Scaffolding Self-Explanation to Improve Learning in Exploratory Learning Environments.

Andrea Bunt, Cristina Conati and Kasia Muldner

Department of Computer Science, University of British Columbia
201-2366 Main Mall
Vancouver, British Columbia
V6T 1Z4
(604)822-4632

Abstract. Successful learning though exploration in open learning environments has been shown to depend on whether students possess the necessary meta-cognitive skills, including systematic exploration, hypothesis generation and hypothesis testing. We argue that an additional meta-cognitive skill crucial for effective learning through exploration is self-explanation: spontaneously explaining to oneself available instructional material in terms of the underlying domain knowledge. In this paper, we describe how we have expanded the student model of ACE, an open learning environment for mathematical functions, to track a learner's self-explanation behaviour and how we use this model to improve the effectiveness of a student's exploration.

1 Introduction

Several studies in Cognitive Science and ITS have shown the effectiveness of the learning skill known as *self-explanation*, i.e., spontaneously explaining to oneself available instructional material in terms of the underlying domain knowledge [6]. Because there is evidence that this learning skill can be taught (e.g., [2]), several computer-based tutors have been devised to provide explicit support for self-explanation. However, all these tutors focus on coaching self-explanation during fairly *structured* interactions targeting problem-solving skills (e.g., [1], [7, 8] and [10]). For instance, The SE-Coach [7][8] is designed to model and trigger students' self-explanations as they study examples of worked out solutions for physics problems. The Geometry Explanation Tutor [1] and Normit-SE [10] support self-explanations of problem-solving steps, in geometry theorem proving and data normalization respectively. In this paper, we describe how we are extending support for self-explanation to the type of *less structured* pedagogical interactions supported by open learning environments.

Open learning environments place less emphasis on supporting learning through structured, explicit instruction and more on allowing the learner to freely explore the available instructional material [11]. In theory, this type of active learning should enable students to acquire a deeper, more structured understanding of concepts in the

domain [11]. In practice, empirical evaluations have shown that open learning environments are not always effective for all students. The degree of learning from such environments depends on a number of student-specific features, including activity level, whether or not the student already possesses the meta-cognitive skills necessary to learn from exploration and general academic achievement (e.g., [11] and [12]).

To improve the effectiveness of open learning environments for different types of learners, we have been working on devising adaptive support for effective exploration. The basis of this support is a student model that monitors the learners' exploratory behaviour and detects when they need guidance in the exploration process. The model is implemented in the Adaptive Coach for Exploration (ACE), an open learning environment for mathematical functions [3, 4]. An initial version of this model integrated information on both student domain knowledge and the *amount* of exploratory actions performed during the interaction to dynamically assess the effectiveness of student exploration. Empirical studies showed that hints based on this model helped students learn from ACE. However, these studies also showed that the model sometimes overestimated the learners' exploratory behaviour, because it always interpreted a large number of exploratory actions as evidence of good exploration. In other words, the model was not able distinguish between learners who merely performed actions in ACE's interface and learners who *self-explained* those actions.

In this paper, we describe 1) how we modified ACE's student model to assess a student's self-explanation behaviour, and 2) how ACE uses this assessment to improve the effectiveness of a student's exploration through tailored scaffolding for self-explanation. ACE differs from Geometry Explanation Tutor and the Normit-SE not only because it supports self-explanation in a different kind of educational activity, but also because these systems do not model a student's need or tendency to self-explain. The Geometry Explanation Tutor prompts students to self-explain *every* problem-solving step, Normit-SE prompts students to self-explain every *new* or *incorrect* problem-solving step. Neither of these systems considers whether it is dealing with a self-explainer who would have initiated the self-explanation regardless of the coach's hints, even though previous studies on self-explanations have shown that some students do self-explain spontaneously [6]. Thus, these approaches are too restrictive for an open learning environment, because they may force spontaneous self-explainers to perform unnecessary interface actions, contradicting the idea of interfering as little as possible with students' free exploration. Our approach is closer to the SE-Coach's, which prompts for self-explanation only when its student model assesses that the student actually needs the scaffolding [9]. However, the SE-Coach mainly relies on the time spent on interface actions to assess whether or not a student is spontaneously self-explaining. In contrast, ACE also relies on the assessment of a student's self-explanation tendency, including how this tendency evolves as a consequence of the interaction with ACE. Using this richer set of information, ACE can provide support for self-explanation in a more timely and tailored manner.

