Teaching meta-cognitive skills: implementation and evaluation of a tutoring system to guide self-explanation while learning from examples.

Cristina Conati[1] and Kurt VanLehn[1, 2*]

[1] *Intelligent Systems Program, University of Pittsburgh, U.S.A.*
[2] *Department of Computer Science, University of Pittsburgh, U.S.A.*

The SE-Coach is a tutoring module designed to help students learn effectively from examples through guiding self-explanation, a meta-cognitive-skill that involves clarifying and explaining to oneself the worked out solution for a problem. The SE-Coach provides this guidance through (a) an interface that allows the student to interactively build self-explanations based on the domain-theory (b) a student model that assesses the quality of the student's explanations and the student's understanding of the example. The SE-Coach uses the assessment in the student model to elicit further self-explanation to improve example understanding.

In the paper we describe how the SE-Coach evolved from its original design to the current implementation via an extensive and thorough process of iterative design, based on continuous evaluations with real students. We also present the results of the final laboratory experiment that we have performed with 56 college students. We discuss some hypotheses to explain the obtained results, based on the analysis of the data collected during the experiment.

1 Introduction

Computer-based tutors generally focus on teaching domain specific cognitive skills, such as performing subtractions in algebra or finding the forces on a body in Newtonian physics. However, a key factor that influences the quality of learning is what cognitive processes are triggered when the student learns. Tutoring is more effective when it encourages cognitive processes that stimulate learning and discourages counterproductive cognitive processes.

We have developed a tutoring module, the SE-Coach, that instead of teaching directly the knowledge necessary to master a target domain, stimulates and guides the application of self-explanation, a learning process that allows the effective acquisition of knowledge in many domains where it is possible to learn from examples. Self-explanation is the process of generating explanations and justifications to oneself when studying an example. There are many studies showing that students who self-explain learn more[1-3]. When students are either explicitly taught [4] or even just prompted [5] to self-explain, most students will do so and thus increase their learning. The SE-Coach provides tutoring for self-explanation within Andes, a tutoring system designed to teach Newtonian physics to students at the US Naval Academy [6]. Within Andes, the SE-Coach makes sure that students thoroughly self-explain the available examples, especially those parts that may be challenging and novel to them.

A first prototype of the SE-Coach was described in [7]. It included: (a) a Workbench, that interactively presents examples and provides tools to construct theory-based self-

* We thank Prof. Jill Larkin for her invaluable help at the different stages of the system design and evaluation. This research is sponsored by ONR's Cognitive Science Division under grant N00014-96-1-0260.

explanations, (b) a probabilistic student model, that uses both the students' workbench actions and estimates of their prior knowledge to assess the students' understanding of an example, and (c) a Coach, that uses the assessment from the student model to identify deficits in the students' understanding and elicits self-explanations to remedy them.

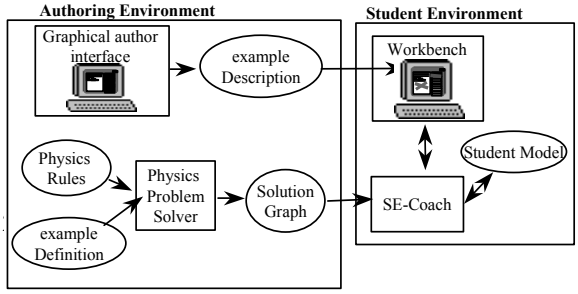
In this paper we describe how the initial prototype evolved into the current implementation through successive evaluations with real students. We focus in particular on the changes to the Workbench and to the Coach. Details on the implementation and performance of the SE-Coach student model can be found in [8]. In Section 2 we outline the features of the self-explanation process that influenced the design of the SE-Coach. In Section 3 we give an overview of the SE-Coach's architecture. In Section 4 and 5 we describe the development of the Workbench and the SE-Coach respectively. In Section 6 we discuss a laboratory experiment that we performed with 56 college students to formally evaluate the effectiveness of the SE-Coach. Although the subjects that used the SE-Coach performed better than the control group, the difference did not reach statistical significance. However, the analysis of the log data files generated during the experiment provides interesting insights on how the students perceived and used the systems. In the last section of the paper we discuss these insights and further changes that could help improve the effectiveness of the tutor.

