Title

Automated Proof Planning for Instructional Design

Permalink

https://escholarship.org/uc/item/4sj2f4d1

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 23(23)

ISSN

1069-7977

Authors

Melis, Erica
Glasmacher, Christoph
Ullrich, Carsten
et al.

Publication Date

2001

Peer reviewed

Automated Proof Planning for Instructional Design

Erica Melis (melis@dfki.de)
Christoph Glasmacher (christoph.glasmacher@cops.uni-sb.de)
Carsten Ullrich (cullrich@ags.uni-sb.de)
Peter Gerjets (pgerjet@gwdg.de)
DFKI Saarbrücken/Department of Psychology; Saarland University
D - 66041 Saarbrücken, Germany

Abstract

Automated theorem proving based on proof planning is a new and promising paradigm in the field of automated deduction. The idea is to use methods and heuristics as they are used by human mathematicians and encode this knowledge into so-called `methods`. Naturally, the question arises whether these `methods` can be beneficially used in *learning* mathematics too. This paper investigates and compares the effect of different instruction materials (textbook-based, example-based, and method-based) on problem solving performance. The results indicate that the performance for the method-based instruction derived from automated proof planning in the ΩMEGA system is superior to that of the other instructions that were derived from a textbook and an example-based classroom lesson. These results provide a first support for introducing proof planning based on methodological knowledge into the school curriculum for mathematics.

Introduction

Recent developments in automated deduction, one of the areas of Artificial Intelligence (AI), have shown the advantage of employing methods and heuristics used by human mathematicians. Naturally, the question is whether they can be beneficially used in teaching mathematics, for instance in interactive e-courses such as ACTIVEMATH (Melis et al., 2001).

The goal of the research reported in this paper has been to gather empirical evidence for the hypothesis that the knowledge we made explicit in proof planning `methods` for a restricted area of mathematics, namely limit problems, is indeed useful for learning to prove theorems in this area. A positive answer in this and other areas of mathematics can serve as a basis for the long-term goal to acquire `methods` to solve mathematical problems and then to use them to gradually change the teaching of mathematics.

To understand the interdisciplinary context, we will have a quick look at automated theorem proving.

Automated and Human Theorem Proving Traditional automated theorem proving systems such as OTTER have attained a remarkable strength in deductive search. They are, however, weak when it comes to nontrivial mathematical theorems where long range planning or other global search control is needed. Moreover, long proofs generated by these systems are almost incomprehensible. Therefore, techniques like proof planning that more closely follow the reasoning patterns observed in humans became more prominent.

The goal of automated proof planning (Bundy, 1988; Melis & Siekmann, 1999) is to identify and to employ human-like strategies and methods for theorem proving in order to avoid the almost exhaustive search in super-exponential search spaces that makes traditional automated theorem proving infeasible for most nontrivial mathematical conjectures. We investigated reports and mathematical textbooks (Melis, 1994) to make such strategies and methods explicit and then available for the ΩMEGA proof planner. Essentially, these methods are (generalized) macro-steps. This is in accordance with Koedinger and Anderson (1990) who investigated human theorem proving in geometry and found that humans employ macro-steps when proving theorems.

The identification and design of `methods` and control knowledge is very laborious as this kind of knowledge is not explicit in mathematical texts. However, some progress has now been made in the identification of mathematical methods and control knowledge (Melis, 1998). Based on these achievements we focus on questions such as

Is the knowledge that was made explicit for automated proof planning useful for supporting human learning of mathematical problem solving?

We are inclined to say *yes*. One reason is the explicit availability of this knowledge that can be used for proof presentation. An automated proof planner produces proof plans which in turn can be *presented in a more comprehensible way*. We investigated how proof presentation for teaching and learning can be generated from proof plans, see Melis and Leron (1999). Moreover, we investigated how such a presentation of proof plans can meet pedagogically and cognitively motivated requirements for presenting mathematical problem solutions and proofs, in particular the requirement for a hierarchically structured presentation originating from empirical results in Leron (1983) and Catrambone (1994).

