An Easily Implementable Method to Support Goal-Directed Learning*

Carsten Ullrich and Aiping Chen
German Research Center for Artificial Intelligence /Saarland University

Abstract

More and more in education, the importance of learning how to learn is recognized. The faster factual knowledge changes, the more important skills on how to learn become. In this article, we describe a method for supporting the learner on one aspect of these meta-cognitive skills, namely setting learning goals and controlling their achievement. The method we propose uses data available in virtually every learning management system and, therefore, can be easily implemented and integrated in existing systems.

1. Introduction

Meta-cognition is often defined as thinking about thinking. In the domain of education, Schoenfeld suggests to regard meta-cognition as "awareness of your thinking and your progress as you're problem solving" [7].
One major meta-cognitive skill is to have a well-founded notion about one self's capabilities. In daily working and learning, such knowledge is a necessary condition to judge reliably in what time and with what effort one can achieve a specific task. A technique to train this skill is to explicitly define goals, to control the achievement of these goals, and to adapt the goals, if necessary.
Schoenfeld puts it nicely when he suggests to view meta-cognition as a management approach: "Aspects of management include (a) making sure that you understand what a problem is all about before you hastily attempt a solution; (b) planning; (c) monitoring, or keeping track of how well things are going during a solution; and (d) allocating resources, or deciding what to do, and for how long, as you work on the problem".
In this article, we will present a method that supports learners in acquiring this learning management skill. Our method uses the amount of learning materials read by the student and the time required by the student to do so. This data is available in virtually every learning management system, making, consequently, our method implementable almost everywhere, in contrast to other, more elaborated systems such as SciWise [10] or Andes [2].
We will start by providing a general description of our method, illustrate the method by giving an example of an actual implementation, and finally give an outlook on possible extensions.

2. Supporting the Learner's Learning Management

Managed learning, or here, more specifically, goal-directed learning can be structured in three distinct phases. (1) stating the goals, (2) checking the goals, and (3) adapting the goals. In "normal" learning, when meta-cognitive aspects are not introduced to the learners, more often than not, each of these phases is neglected. Goals are not made explicit (What do I want to learn?); the achievement of a goal is not controlled (Did I successfully fulfil a goal?); and if one fails to reach a goal, no adequate actions are taken (I am not progressing as fast as I planned to, but what the heck!). Everybody has lived through the consequences of such behaviour: lack of time, pressure, and finally, frustration.
Hence, we propose to make these three phases of learning explicit. The general scheme is as follows: The learner specifies her learning goals and the time in which she wants to achieve these goals. As a result, a mean workload arises. Consequently, the system supports the learner in achieving her goals by comparing the performed workload to the planned workload, and providing appropriate feedback.

1. Setting the Goals

More concretely, the planned mean workload w_p is defined as the ratio of the total number of goals g the learner wants to achieve and time t she has decided to spend on them: $w_p = \frac{g}{t}$. The actual workload w_a is defined as the ratio of the de facto achieved goals g_a and the time spent t_a: $w_a = \frac{g_a}{t_a}$.

* This publication was generated in the MMiSS Project, funded by the German Federal Ministry of Education and Research (BMBF), contract 08NM070E and in the LeActiveMath project, funded under FP6, Cntr. 507826. The authors are solely responsible for its content.

Now, the system can easily determine whether the learner is still on track by comparing the planned workload to the achieved workload. If $w_a \geq w_p$, then learning progress is adequate to reach the learning goals. Otherwise, if $w_a < w_p$, then the learner is not advancing fast enough. In both cases, the system should provide appropriate feedback. We will discuss the positive case first.

2. Providing Positive Feedback

If the learner shows better progress than expected, appropriate feedback is at least as important as in the opposite case. Deserved appraisal significantly increases the motivation to learn [5]. Here, motivating feedback can consist of telling the learner how much faster he reaches his goals if he keeps up his current learning speed. This new amount of necessary time t_n is calculated as the ratio of the total numbers of goals g and her actual workload w_a: $t_n = \frac{g}{w_a}$.

Thus, the system can provide the following feedback:

You are making very good progress. Instead of 10 days, you will have finished in 5 days!

