Adaptive support for coaching meta-cognitive skills

Cristina Conati

Department of Computer Science
University of British Columbia
2366 Main Mall,, Vancouver, BC V6T2Z3, Canada.
conati@cs.ubc.ca

Abstract

We describe a computational framework designed to provide adaptive support to learning from examples by coaching the meta-cognitive skill known as self-explanation - generating explanations to oneself to clarify an example solution. The framework includes an interface to scaffold self-explanation, a probabilistic student model and a coaching component. The probabilistic student model allows the coach to change the interface scaffolding in order to improve the student's studying behavior. We discuss how the types of tailoring and scaffolding that the framework enables can be extended to generate more adaptive support to effectively learning from examples.

Background work

As computers become increasingly more available in instructional settings, they provide an invaluable opportunity to explore innovative ways to improve human learning. Research in cognitive science has been long investigating the variety of cognitive processes that trigger learning and the educational interventions that can bolster those processes outside the limited set targeted by traditional schooling. Computers have good potential to help deploy these interventions and the use of Artificial Intelligence techniques has helped turn this potential into reality. However, to date research on Intelligent Computer Aided Instruction (ICAI) has mainly focused on helping students acquiring domain specific, problem solving skills.

We are interested in exploring innovative ways in which computers can enhance education and learning, by covering other phases of the learning process and by helping students acquire meta-cognitive, domain independent learning skills (Shute and Psotka 1996).

In our current research, we have been focusing on learning from examples and on the meta-cognitive learning skill known as self-explanation – generating explanations to oneself to clarify an example's worked out solution. Effectively learning from examples is important because students heavily rely on examples when learning a new skill. Furthermore, several studies in cognitive science show that students who spontaneously self-explain when they study examples learn more (Renkl 1997; Chi in press). These studies also show that most students do not spontaneously self-explain. However, students start self-explaining when they are guided to do so (Bielaczyc, Pirolli et al. 1995; Renkl, Stark et al. 1998).

These results suggest that it can be greatly beneficial to integrate computer-based support to problem solving with individualized guidance to learning from examples through self-explanation. This entails providing a computer tutor with the capability to monitor students as they study examples and to elicit further self-explanation that can improve the students' example understanding. This task, apparently simple, entails additional hardship for the traditional challenges of ICAI - user interface design, student modeling and providing adequate help.

User interface design. Example studying and self-explanation are mainly reading and verbal tasks. How to design an interface that monitors students' attention and allows them to constructively generate their self-explanations, given that eye-tracking technology and natural language processing are still not powerful and reliable enough to be readily usable in non-laboratory setting?

Student modeling. To model a student during example studying requires assessing how well the student understands the example and learns from it. How to perform this assessment by relying on actions like reading and self-explaining, that are largely ambiguous and have less direct correspondence to example understanding than problem solving actions have to problem solutions?

Proving adequate help. One of the benefits of self-explanation come from the fact that spontaneous self-explainers selectively generate self-explanations to target their specific needs (Chi in press). How can a computer tutor decide what further self-explanations can be more beneficial for those students that do not spontaneously self-explain? When and how should the tutor elicit these self-explanations from students that are naturally reluctant to self-explain?

In (Conati 1999) we have proposed solutions to these challenges, including:

1. An interface that monitors student's attention through a masking mechanism that requires the student to explicitly move the mouse over an example part in order to uncover and view it. Pilot studies have revealed that the masking mechanism provides a first level of scaffolding for self-explanation, by helping students focus their attention. The interface also provides menu-based tools to constructively generate explanations without using natural language. The tools reflect the content of a rule-based representation of the instructional domain and provide structured accessed to it.

2. A student model based on a Bayesian network (Pearl 1988), to handle in a principled way the high level of uncertainly involved in modeling learning from reading and self-explanation actions. The student model Bayesian network is created automatically, for each example and for each interaction, from the rule-based domain representation and from the student's interface actions. The model uses latency data on student's attention and current estimates of the student's domain knowledge to evaluate the probability that the student is spontaneously self-explaining. The model also records the student's self-explanations explicitly built through the interface and uses them to estimate the changes in the student's knowledge due to the interaction with the system.

3. An algorithm to decide, given the probabilities in the student model, which parts of the example the student should further self-explain and how. Through successive pilot evaluations, we have devised a modality of intervention that coveys this information to the students through dynamic changes in the example presentation (e.g. highlighting of example parts that require further self-explanations and generation of prompts that suggest how to generate them).