A second reason is that this knowledge is needed for problem solving but not always present in textbooks (VanLehn, Jones, & Chi, 1992). Indeed, interviews with teachers of mathematics indicate a need for teaching methodological knowledge as captured in `methods`. Some even claim this is the essence of good teaching and a source of improved learning and thus de-mystifying mathematics to some extent. As opposed to merely checking the correctness of single proof steps as in learning with traditional mathematical instruction, learning of *methods* should help in understanding the discovery of a proof. This leads to an improved performance based on understanding. The methodological knowledge includes the systematic construction of mathematical objects which is needed in many proofs.

The idea of making an expert's tacit problem solving knowledge explicit to learners is in accordance with some well known approaches in instructional psychology such as cognitive apprenticeship (Collins, Brown, & Newman, 1989) or the provision of instructional explanations (Chi, 1996).

Certainly, the success largely depends on the actual proof planning *methods* made explicit and encoded and therefore another direction of research, see Melis and Pollet (2000), aims at describing `methods` for interactive proof planning most appropriately. In addition to the evaluation of the concrete `methods` there is the more general question on whether the explicit teaching of relatively abstract methods helps in learning mathematics.

Although there are reasons to believe in instructional benefits, empirical evidence is required to substantiate the *yes*, and this is the focus of this report.

In this paper we present first empirical results. To begin with, proof planning is briefly reviewed, in particular proof planning of limit theorems which is the object of the described experiment.

Proof Planning Basics

Proof planning is based on classic AI-planning (Fikes & Nilsson, 1971) which reduces a goal to subgoals by introducing operators until all open subgoals match one of the initial state descriptions. When the sequence of operators is applied (in forward direction), the initial state is transformed into a state in which the goals hold. In proof planning, the goal is the theorem to be proved and the initial state consists of the proof assumptions.

For instance, for proving the theorem LIM+ which states that the limit of the sum of two real-valued functions f and g for a real number a is the sum of their limits L_1 and L_2, the conjecture to be proven is

$$\lim_{x \quad a}(f(x) \quad g(x)) = L_1 \quad L_2$$

and the proof assumptions are

$$\lim_{x \quad a} f(x) = L_1 \text{ and } \lim_{x \quad a} g(x) = L_2.$$

A proof plan is a sequence of operators whose application realizes an inference from the proof assumptions to the theorem. In proof planning, the operators are called `methods`. They are frequently designed in a way corresponding to typical mathematical techniques such as `proof by induction`, `proof by refutation`, and `proof by diagonalization`, to quote some of the best-known `methods`. There are, however, less well-known `methods` which do not have a distinct name in mathematics. For instance, certain `estimation methods` for inequalities are typically not explicitly mentioned although they encode a frequently used trick. One of these `estimation methods` (`ComplexEstimate`) and another `method` (`TellCS`). These have been used in our experiments and are explained below.

Those non-name `methods` are often used only implicitly in course materials. This implicit treatment of proof `methods` is one reason why textbooks do not provide enough explanation of *how to find* a proof.

Proof Planning in the Limit Domain

In this section we describe a class of theorems, the way their proofs can be discovered mathematically, and the way proof planning in the ΩMEGA system implements this with `methods`.

The Theorems Limit theorems are taught at German high schools. Limit theorems claim something about the limit $\lim_{x \quad a} f(x)$ for a function f or about the continuity of a function f.[1]

The definition of $\lim_{x \quad a} f(x) = l$ describes formally that if x converges to a, then $f(x)$ converges to the limit l. The convergence $x \quad a$ means that the distance $x - a$ of x and a becomes arbitrary small. The definition of the limit describes that if $x - a$ becomes arbitrary small, then $f(x) - l$ becomes arbitrary small too. Put formally, for every arbitrary small real number ε exists a real number δ such that if $x - a < \delta$,[2] then $f(x) - l < \varepsilon$.[3]

Example Take the linear function $f(x) = 2 \quad x \quad 3$. When x converges to 0, then $f(x)$ converges to 3, i.e., for any arbitrary small ε, there is always a δ-environment $U_\delta(0)$ of $a = 0$ such that for any x in that environment $f(x)$ is in the ε-environment.