2 Self-explanation with the SE-Coach

A distinguishing characteristic of the SE-Coach is that it focuses on correct self-explanations. In all the previous studies, even incorrect statements were classified as self-explanations. When human tutors guided self-explanation[4, 5], the experimenters did not give feedback on the self-explanations content or correctness. In all these experiments, students' problem solving improved, leading some researches to argue that it is the self-explanation process per se, and not the correctness of its outcome, that elicits learning [2]. Although we agree that even incorrect and incomplete self-explanations can improve learning, we also believe that correct self-explanation can extend these benefits. Therefore, the SE-Coach is designed to verify the validity of students' explanations, and to provide feedback on their correctness.

A second characteristic of the SE-Coach is that it focuses on two specific kinds of self-explanation: (a) justify a solution step in terms of the instructional domain theory, and (b) relate solution steps to goals and sub-goals in the underlying solution plan. While students generally produce a high percentage of theory-based self-explanations, they tend not to generate goal-related explanations spontaneously [3], although these self-explanations can help acquire highly transferable knowledge [10]. We designed the SE-Coach to target these useful but uncommon self-explanations specifically, thus hoping to further improve the benefits for learning.

Another kind of quite frequent self-explanations involves knowledge outside the instructional domain. Unfortunately, the SE-Coach cannot monitor and guide the generation of these explanations. The system would require a natural language based interface, and a much more complex knowledge base and student model to process and evaluate them. However, even if the SE-Coach cannot explicitly guide self-explanations based on background knowledge, hopefully it does not prevent the students from generating them spontaneously.



3 The SE-Coach architecture

The SE-Coach has a modular architecture, as shown in Figure 1. The left side shows the authoring environment.

Prior to run time, an author creates both the graphical description of the example, and the corresponding coded example definition. A problem solver uses this definition and the set of rules representing Andes' physics knowledge to automatically generate a model of the example solution called the *solution graph.* The solution graph is a dependency network that encodes how physics rules generate intermediate goals and facts in the example solution to derive the example's desired quantities [11].

The right side of the figure shows the run-time student environment. Students use the Workbench to study examples and to generate self-explanations. The Workbench sends the students' explanations to the SE-Coach, which tries to match them with rules in the solution graph and provides immediate feedback on their correctness[7]. The student's workbench actions are also sent to the student model, which uses them to assess the quality of the student's explanations and example understanding [8]. The SE-Coach refers to the student model to make decisions about what further self-explanations to elicit from the student.

4 The Workbench for self-explanation

When the student selects an example to study, the Workbench presents it with all the text and graphics covered with gray boxes, each corresponding to a single "unit" of information. When the student moves the mouse pointer over a box, it disappears, revealing the text or graphics under it. This allows the SE-Coach to track what the student is looking at, and for how long. Whenever the student unmasks a piece of the example, if it contains an idea worthy of explanation the Workbench will append a button labeled "self-explain". Pressing the button gives the student a choice between "*This fact is true because...*" and *"This fact's role in the solution plan is....*". If the student selects the first choice, a rule browser is displayed in the right half of the window (see Figure 1), whereas if the student selects "*The role of the fact in the solution plan is....*" then the right part of the window displays a plan browser. The next sections describe how the interaction proceeds in the two cases.

4.1 The rule browser

The rule browser (Figure 2) contains all the system's physics rules, organized in a tree structure so that clicking on the + and – buttons reveals and hides subtrees of the hierarchy. Using this browser, the student finds and selects a rule that justifies the uncovered fact.