In the rest of the paper, we first describe ACE's interface, and the general structure of its student model. Next, we illustrate the changes made to the interface and the model to provide explicit guidance for self-explanation. Finally, we illustrate the model's behaviour based on sample simulated scenarios.

2 The ACE Open Learning Environment

ACE is an adaptive open learning environment for the domain of mathematical functions. ACE's activities are divided into units, which are collections of exercises. Figure 1 shows the main interaction window for two of ACE's units: the *Machine Unit* and the *Plot Unit*. Ace's third unit, the *Arrow Unit*, is not displayed for brevity.

The Machine and the Arrow Units allow a learner to explore the relation between input and output of a function. In the Machine Unit, the learner can drag the inputs displayed at the top of the screen to the tail of the "machine" (the large arrow shown in Fig. 1, left), which then computes the corresponding output. The Arrow Unit allows the learner to match a function's inputs and outputs, and is the only unit within ACE that has a clear definition of correct behaviour. The Plot Unit (Fig. 1, right), allows the learner to explore the relationship between a function's graph and its equation by manipulating one entity, and then observing the corresponding changes in the other.

To support the exploration process, ACE includes a coaching component that provides tailored hints when ACE's student model assesses that students have difficulties exploring effectively. For more detail on ACE's interface and coaching component see [4]. In the next section, we describe the general structure of ACE's student model.

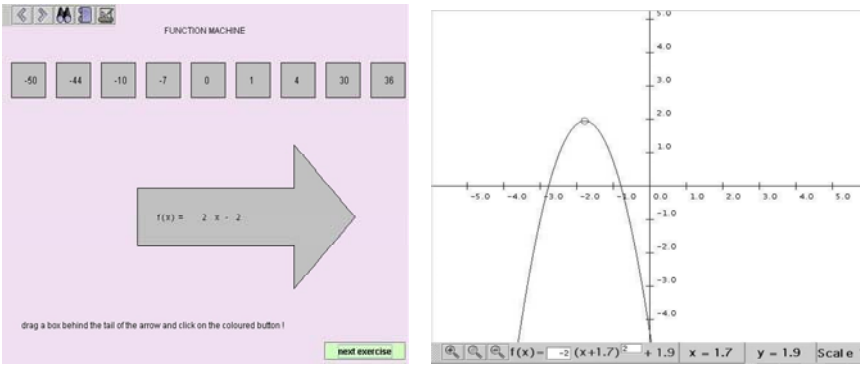

Fig. 1. Machine Unit (Right) and Plot Unit (Left)

3 General Structure of ACE's Student Model

ACE's student model uses Bayesian Networks to manage the uncertainty inherent to assessing students' exploratory behaviour. The main cause of this uncertainty is that both exploratory behaviour and the related meta-cognitive skills are not easily observable unless students are required to make them explicit. However, forcing students to articulate their exploration steps would clash with the unrestricted nature of open learning environments.

The structure of ACE's student model derives from an iterative design process [3] that gave us a better understanding of what defines effective exploration. Figure 2 shows a high-level description of this structure, which comprises several types of nodes used to assess exploratory behaviour at different levels of granularity:

- *Relevant Exploration Cases*: the exploration of individual exploration cases in an exercise (e.g., dragging the number 3, a small positive input, to the back of the function machine in the Machine Unit).
- *Exploration of Exercises*: the exploration of individual exercises.
- *Exploration of Units*: the exploration of individual units.
- *Exploration of Categories*: the exploration of groups of relevant exploration cases that appear across multiple exercises (e.g., all the cases involving a positive slope in the Plot unit).
- *Exploration of General Concepts*: the exploration of general domain concepts (e.g., the input/output relation for different types of functions).