Counter Example Take as a counter example the function

$$f(x) = \begin{array}{rcl} 2 & : & x > 0 \\ -2 & : & x < 0 \end{array}$$

in Figure 1 which does not converge at point $x = 0$.

If ε is smaller than 2, there is always an x close to 0 for which $f(x)$ is not in the ε-environment of $l = \quad 2$ or of $l = -2$.

The Proofs The proofs of limit theorems have to suggest a δ, in relation to the given ε, such that the limit inequalities, e.g. $f(x) - l < \varepsilon$, hold. That is, a relation between ε and δ has to be determined such that for each x from the δ-environment of a the value $f(x)$ is in the ε-environment of l. Therefore, the standard proofs of these theorems are often called ε–δ–proofs.

Typically, textbooks postulate an appropriate relation between ε and δ out of the blue. Then they show

[1] or about the limit of a sequence which is a special case of a function.

[2] i.e., x is in the δ-environment $U_\delta(a)$ of a

[3] i.e., $f(x)$ is in the ε-environment $U_\varepsilon(l)$ of l

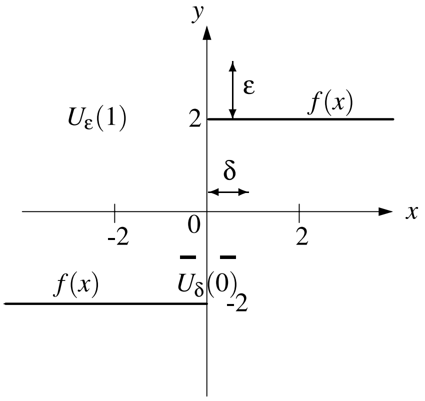


Figure 1: A function that does not converge at point $x = 0$

that the stipulated δ which is dependent on ε make the (inequality)-conjectures true. In contrast, proof *discovery* reveals the relation either by intuition or by systematically detecting conditions/constraints under which $f(x)-l$ becomes arbitrary small given that $x-a$ becomes arbitrary small. Those constraints result from analyzing the inequalities to be proven. This analysis often includes an abduction of new simpler inequalities/constraints sufficient to not invalidate the original ones.

These constraints may restrict the relation between ε and δ. For instance, if the constraints are $0 < \delta$ and $\delta < \varepsilon$, then $\delta = 2\ \ \varepsilon$ would be an invalid relation but $\delta = \frac{\varepsilon}{2}$ would be a valid one. When all possible constraints have been collected, then it is more transparent how to choose the relation between ε and δ. For instance, if the collected constraints are $0 < \delta$ and $\delta < \varepsilon$, then it is easy to see that the relation $\delta = \frac{\varepsilon}{2}$ satisfies the constraints. In particular, for complicated problems the systematicity is indispensable because ad hoc guesses and trial and error do not help much.

Proof Planning Proof planning for ε–δ–proofs (in a backward fashion) introduces a sequence of `methods` transforming $x-a\ \ < \delta$ to $f(x)-a\ \ < \varepsilon$:

$$\begin{array}{rrcl} & f(x)-l & < & \varepsilon \\ = & \dots & < & \varepsilon \\ = & x-a & < & \delta. \end{array}$$

Each of the methods may yield restrictions on the relation of ε and δ. Therefore, proof planning systematically restricts the relation of ε and δ by uncovering constraints sufficient for making the inequalities true which are required in the theorem.

If a subgoal is a primitive inequality such as $0 < 1$ or $\delta < \varepsilon$, then `TellCS`[4] just collects it as a new constraint. If the constraints are not as immediate/primitive, then they can only be shown via a reduction to less complicated, primitive inequalities. For instance, to show $x^2 - a^2 < \varepsilon$ one might reduce the goal to the subgoals $x \quad a < r$ and $x - a < \frac{\varepsilon}{r}$ for a number r to be determined and then conclude $x^2 - a^2 = (x \quad a)\ \ (x - a) < r\ \ \frac{\varepsilon}{r} = \varepsilon$ and therefore $x^2 - a^2 < \varepsilon$. In proof planning such reductions are realized by estimation methods. One of those methods is `ComplexEstimate` whose simplified version is used in one of the instruction materials and described below.