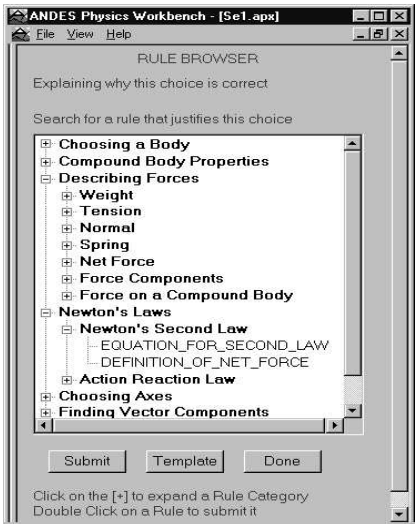


Figure 2: the rule browser

If the student then selects "submit," the SE-Coach will use red/green feedback to indicate whether the selected rule is the one that explains the uncovered information. The SE-Coach does not provide additional help besides red/green feedback, since one feature that makes self-explanation effective for learning is that students elaborate the available material and knowledge by themselves. Thus, when a wrong rule is selected, the only way for the student to correct the mistake is to keep browsing the hierarchy until the correct rule is found. For this reason, the organization of the rule names in the browser is crucial to make the search for the correct rule a thought provoking activity, instead of a frustrating one that may result in the student clicking exhaustively on all the entries until the correct rule is found.

The current organization of the rule hierarchy is the result of successive evaluations with pilot subjects, which helped reduce the amount of floundering observed in the first versions of the browser. A interesting behavior that surfaced during these evaluations is that most

students did not try to click on rule names randomly when they got stuck. Rather, when they could not find plausible candidates in the category that they had expanded they would stop, instead of browsing other parts of the hierarchy. We repeatedly changed the category names and arrangement to maximize the chance that students immediately enter the right part of the hierarchy. We also provided cross references for rules that could plausibly belong to different categories, such as the rule encoding the definition of Net Force, which rightfully belongs to the category *Newton's Second Law* but that students often tried to find in the category *Forces*.

4.2 The rule templates

The rule browser lists only the names of the rules, and most students will need to know more about a rule before they can be sure that it is the explanation they want. To explain more about a rule, the student can click on the "template" button in the rule browser (Figure 2).

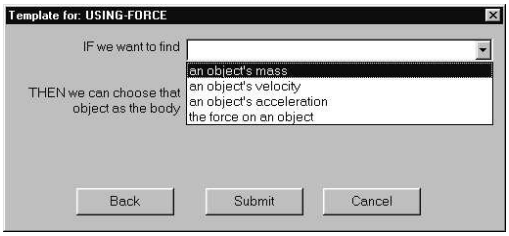


Figure 3: rule template

A dialog box comes up (see Figure 3) with a partial definition of the rule that has blanks for the student to fill in. Clicking on a blank brings up a menu of possible fillers. After completing a template, the student can select "submit," which will cause the SE-Coach to give immediate feedback. By filling in a rule template, students can explain in a much more active way what a rule says than by simply reading and selecting the rules from menus. Again, pilot evaluations were fundamental to assess and improve the clarity and meaningfulness of the template fillers in the pull down menus. For example, we discovered that students tended to ignore fillers that were too verbose, even when they were the only obviously correct choices.

Another relevant insight that we gained from pilot evaluations was that, if students are given too much freedom as to whether to access a template or not, they tend not to do it. In the first version of the system, once a correct rule was selected the student could either click on the *Template* button at the bottom of the browser or click *Done* and quit. Most students never accessed templates. When asked why, they said that they did not remember what a template was, although the experimenter had extensively explained the interface at the beginning of the evaluation session. The simple change of giving only the *Template* choice after rule selection highly increased the percentage of students that filled templates, although students could still close a template without filling it by clicking on the *Cancel* button at the bottom (Figure 3).

4.3 Plan browser

If the student had selected "*The role of the fact in the solution plan is...."* after pushing the self-explain button, then the right part of the window would display a plan browser instead of a rule browser. The plan browser displays a hierarchical tree representing the solution plan for a particular example. The student indicates the explanation of the role of the uncovered fact in the solution plan by navigating through the goal hierarchy and selecting a plan step that most closely motivates the fact. The "submit" button causes SE-Coach to give immediate feedback.