3. Providing Supporting Feedback

In the opposite case, in the event of inadequate progress, a system should provide more than a negative statement. Constructive feedback distinguishes itself by pointing out alternatives or suggestion for improvement. Here, valuable feedback can consist of proposing a new workload that allows the learner to still reach his learning goals in the chosen time. The new workload w_n is calculated as the ratio of the concepts that are still to be learned and the time left: $w_n = \frac{(g - g_a)}{(t - t_a)}$.

As a result, the system can present the following feedback to the learner:

You are advancing slower than planned. But you can still reach your goals if you study 4 concepts a day!

4. Adapting a Plan

Especially if a learner was only recently introduced to meta-cognitive aspects and just starts to explicitly set her goals, it happens that a plan was overly optimistic, and, even with an increased workload, can not be achieved. Likewise, new and unforeseen events can arise and lead to inadequate plans. In these cases, the learner has to adapt her plan. Here, adaptation means that the learner assigns a new value to the variable time t. In general, the content of a course is set by a teacher, hence it proves less likely to be adaptable than time on which the learner has a higher influence, by spending additional hours to the topic, or by working on week-ends. Consequently, the new workload is calculated as the ratio of the still to be achieved goals and the updated time t_u: $w_n = \frac{(g - g_a)}{t_u}$.

5. Used Data

We attached great importance to the general applicability of the presented method. Most learning management systems represent the information used in the above calculations. The learning materials the student has read and the time she spent on studying them are minimum requirements of a learning management system and represented in specifications such as SCORM [9] or standards for system architectures such as the upcoming LTSA [4].

Moreover, if the learning management system is compliant to the Learning Object Meta-Data (LOM) Scheme [3], the meta-data difficulty and typicallearningtime can be used to improve the precision of the planned workload w_p. For instance, difficulty can serve as a weight of different goals. E.g., 0.5 for very easy, 0.8 for easy, 1 for normal, 1.5 for difficult, and 2 for very difficult. If g_{ve}, g_e, g_n, g_h, g_{vh} denotes the number of goals with a difficulty level of “very easy, easy, normal, hard, very hard”, then the new workload w'_p is defined as: $w'_p = \frac{(g_{ve} \times 0.5 + g_e \times 0.8 + g_n \times 1 + g_h \times 1.5 + g_{vh} \times 2)}{t}$. A simplified example illustrates the improvement: Suppose the total number of the goals g is 4, in which there are 2 very easy ones and 2 very hard ones, hence $g_{ve} = 2$, $g_e = 0$, $g_n = 0$, $g_h = 0$, $g_{vh} = 2$. The student plans to spend $t = 2$ days on these goals. Using the old formula, the planned mean workload will be: $w_p = \frac{g}{t} = \frac{4}{2} = 2$. Using the new formula results in:

$$w'_p = \frac{(g_{ve} \times 0.5 + g_e \times 0.8 + g_n \times 1 + g_h \times 1.5 + g_{vh} \times 2)}{t} = \frac{(2 \times 0.5 + 0 \times 0.8 + 0 \times 1 + 0 \times 1.5 + 2 \times 2)}{2} = 2.5 .$$

If the student follows $w_p = 2$ without difficulty differentiation of the goals, she could study for the two very easy goals at the first day, and the two very difficult ones at the second day. But she will lack time at the second day since those two goals are much more difficult than the other two. If she follows $w'_p = 2.5$, she could also study for the two very easy goals at the first day. However, since these two goals do not correspond to half of the work, she could also study for part of the difficult goals at the first day. Then, she should have sufficient time to finish the two difficult goals at the second day.
The following section provides a concrete example of an implementation in an existing learning environment.

3. Exemplary Integration in ACTIVEMATH

ACTIVEMATH [6] is a web-based learning environment that dynamically generates interactive courses adapted to the student's goals, preferences, capabilities, and prior knowledge. The content is represented in a reusable XML-knowledge representation specifically designed for an educational context. ACTIVEMATH supports individualized learning material in a user-adaptive environment, active and exploratory learning by using (mathematics) service tools and with feedback, better reusability and interoperability of the encoded content and exercises. For different purposes and for different users the learning material and its presentation can be adapted: the selection of the content, its organization, the means for supporting the user have to be different for a novice and an expert user, for an engineer and a mathematician, for different learning situations such as a quick review and a full study. Since there is no chance to know the goals, the profile, and the preferences of any user when designing the system, ACTIVEMATH relies on adaptive course generation.