Simplified ComplexEstimate The simplified `ComplexEstimate method` delivers the first reduction step in the following plan.

$$\begin{array}{rrcl} & f(x)-l & < & \varepsilon \\ = & k \quad x-a & < & \varepsilon \\ = & x-a & < & \frac{\varepsilon}{k} \\ = & x-a & < & \delta \end{array}$$

It rewrites $f(x)-l$ to $k \quad x-a$, determines the k which can be a number but also, in more complicated cases, a term like $x \quad 1$ (see the Binomial computation above), and conjectures the subgoal that k has an upper bound (a real number r). The latter subgoal $k\ \ < r$ is a constraint and gives rise to establishing the relation $\delta = \frac{\varepsilon}{r}$ in order to guarantee $\delta < \frac{\varepsilon}{k}$ which implies the last proof step.

`ComplexEstimate`'s general procedure to determine k is polynomial division but manual computation may use simpler procedures in simpler cases, e.g. a Binomial formula.

This general `ComplexEstimate` (not used in the instruction materials) reduces an inequality goal to three subgoals (rather than two in the simplified version) by means of decomposing a term t into a linear combination $t = k \quad a \quad m$ for which an estimation of a is already known. It justifies the original goal by the three subgoals and the Triangle Inequality.[5] For difficult decompositions the `method` can call a polynomial division function without any problems.

The general `ComplexEstimate` as used in the automatic proof planner ΩMEGA covers the simpler cases for $k = 1$ and $m = 0$. Its generality allows for proving pretty complicated theorems that are beyond the range of our experiments. All test problems in the experiment require the special case $m = 0$ only. In the first, second, third, fourth, and fifth test problem, k is a real number, whereas in the sixth test problem k is the term $(x-1)$.

Hypotheses

The overall goal of the study presented in this paper is an empirical validation of the assumption that the instructional presentation based on `methods` leads to an improved problem solving performance in mathematics. This differs from typical textbooks or classroom lessons where the methodological knowledge is currently not explicitly used.

The first hypothesis states that instructional material that includes information about proof-generation methods improves the overall problem solving performance.

[4] for ”Tell the Constraint Solver”.

[5] $A \quad B < A \quad B$

The second hypothesis postulates that the method-based instruction is especially helpful in solving far-transfer test problems that presuppose the generation of new solution paths.

To test the first hypothesis instructional material based on ΩMEGA's proof plan methods was designed. The method-based instructions were contrasted with conventional instruction materials: textbook-based instruction and example-based instruction.

To test the second hypothesis test problems of different transfer distance were used[6] .

Experiment

Method

Participants The subjects were 38 students of Saarland University, Germany who either participated for course credit or payment. Average age was 24.1 years.

Materials and procedure Each student was provided with the following material in a booklet: (1) An introduction that described the nature and purpose of limits. Additionally, the introduction presented a definition of the notion of an *environment* as a prerequisite for the formal definition of *limit*. (2) A formal definition of the notion *limit* together with an illustrating graph. (3) One worked-out example that illustrated how the limit $\lim_{x \quad a} f(x)$ for a given function f and a given value a can be proven. Depending on the experimental conditions different solution approaches were selected in the worked-out examples.

Subjects were advised to study the instructional material carefully. After reading the booklet subjects had to solve six test problems that differed in their transfer distance with respect to the instructional example. The six text problems were of increasing difficulty and decreasing structural similarity to the example explained in the instruction.

Design and dependent measures Four different instructional materials were designed as independent variables: Textbook-based instruction, example-based instruction, and two types of method-based instruction (only differing in the sequence of the parts of instructional materials). The instructional conditions differed only with respect to the solution approach for the worked-out example and with respect to the sequence of the instructional materials.

In the textbook-based instruction the introductory page was immediately followed by a short formal definition of the notion *limit* and an illustrating graph. Subsequently, one example of a worked-out ε-δ-proof for a linear function ($f(x) = x \quad 2$, with x being undefined at $x = 1$) was presented. The example solution was taken from an university-level textbook. The textbook-based instruction merely postulated the pivotal relation between ε and δ without derivation from more general principles. The mere stipulation of pivotal assumptions is a frequent feature of example proofs in textbooks.