There are no templates associated with the plan browser, since they would simply explicitly spell out information on the plan structure already encoded in the browser hierarchy (e.g. ***If*** *the goal is to apply Newton's law* ***and*** *we have selected a body,* ***then*** *the next subgoal is to describe the properties of this body)*

5 SE-Coach's advice

Initially, self-explanation is voluntary**.** The SE-Coach keeps track of the students' progress through the example, including how much time they looked at a solution item and what they chose to self-explain via the rule and plan browsers. This information is passed to the probabilistic student model, which integrates it with estimates on the student's current knowledge of physics rules to assess what solution items need more self-explanation. In particular, when a student fills a template or select a plan step correctly, the probability of the corresponding rule is updated by taking into consideration the prior probability of the rule and how many attempts the student made to find the correct selection[8].

If a student tries to close an example, the SE-Coach consults the student model to see if there are solution items that require further explanations. The student model returns solution items that correspond to facts or goals derived from rules with a low probability of being known, or items with reading time not sufficient for self-explanation [8].

If the student model indicates that there are lines that need further explanation, the SE-Coach tells the student "*You may learn more by self-explaining further items. These items are indicated by pink covers*", and colors some of the boxes pink instead of gray. It also attaches to each item a more specific hint such as "*Please self-explain by using the Rule browser*" *or "Please read more carefully*". The color of the boxes and the related messages change dynamically as the student performs more reading and self-explanation actions.

If the student tries to close the example when there are still some pink covers left, the SE-Coach generates a warning such as "*There are still some items that you could self-explain. Are you sure you want to exit?*", but it lets the student quit if the student wants to.

The SE-Coach's advice is probably the feature that was most affected by the feedback from pilot evaluations. In the original version of the system, the Coach would point out lines that required self-explanations one at a time, instead of indicating them all at once by changing their color. When the student tried to close the example, the SE-Coach would generate a first, generic warning such as "*There are still some items that you could self-explain. Do you want to try?"* The student could either (a) reject the advice, (b) accept it and go back to study the example without any further indication of what to self-explain, (c) ask for more specific hints.

If the student chose the latter, the SE-Coach would say, for example "*Why don't you try to use the rule browser to explain this line?*", and it would uncover the line. At this point the student would go back to the example, and possibly explain the line, but the only way for the student to get additional suggestions from the Coach would be to close the example again.

The rationale behind this design was to stimulate as much spontaneous self-explanation as possible. We thought that directing the student to a particular example line could be enough to also trigger explanations on other lines. This did not happen. Either students were natural self-explainers and explained most of the example the first time through, or they strictly followed individual SE-Coach hints but rarely initiated any additional self-explanation. For non-spontaneous self-explainers, the interaction with the coach would quickly become quite uninspiring, since after doing what the Coach had suggested (e.g. finding a rule name in the rule browser), they would try to close the example and they would get another hint ("*there is something else that you could self-explain, do you want me to show you*?"), suggesting further explanation either on the current line via template/plan browser or on a different line. A student would have to repeat this cycle to access each new piece of advice, and most students lost interest and chose to close the example after the first couple of hints.

The current design, based on the coloring of example lines, allows the students to see at once all the parts that they should self-explain, and what workbench tool they should use for the explanations. It also gives the students better feedback on the progresses that they making, since line color and hints change dynamically as students generate more self-explanations.

6 Empirical evaluation of the SE-Coach

Once we had iteratively improved the system design through pilot evaluations, we performed an empirical evaluation to test its effectiveness.

6.1 Experiment design

We conducted a laboratory experiment with 56 college students who were taking introductory physics classes at the University of Pittsburgh, Carnegie Mellon University and U.S.Naval Academy. The design had two conditions:

Control: 27 students studied examples with the masking interface only.

Experimental: 29 students studied examples with the SE-Coach.