Learning Management in ACTIVEMATH

For the first implementation in ACTIVEMATH, we decided for a straightforward approach. Since a learner can either read a fixed, non-adapted curriculum written by an author or let the system generate an individualized curriculum, the number of goals is given by the number of pages of a curriculum. The learner can then set the time in which she plans to finish the curriculum by specifying a number of days. A goal is defined to be achieved, if the page it appears on is read (a very crude estimation, certainly, but see below for suggestions on how to improve it). Figure 1 and 2 contain screen-shots of both adequate (Figure 1) and insufficient (Figure 2) learning progress.
The kind of supportive feedback the system gives can be deepened easily according to the available information. For instance, ACTIVEMATH contains a very elaborate student model that stores explicit information about what learning materials the learner has studied and his mastery of these learning materials. The mastery is represented as a triple of Knowledge, Comprehension, and Application, values that represent a subset of Bloom's taxonomy of educational objectives [1]. In the next implementation, the method will use these values to get a more accurate estimation regarding the achievement of goals.
A more substantial extension of the method is in the spirit of interactive learner modelling [8]. During such an interaction (dialog), the student and system have a discourse on the goals and their achievements. For instance, the system presents its view on the achievement of the goals. Then, if the learner disagrees the system's conclusion, she will be asked to prove her opinion. The system will try to ask her several questions about the concepts and some related knowledge, for instance, by presenting exercise for the learner to solve. At the end, either the system will convince the learner or the learner will convince the system.

4. Tests and Conclusion

Recently, we have performed a study in order to get information about the students' opinion of this method (e.g., whether they believe it can serve as a valuable learning tool), and, on the longer run, whether and to what extend it indeed teaches what it was designed for, namely, meta-cognitive learning management skill. During the study, the students gave very positive feedback and some suggestions. They think, having the possibility to plan for their studies is very useful, and following the planned workload w_p given by the system did help them to organize their studies. Furthermore, the students also gave some suggestions for improving the method, e.g.:

"If it's going to be over a long period of time, we (i.e. the students) should be able to configure how often (e.g. every week) the system reminds the user how far she has come/should have come."