The example-based instruction differed from the textbook-instruction in that the example problem was presented immediately after the introductory page. To establish a general relation between ε and δ, suitable values for δ are introduced for several concrete ε values of decreasing size. This approach allowed for an inductive derivation of a general relation between these two parameters. Additionally, the example-based instruction differed from the textbook-instruction in the sequence of the instructional materials: The example proof was presented before the formal definition of the notion *limit* and the respective illustrating graph were introduced.

The method-based instruction took the methods simplified `ComplexEstimate` and `TellCS` from ΩMEGA's proof planner and described an example solution explicitly using `ComplexEstimate` and `TellCS` (the collection of constraints). It shows how `ComplexEstimate` reduces a complicated estimation to several simpler ones. As a general approach it also employs the collected constraints for defining a relation between ε and δ. The methods are applied to prove the example problem and an abstract description of the method is provided.

Two versions of this method-based instruction were designed that differ with respect to the sequence of instructional materials. In version A the definition of the notion *limit* was followed by an abstract description of `ComplexEstimate` and an illustrating example applying this method. In version B the worked-out example was presented before the notion *limit* was defined and the `ComplexEstimate` method was described in a more abstract way.

As dependent variables problem-solving time and problem solving performance for the six test problems were registered. The test problems differed in transfer distance. The first two test problems were isomorphic to the example used in the instructional material (proving a limit for a linear function of the form $f(x) = x \quad b$). The next three test problems were near-transfer problems (proving a limit for a linear function of the form $f(x) = ax \quad b$). Finally, a far-transfer test problem had to be solved (proving a limit for a quadratic function of the form $f(x) = ax^2 \quad bx \quad c$). After the experiment, data were collected by means of a questionnaire, in particular, the subjects' last maths grade in school, the subjects' interest in mathematics, sociodemographic data, and whether they were taught anything about limit theorems in (past) school lessons.

Results

The six test-problem solutions were scored as follows. For a totally correct answer a score of 1 for isomorphic problems, a score of 2 for near-transfer problems, and a score of 4 for far-transfer problems was assigned. Hence, the maximum total score is 12. 50% of the full score were assigned to a solution, if the answer was correct except for minor, nonconceptual mistakes (e.g. numerical calculation errors, mixing up δ and ε in the solution

[6] Transfer distance is a measure for structural similarity.

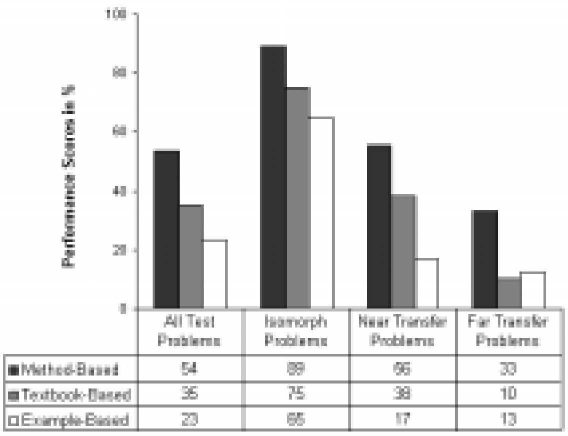


Figure 2: Mean performance scores (in percentage of possible maximum score) as a function of instructional condition and transfer distance between test problems and example problems

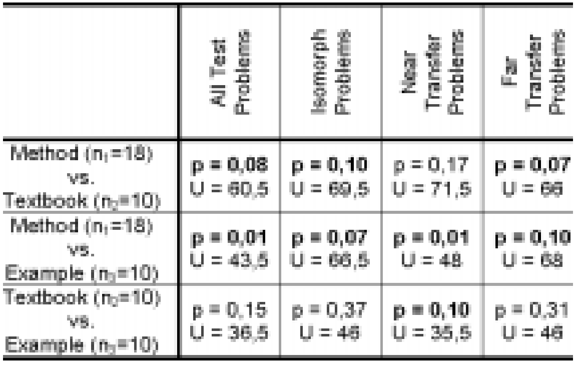
equation). 75% of the full score were assigned in case of incorrect solution of the polynomial in the last test problem.