The evaluation consisted of one session in which students 1) took a paper and pencil physics test, 2) studied examples on Newton's second law with the system, 3) took a paper and pencil post-test with questions equivalent but not identical to the ones in the pre-test, 4) filled out a questionnaire designed to assess the students impressions on the system.

Timing was a heavy constraint in the experiment. The sessions needed to be held when students already had some theoretical knowledge to understand the examples and generate self-explanations, but were not so far ahead into the curriculum that our examples would be too trivial for them. To satisfy this constraint, we ran subjects in parallel in one of the Pitt University computer labs. Another constraint was that we had to concentrate the evaluation in one session, to avoid that the post-test performance be influenced by knowledge that students were gaining from their physics class. This, and the fact that the computer lab was available in 3-hour slots, obligated us to limit the length of pre-test and post-test. Thus, we could not insert any items to specifically test knowledge gained from goal-based explanations built with the plan browser, and we had to rely on the possibility that students would show such knowledge in the resolution of the problem solving questions available in the test.

In order to roughly equate time on task, students in the control condition studied 6 examples and students in the experimental condition studied 3 examples. Despite this, there is a statistically significant difference between the average time on task of the experimental group (52') and the control group (42' 32''). However, we found no significant correlation of time on task with post-test scores.

6.2 Results

Two different grading criteria were used for pre and post test. The first criterion, called *objective grading*, comprised only those questions in the test that required a numeric answer or a selection from a set of choices, and looked only at the correctness of the final result.

Group	N	Mean	StdDev
control	27	2.30	2.38
se-group	29	2.38	1.76

Group	N	Mean	StdDev
control	27	5.04	4.35
se-group	29	6.04	4.49

Table 1: (a) objective-based gain scores (b) Feature-based gain scores

The second criterion, called *feature-based grading*, included also those items in the test that required more qualitative definitions, and took into account how students got their answers. For both grading systems, there were no significant differences between conditions on the pre-test scores. Unfortunately, the gain scores were also not significantly different, although the trend was in the right direction and the gain score difference was higher for feature-based grading (table 1), which was more apt to capture knowledge gains due to self-explanation.

A possible explanation for the non-significant result is that students in the experimental condition did not generate sufficient self-explanations with the Workbench tools, because they had problems using them and/or because they did not follow the SE-Coach advice. To test this explanation, we extracted from the experimental group log data file information on Workbench tools usage and SE-Coach's performance: (a) how many times students initiated

self-explanations with rule browser, templates and plan browser, (b) how many of these explanations were successful, (c) how many attempts it took the students on average to find a correct answer in each of the self-explanation tools, or to decide to quit the explanation, and (d) how often students followed the SE-Coach's advice.

Rule browser usage. On average, students initiated 28.8 rule browser explanations, which represents 62% of the total rule browser explanations that can be generated in the available examples. Of the initiated rule browser explanations, 87% successfully ended with the selection of the correct rule. On average it took the students 1.27 attempts to get the correct answer, with a average maximum of 9.2 attempts. Although on average students did not flounder much to find a correct rule, for almost all of them there was at least one rule that was very hard to find. The rule browser accesses that failed to find the correct rule took an average of 4 attempts and students spent an average of 4 minutes on failed rule browser explorations, a minor fraction of the average total time on task (52 minutes).

This data shows that, although the rule browser did not seem to cause many problems to students, it could have generated some degree of distraction and frustration in the few situations in which a student took a long time to find the correct rule or could not find it at all. The system may benefit from an additional form of help, that leads the student to find the right browser category when the student is floundering too much. This was in fact the main suggestion that students wrote in the questionnaire that they filled after the post-test.

Template usage. On average students accessed 23.8 templates, 55.5% of the available template explanations. This data is not indicative of how effectively templates stimulate self-explanations since, as we described in Section 4.2, template access is mandatory when a correct rule is selected in the Rule browser. More indicative is the fact that, although it is not mandatory to fill a template after opening it, 97% of the accessed templates were filled correctly, with an average of only 0.5 attempts and an average maximum of 2.5 attempts. On average students spent only 59 seconds trying to fill templates for which they could not find the correct answer. This data allows us to discard user interface problems with templates as a cause for the non-significant results.