The proposed solutions have been tested in the context of a computer tutor, the SE (self-explanation)-Coach that provides tutoring for self-explanation within Andes, a tutoring system for Newtonian physics that supports students during both example studying and problem solving. An empirical evaluation of the SE-Coach has confirmed the usability and effectiveness of the SE-Coach interface (Conati and VanLehn 1999) and has provided statistical evidence that the tutorial interventions suggested by the probabilistic student model increase students' learning (Conati 1999).

Toward more adaptive support to effectively learn from examples.

Although our previous work formalizes some of the crucial processes involved in helping students learn through self-explanation, it does not cover all of them. One of our research goals is to extend the work to make it a more complete computational framework to support learning through self-explanation, in terms of the types of self-explanations that it handles and of the tailored guidance that it provides to generate them.

Currently, the framework explicitly guides only self-explanations local to a certain solution step (justify the step in terms of the domain theory; map the step onto the underlying solution plan). However, often the examples found in instructional material have gaps, that is they omit some of the solution steps. This is because as students become more familiar with a new topic, they should be able to generate self-explanations to fill the missing steps, thus becoming gradually accustomed to generate a problem solution. We want to add to the existing framework the capability to (a) guide self-explanations that fill example solution gaps; (b) dynamically tailor the amount and kind of the omitted steps to scaffold students in the transition from example studying to problem solving.

The probabilistic student model is already set up to monitor gap-filling self-explanations (Conati 1999), but the existing self-explanation interface cannot explicitly support them. Thus, adding to the framework the capability to scaffold gap-filling self-explanations entails the following objectives:

- Devise an interface that can guide the generation of the missing parts of the example solution.
- Formalize criteria to tailor the level of detail of an example solution to the student's knowledge state.

This second objective includes exploring how to make the generation of a new example completely automatic. Currently, a problem solver automatically builds the internal representations of an example complete solution and the related student model Bayesian network, from the example initial data and from the domain rule-based representation. However, the interface presentation of the example and the mapping that links what the student does on this presentation to its internal formalization is coded by hand. We will explore how to automatically generate the example presentation from the example internal representation. This will not only allow the system to adapt the level of detail of an example solution to the student's needs. It will also facilitate adding new examples for students to study.

We also plan to start experimenting with available eye-tracking and natural language technology to verify if and how they can improve the coaching of self-explanation. Natural language and eye-tracking could make the interaction with the system less constrained, by allowing students to generate their explanations in free text and by removing the need for the masking mechanism to track attention. However, less constrained interaction does not necessarily imply better learning. In particular, we conjecture that students that do not spontaneously self-explain or that have lower verbal skills will benefit more from the stronger guidance provided by the current, menu based interface with masking mechanism. As students' knowledge and learning skills improve through the interaction with the system, they may benefit more from the less constrained interaction supported by eye-tracking and natural language. We plan to test our hypothesis by augmenting the probabilistic student model to support the dynamic adaptation of input modalities to the student's level of expertise and by empirically testing how this affects students' learning.

References

Bielaczyc, K., P. Pirolli, et al. (1995). "Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem-solving." *Cognition and Instruction* **13**(2): 221-252.

Chi, M. T. H. (in press). "Self-Explaining: The dual process of generating inferences and repairing mental models*." Advances in Instructional Psychology*.

Conati, C. (1999). An intelligent computer tutor to guide self-explanation while learning frome examples. Intelligent Systems Program. Pittsburgh, University of Pittsburgh.

Conati, C. and K. VanLehn (1999). Teaching meta-cognitive skills: implementation and evaluation of an tutoring system to guide self-explanation while learning from examples. *AIED'99, 9th World Conference of Artificial Intelligence and Education*, Le Mans, France.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA, Morgan-Kaufmann.

Renkl, A. (1997). "Learning from worked-examples: A study on individual differences." *Cognitive Science* **21**(1): 1-30.

Renkl, A., R. Stark, et al. (1998). "Learning from worked-out examples: the effects of example variability and elicited self-explanation." *Contemporary educational psychology* **23**: 90-108.

Shute, V. and J. Psotka (1996). Intelligent tutoring systems: Past, Present and Future. *Handbook of Research on Educational Communications and Technology. D. Jonassen, Scholastic Publications*.