Nonparametric tests were used in all performance analysis because of distorted distributions. In a first step, we compared the two method-based instructions with respect to performance differences. Mann-Whitney U-tests revealed that there were no differences in the total problem-solving score (U(9, 9) = 36; p(two-tailed) = .69) or in problem-solving time (U(9, 9) = 39; p(two-tailed) = .89). Thus, both method-based instructions were collapsed for further analysis. An overall comparison of the method-based, the example-based and the textbook-based instruction with Kruskal-Wallis H-test revealed that there were significant differences in the total problem-solving score ($\chi^2(2, N = 38) = 5.87$; p = .05) but not in problem-solving time ($\chi^2(2, N = 38) = 2.45$; p = .29). The instructional conditions did not differ with respect to the last math grade in school, domain-specific knowledge they were taught in school, interest in mathematics, sex, and age. Figure 2 provides the mean performance scores (in percentage of possible maximum score) for all three instructional conditions and all levels of transfer distance.

Paired one-tailed comparisons with Mann-Whitney U-tests (see Table 1) yielded that the method-based instruction outperformed the textbook-based instruction (marginally) as well as the example-based instruction with respect to the total problem-solving score. The textbook-based instruction and the example-based instruction did not differ in total problem-solving score.

A more detailed analysis revealed that the method-based instruction and the textbook-based instruction differed marginally with respect to isomorphic problems and to far-transfer problems but not with respect to near-transfer problems. The method-based instruction and the example-based instruction differed with respect to all performance measures, at least marginally. The textbook-based instruction and the example-based instruction did not differ with respect to isomorphic problems and far-transfer problems. However, there was a marginal significant difference with respect to near-transfer problems.

Table 1: Comparison between all instructional conditions with respect to all levels of transfer distance (one-tailed Mann-Whitney U-tests)



	All Test Problems	Isomorph Problems	Near Transfer Problems	Far Transfer Problems

Method (n_1=18)
vs.
Textbook (n_2=10)
	p = 0,08
	U = 60,5
		p = 0,10
		U = 69,5
			p = 0,17
			U = 71,5
				p = 0,07
				U = 66

Method (n_1=18) vs. Example (n_3=10)	**p = 0,01** U = 43,5	**p = 0,07** U = 66,5	**p = 0,01** U = 48	**p = 0,10** U = 68

Textbook (n_2=10)
vs.
Example (n_3=10)
	p = 0,15
	U = 36,5
		p = 0,37
		U = 46
			p = 0,10
			U = 35,5
				p = 0,31
				U = 46

Discussion

As postulated in our first hypothesis the method-based instructional material based on ΩMEGA's proof plan presentation has a significant beneficial effect on learners' subsequent problem-solving performance. Compared to more conventional instructional formats usually found in textbooks and highschool lessons the method-based instruction improves learners' problem-solving performance without requiring more time to be invested.

Contrary to the expectation expressed in our second hypothesis, the performance improvements due to the method-based instructional format are not larger for far-transfer test problems than for isomorphic and near-transfer test problems. An explanation for this unexpected result might be that the far-transfer test problem has been chosen as a too-far one that requires an additional computation (polynomial division) the subjects might have been not capable to carry out or did not even try.

To conclude, the results indicate that the method-based instruction that originated from proof planning *methods* implemented in ΩMEGA is superior to the two other instructions in terms of subsequent problem solving performance. These results provide first evidence that proof planning based on mathematical knowledge may also be used and introduced into highschool curricula for mathematics.

Conclusion

Is the methodological knowledge used in proof planning useful for human learning of maths problem solving?
The results of our experiments indicate that the method-based instruction that originated from automated proof planning is, indeed, superior to the two other instructions in terms of subsequent performance. These results provide first support for introducing proof planning based on

methodological knowledge into the highschool curricula for mathematics.

It is not necessary to restrict this methodological knowledge to `methods` which have been acquired for and used in automated proof planning. We can, however, re-use these results. Then the advantage is that those `methods` are formalized and implemented and, therefore, can be employed by a system supporting *interactive* problem solving.

The presented empirical results are limited, however, to only one area of highschool mathematics. Future work will try to provide similar evidence in other areas as well.