Plan browser usage. Students initiated only 38% of the possible plan browser explanations. Students did not have many problems using the plan browser. Of the initiated explanation, 85% resulted in the selection of the correct plan step, with an average of 1 attempt. Students spent on average only 29 seconds on plan browser accesses that did not lead to a correct explanation. Despite good plan browser performance, we could not detect any gain in the students' planning knowledge because the post-test did not have any question that specifically tapped it. Furthermore, many students wrote in the questionnaire that they found the plan browser not very useful. This outcome is not surprising. As we mentioned in Section 2, goal-related explanations are quite unnatural for students. This is especially true if students don't have any theoretical knowledge on the notion of solution planning. The plan browser was designed with the idea of evaluating the system at the Naval Academy, with students that had been introduced to the idea of abstract planning by the physics professors participating to the Andes project. We hope to be able to perform this evaluation in the near future, to verify the effectiveness of the plan browser when used in the optimal instructional context.

SE-Coach result. As described in Section 5, the SE-Coach gives its suggestions by changing the color of the lines to self-explain and by attaching to each line specific hints indicating if the line should be explained with rule browser/template, with the plan browser, or if it should be simply read more carefully. In the three evaluation examples, the SE-Coach can generate a maximum of 43 rule browser hints, 34 plan browser hints and 43 hints to read more carefully. The Coach gave an average of 22.6 rule browser hints, 22.4 plan browser hints and 7 reading hints. Each student followed an average of 38.6% of these rule browser hints, 42% of the plan browser hints and 34% of the hints suggesting to read more carefully.

As we explained in Section 5, the SE-Coach hints are given based on the student's model assessment of what solution items correspond to rules that have low probability (< 0.75) of being known by the student. As students correctly explain the suggested solution items, probabilities of the corresponding rules are increased in the student model. So an indicator of the effectiveness of the SE-Coach is the percentage of physics and planning rules that, at the end of the evaluation session, have changed their probability from less to more than 0.75. On average, 79.3% of the physics rules used in the three examples and 77% of the plan rules reached the 0.75 threshold.

6.3 Results discussion

The results on Workbench usage suggest that user interface problems are not likely to be a primary cause for the non-significant difference in gain scores, although changes that reduce floundering in the rule browser could help improve the effectiveness of the system. On the other hand, the results on the effectiveness of the SE-Coach's advice show that, although the current design works much better than the original one described in Section 5, better learning for the experimental group could be obtained with a stronger form of coaching, that leads students to self-explain more exhaustively. As a matter of fact, in all the experiments in which human tutors elicited self-explanation, the tutor made sure that students self-explained every item in the target examples. We did not want to make the SE-Coach's suggestions mandatory because they are based on a probabilistic student model whose accuracy had not been tested at the time of the evaluation. In particular, the student model predictions strongly depend on estimates of student's initial physics knowledge [8]. At the time of the evaluation we had no way of obtaining these estimates for every student, so we assigned to every rule a probability of 0.5. Given the possible inaccuracy of the model, we did not want to risk frustrating the students by forcing them to explain example lines that they may have already understood. We may obtain better results from the SE-Coach with an evaluation in which we set the initial probabilities of the student model by using the results of the student's pre-test, and we make the SE-Coach hints mandatory.

Three more hypotheses for the lack of significant gain scores should be considered. The first hypothesis is that students in the control group self-explained as much as students in the experimental group. This hypothesis is not easy to test since we have no simple way to ascertain whether control students self-explained or not. We are currently working on the analysis of control group log data files, to see if we can identify any correlation between how students read the examples and their posttest results.