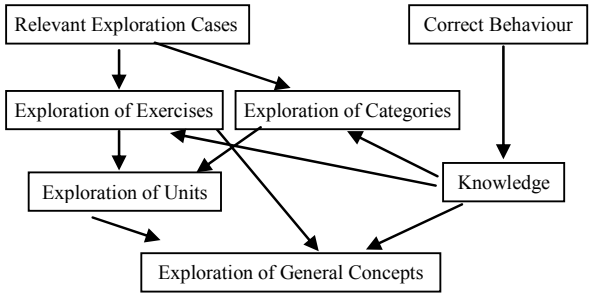

Fig. 2. High-Level Structure of ACE's Student Model

The links among the different types of exploration nodes represent how they interact to define effective exploration. Exploration nodes have binary values representing the probability that the learner has sufficiently explored the associated item.

ACE's student model also includes binary nodes representing the probability that the learner understands the relevant domain knowledge (summarized by the node *Knowledge* in Fig. 2). The links between knowledge and exploration nodes represent the fact that the degree of exploration needed to understand a concept depends on how much knowledge of that concept a learner already has. Knowledge nodes are updated only through actions for which there is a clear definition of correctness (e.g., linking inputs and outputs in the Arrow Unit).

4 Extending ACE to Track and Support Self-Explanation

As we mentioned in the introduction, initial studies on ACE generated encouraging evidence that the system could help students learn better from exploration [3, 4].

However, these studies also showed that sometimes ACE overestimated students' exploratory behaviour (as indicated by post-test scores).

We believe that a likely cause of this problem was that ACE considered a student's interface actions to be sufficient evidence of good exploratory behaviour, without taking into account whether s/he was *self-explaining* the outcome of these actions. To understand how self-explanation can play a key role in effective exploration, consider a student who quickly moves a function graph around the screen in the Plot unit, without reflecting on how these movements change the function equation. Although this learner is performing many exploratory actions, s/he can hardly learn from them because s/he is not self-explaining their outcomes. We observed this exact behaviour in a number of our subjects who tended to spend little time on exploratory actions, and who did not learn the associated concepts (as demonstrated by pre-test / post-test differences).

To address this limitation, we decided to extend ACE's interface and student model to provide tailored support for self-explanation. We first describe modifications made to ACE's interface to provide this support.

4.1 Tailored Support for Self-explanation in ACE

The original version of ACE only generated hints indicating that the student should further explore some elements of a given exercise. The new version of ACE can also generate tailored hints to support a student's self-explanation, if this is detected to be the cause of poor exploration. Deciding when to hint for self-explanation is a challenging issue in an open learning environment. The hints should interfere as little as possible with the exploratory nature of the interaction, but should also be timely so that even the more reluctant self-explainers can appreciate their relevance. Thus, ACE hints to self-explain individual actions are provided as soon as the model predicts that the student is not self-explaining their outcomes when s/he should be.

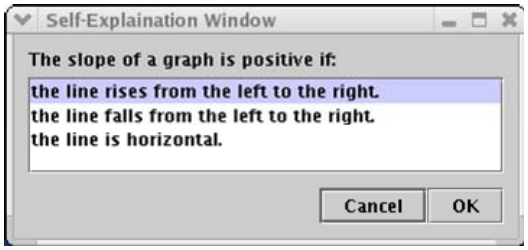

Fig. 3. Example of a Self-Explanation Tool for the Plot Unit**.**

The first of these hints is a generic suggestion to slow down and think a little more about the outcome of the performed actions. Following the approach of the SE-Coach [7, 8], ACE provides further support for those students who cannot spontaneously self-explain by suggesting the usage of interface tools, designed to help students generate relevant self-explanations. One type of hint targets self-explanations related