Interestingly, we met many committed mathematics teachers in Germany who have been engaged in activities targeting a similar idea without knowing, of course, about automated theorem proving and proof planning. Their concern is a reshaping of mathematics lessons that aims at learning problem solving methods, heuristics, and structuring problems and solutions rather than at memorizing facts and procedures.

Future Work

In the future we will replicate the experiment reported here with several augmentations. First, we will obtain think-aloud protocols to get more detailed insights into the learning and problem-solving processes elicited by different instructional materials. Second, we will try to shed more light on the results with respect to second hypothesis by adjusting the difficulty of the far transfer test problems. Third, we will additionally consider certain features of instructional situations like domain-specific prior knowledge or degree of time pressure that have been shown to influence the profitability of different instructional materials (Gerjets, Scheiter, & Tack, 2000).

Another line of research will pertain to the fact that the provision of profitable instructional materials does not ensure that learners indeed use these materials appropriately. This is especially true for computer-based learning environments that allow learners to control for many aspects of the learning process, e.g. the selection of instructional materials (Gerjets, Scheiter, & Tack, 2001). Therefore, we will examine whether learners select method-based instructional materials when they are allowed to choose between different types of information in electronic learning environments. Finally, we will design experiments investigating the influence of explicitely teaching control knowledge (i.e. knowledge on when to choose which method) in addition to the teaching of method knowldege.

References

Bundy, A. (1988). The use of explicit plans to guide inductive proofs. In E. Lusk & R. Overbeek (Eds.), *Proc. 9th international conference on automated deduction (cade-9)* (Vol. 310, p. 111-120). Argonne: Springer.

Catrambone, R. (1994). Improving examples to improve transfer to novel problems. *Memory & Cognition*, *22*(5), 606-615.

Chi, M. (1996). Constructing self-explanations and scaffolded explanations in tutoring. *Applied Cognitive Psychology*, *10*, 33-49.

Collins, A., Brown, J., & Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of robert glaser* (p. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates.

Fikes, R., & Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, *2*, 189-208.

Gerjets, P., Scheiter, K., & Tack, W. (2000). Resource-adaptive selection of strategies in learning from worked-out examples. In L. Gleitman & A. Joshi (Eds.), *Proc. of the annual conference of the cognitive science society* (p. 166-171). Mahwah, NJ: Erlbaum.

Gerjets, P., Scheiter, K., & Tack, W. (2001). *Problems of example selection and example processing in hypertext-based learning environments* (Tech. Rep.). Saarbr"ucken, Germany: Saarland University.

Koedinger, K., & Anderson, J. (1990). Abstract planning and perceptual chunks: Elements of expertise in geometry. *Cognitive Science*, *14*, 511-550.

Leron, U. (1983). Structuring mathematical proofs. *The American Mathematical Monthly*, *90*, 174-185.

Melis, E. (1994). How mathematicians prove theorems. In *Proc. of the annual conference of the cognitive science society* (p. 624-628). Atlanta, Georgia U.S.A.

Melis, E. (1998). The "limit" domain. In R. Simmons, M. Veloso, & S. Smith (Eds.), *Proc. of the fourth international conference on artificial intelligence in planning systems* (p. 199-206).

Melis, E., Andres, E., Goguadse, G., Libbrecht, P., Pollet, M., & Ullrich, C. (2001). Activemath: System description. In *Proc. of the international conference on artificial intelligence in education.*

Melis, E., & Leron, U. (1999). A proof presentation suitable for teaching proofs. In S. Lajoie & M. Vivet (Eds.), *international conference on artificial intelligence in education* (p. 483-490). Le Mans: IOS Press.

Melis, E., & Pollet, M. (2000). Domain knowledge for search heuristics in proof planning. In *Aips 2000 workshop: Analyzing and exploiting domain knowledge* (p. 12-15).

Melis, E., & Siekmann, J. (1999). Knowledge-based proof planning. *Artificial Intelligence*, *115*(1), 65-105.

VanLehn, K., Jones, R., & Chi, M. (1992). A model of self-explanation effect. *Journal of learning science*, *2*(1), 1-59.