The second hypothesis is that the self-explanations generated with the Workbench did not stimulate as much learning as verbal self-explanations do. Also, the fact that students must concentrate on the self-explanations allowed by the Workbench may actually inhibit the generation of self-explanations based on knowledge outside the physics domain that, as we discussed in Section 2, appeared quite frequently in experiments on verbal self-explanations. A possible way to test this second hypothesis is to compare the SE-Coach interface with an interface that allows students to express their self-explanations by writing.

Lastly, given that the experimental group post-test scores were higher than the control group scores, but the difference was not large compared to the standard deviation, it may be that the SE-Coach works fine but students did not use it long enough. If students studied twice as many examples, perhaps the difference in learning between the two groups would be large enough to be statistically significant.

7 Conclusions

The SE-Coach is a tutoring module that focuses on teaching the meta-cognitive skill known as self-explanation, instead of directly teaching cognitive skills related to a particular

instructional domain. Many studies show that self-explanation, the process of clarifying and making more complete to oneself the solution of an example, can improve problem solving performance, and that guiding self-explanation can extend these benefits.

We believe that empirical evaluations are fundamental for the development of instructional systems of real effectiveness. This is especially true for the SE-Coach, since it focuses on a learning process whose underlying mechanisms are still unclear and under investigation. In this paper, we described how the system evolved through pilot evaluations from the original design proposed in [7] to its current version. In particular, we illustrated how these evaluations shaped two fundamental elements of the system (a) the SE-Coach interface, known as the Workbench, that provides specific tools for constructing self-explanations, and (b) the SE-Coach's advice, which uses the assessment of a probabilistic student model to elicit self-explanations that can improve the students' understanding of the example.

We also illustrated the results of a formal evaluation that we performed with 56 college students to test the effectiveness of the system. Although the learning trend was in the right direction, the results did not reach statistical significance. However, the analysis of the log data files collected during the evaluation allowed us to understand how students used the system, and to generate hypotheses to explain the lack of statistically significant results.

We plan to start testing with formal evaluations those hypotheses that involve minor changes to the system (adding additional help to use the Workbench tools, making the SE-Coach's advice mandatory) and minor changes to the experiment design (adding more specific test questions to tap all the knowledge addressed by the SE-Coach, increasing the time on task by making students study more examples). The insights provided by these new evaluations could be used in the future to develop and study alternative self-explanation user interfaces and coaches in order to see which ones encourage the most learning.

8 References

[1] Chi, M.T.H., et al., Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, 1989. 15: p. 145-182.

[2] Chi, M.T.H., Self-explaining: A domain-general learning activity, in Advances in Instructional Psychology, R. Glaser, Editor. in press, Erlbaum: Hillsdale, NJ.

[3] Renkl, A., Learning from worked-examples: A study on individual differences. Cognitive Science, 1997. 21(1): p. 1-30.

[4] Bielaczyc, K., P. Pirolli, and A.L. Brown, Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem-solving. Cognition and Instruction, 1995. 13(2): p. 221-252.

[5] Chi, M.T.H., et al., Eliciting self-explanations improves understanding. Cognitive Science, 1994. 18

[6] VanLehn, K., Conceptual and meta learning during coached problem solving, in ITS96: Proceeding of the Third International conference on Intelligent Tutoring Systems., C. Frasson, G. Gauthier, and A. Lesgold, Editors. 1996, Springer-Verlag: New York.

[7] Conati, C., J. Larkin, and K. VanLehn, A computer framework to support self-explanation, in Proceedings of the Eighth World Conference of Artificial Intelligence in Education. 1997.

[8] Conati, C. A student model to assess self-explanation while learning from examples. To appear in Proc. of UM'99, 7th Int. Conference on Student Modeling, Banff, Canada.

[9] Catrambone, R., Aiding subgoal learning: effects on transfer. Journal of educational psychology, 1995. 87

[10] Conati, C., et al., On-line student modeling for coached problem solving using Bayesian networks, in User Modeling: Proc. of the 6th International conference, UM97, 1997, Spring Wien: New York.