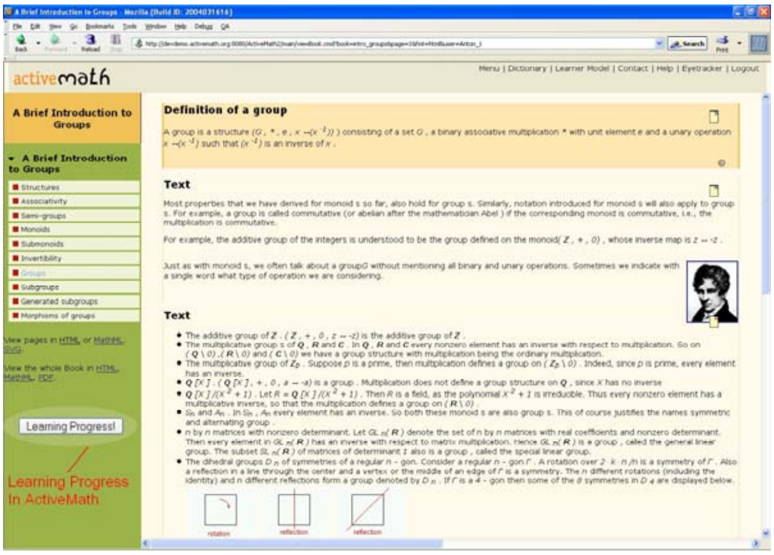

Illustration 1: Learning Progress in ActiveMath

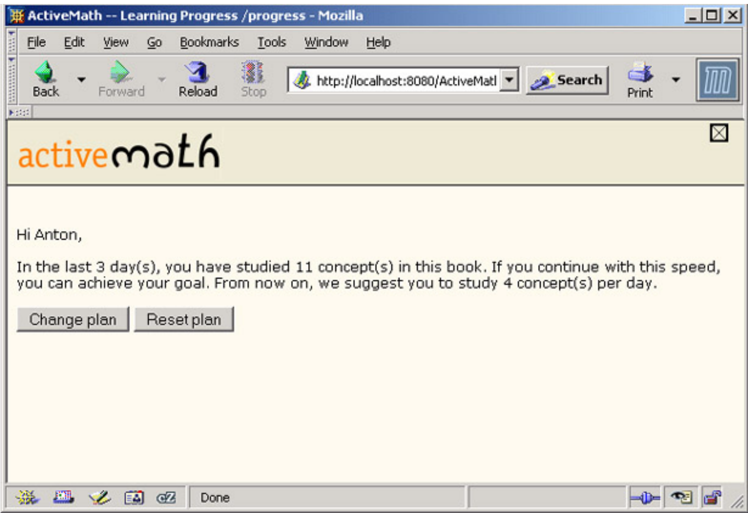

Illustration 1: System feedback in case of sufficient progress

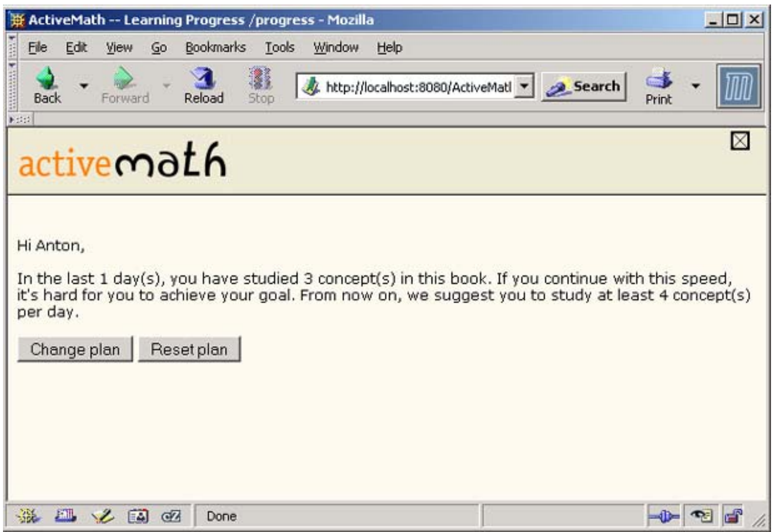

Illustration 2: System feedback in case of insufficient progress

To summarize, supporting and stimulating meta-cognitive learning skills is a necessary requirement for successful teaching. Fortunately, it does not require heavy machinery, but first steps can be implemented and integrated easily in existing learning management system. Allowing a learner to define her learning goals, offering the possibility to check and, if necessary, adapt her goals, can be realized using data available in virtually every system by following the method we described in this paper.

References

[1] B.S. Bloom, editor. *Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain*. Longmans, Green, New York, Toronto, 1956.

[2] C. Conati and K. VanLehn. Teaching meta-cognitive skills: Implementation and evaluation of a tutoring system to guide self-explanation while learning from examples. In S.P. Lajoie and M. Vivet, editors, *Artificial Intelligence in Education*, pages 297–304. IOS Press, 1999.

[3] IMS Global Learning Consortium, Inc. IMS Learning Resource Meta-Data Information Model (LOM) Version 1.2.1 Final Specification, September 2001.

[4] Learning Technology Standards Committee of the IEEE Computer Society. IEEE P1484.1/D9, 2001-11-30 Draft Standard for Learning Technology – Learning Technology Systems Architecture (LTSA), November 2001.

[5] A. F. Lucas. Using psychological models to understand student motivation. In M. D. Svinicki, editor, *The Changing Face of College Teaching. New Directions for Teaching and Learning*, number 42. Jossey-Bass, San Francisco, 1990.

[6] E. Melis, E. Andrès, J. Büdenbender, A. Frischauf, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. ActiveMath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education*, 12(4):385–407, 2001.

[7] A. H. Schoenfeld. What's all the fuss about metacognition? In A. H. Schoenfeld, editor, *Cognitive science and mathematics education*, pages 189–215. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[8] J. A. Self. Bypassing the intractable problem of student modelling. In C. Frasson and G. Gauthier, editors, *Intelligent-tutoring systems: At the crossroad of artificial intelligence and education*, Norwood, New Jersey, 1990. Ablex Publishing Corporation.

[9] ADL Technical Team. Sharable Content Object Reference Model (SCORM) Version 1.2, October 2001.

[10] B.Y. White and T.A. Shimoda. Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education*, 10:151–182, 1999.