N. Stash. AHA! Version 2.0 – more adaptation flexibility for authors. In *Proceedings of the AACE ELearn'2002 conference*, 2002.

[3] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User Adapted Interaction*, 11(1/2):87–110, 2001.

[4] V. Dimitrova. STyLE-OLM: Interactive open learner modelling. *International Journal of Artificial Intelligence in Education*, 13:35–78, 2002.

[5] P. Dolog, N. Henze, W. Nejdl, and M. Sintek. Personalization in distributed elearning environments. In *Proc. of WWW2004*, 2004.

[6] K. Erol, J. Hendler, and D. S. Nau. HTN planning: Complexity and expressivity. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1123–1128. AAAI Press/MIT Press.

[7] S. Fischer and R. Steinmetz. Automatic creation of exercises in adaptive hypermedia learning systems. In *Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 49–55. ACM Press, 2000.

[8] N. Henze and W. Nejdl. Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education*, 12:325–350, 2001.

[9] IMS Global Learning Consortium. IMS content packaging information model, 2003.

[10] IMS Global Learning Consortium. IMS learning design specification, February 2003.

[11] P. Karampiperis and D. Sampson. Adaptive instructional planning using ontologies. In *Proc. of the 4th IEEE International Conference on Advanced Learning Technologies, ICALT 2004*, pages 126–130, 2004.

[12] 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*, volume 3140 of *Lecture Notes in Computer Science*, pages 215–219. Springer, 2004.

[13] C. Knoblock and S. Kambhampati. Tutorial on information integration on the web at the 18th National Conference on Artificial Intelligence. July 2002.

[14] W. R. Murray. Control for intelligent tutoring systems: A blackboard-based dynamic instructional planner. In D. Bierman, J. Breuker, and J. Sandberg, editors, *Proc. 4th International Conference of AI and Education*, pages 150–168, Amsterdam, Springfield VA, Tokyo, 1989. IOS.

[15] Organisation for Economic Co-operation and Development (OECD), editor. *Measuring Student Knowledge and Skills – A New Framework for Assessment*. 1999.

[16] D. R. Peachy and G. I. McCalla. Using planning techniques in intelligent tutoring systems. *International Journal of Man-Machine Studies*, 24:77–98, 1986.

[17] M. Specht and R. Oppermann. ACE - adaptive courseware environment. *The New Review of Hypermedia and Multimedia*, 4:141–162, 1998.

[18] L. Steels. Components of expertise. *AI Magazine*, 11(2):30–49, 1990.

[19] C. Ullrich. The learning-resource-type is dead, long live the learning- resource-type! *Learning Objects and Learning Designs*, 1(1):7–15, 2005.

[20] K. van Marcke. Instructional expertise. In C. Frasson, G. Gauthier, and G. McCalla, editors, *Proceedings of the 2nd International Conference on Intelligent Tutoring Systems*, number 608 in LNCS, pages 234–243. Springer, 1992.

[21] J. Vassileva. Dynamic courseware generation: at the cross point of CAL, ITS and authoring. In *Proceedings of ICCE'95, Singapore*, pages 290–297, December 1995.