to the outcome of specific actions, or *exploration cases*. For instance, Figure 3 shows a dialogue box involved in eliciting a self-explanation for an exploration case associated with a linear function in the Plot Unit. The multiple-choice approach is used here to avoid dealing with parsing of free text explanations, although this is something that we may change in future versions of the system, following [1]. After the student selects one of the statements, the coach provides feedback for correctness. Providing feedback for correctness is consistent with the view adopted by other approaches to coaching for self-explanation: although incorrect self-explanation can still be beneficial for learning [5], helping students to generate correct self-explanations further increases the efficacy of this meta-cognitive skill [1][2] [7][10]. ACE also provides hints to self-explain an exercise as a whole (e.g., the behaviour of a constant function in the machine unit), which are generated when a student tries to leave that exercise. We now describe the changes made to ACE's student model to support the hinting behaviour just described.

5 New Model for Self-Explanation

Obtaining information on a student's self-explanation behaviour in an open learning environment is a difficult challenge. The tools presented in the previous section can provide *explicit* evidence that a student is self-explaining, but because ACE does not force students to use these tools, the model must also try to assess whether or not the learner is self-explaining *implicitly*, i.e., without any tool usage. The only evidence that can be used toward this end is time a student spends on each exploratory action. Unfortunately, this evidence is clearly ambiguous, since a long time spent on an action does not necessarily mean reflection on that action. Similarly, a sequence of actions performed very quickly could be either due to a low self-explanation (SE) tendency, or to a student's desire to experiment with the interface before starting to explore the exercise more carefully. Without any further information on the student's general SE tendency, the latter ambiguity can be solved only by waiting until the student asks to leave the exercise (as the SE-Coach does, for instance [7, 8]). This, however, misses the opportunity to generate hints in context, when they can be best appreciated by the student.

Therefore, to improve its evaluation of students' exploration behaviour, ACE's student model includes an assessment of a student's SE tendency. In addition to assessing SE tendency, the model also assesses how it evolves during the interaction with ACE, to model the finding that SE tendency can be improved through explicit coaching [2]. To represent this evolution, we moved from a Bayesian Network that was mostly static [3] to a full Dynamic Bayesian Network. In this network, a new time slice is created after each student exploratory or SE action. These are described below.

5.1 Implicit Self-Explanation Slices

In the absence of explicit SE actions, the model tries to assess whether the student is implicitly self-explaining each exploratory action. Figure 4 shows two time slices created after two exploratory actions. Since the remainder of the exploration hierarchy (see Fig. 2) has not undergone significant change, we omit it for clarity. In this figure, the learner is currently exploring exercise 0 (node e_0 in Fig. 4), which has three relevant exploration cases (e_0Case_0, e_0Case_1 and e_0Case_2 in Fig. 4). At time T, the learner performed an action corresponding to the exploration of e_0Case_2; at time T+1, the action corresponded to e_0Case_1. Nodes representing the assessment of self-explanation are shaded grey. All nodes in the model are binary, except for *time*, which has values *Low/Med/High*. We now describe each type of self-explanation node:

- *Implicit SE:* represents the probability that the learner has self-explained a case implicitly, without using the interface tools. The factors influencing this assessment include the time spent exploring the case and the stimuli that the learner has to self-explain. Low time on action is always taken as negative evidence for implicit explanation. The probability that self-explanation happened with longer time depends on whether there is a stimulus to self-explain.
- *Stimuli to SE:* represents the probability that the learner has stimuli to self-explain, either from the learner's general SE tendency or from a coach's explicit hint.
- *SE Tendency:* represents the model's assessment of a student's SE tendency. The prior probability for this node will be set using either default population data or, when possible, data for a specific student. In either case, the student model's belief in that student's tendency will subsequently be refined by observing her behaviour with ACE. More detail on this assessment is presented in section 5.3.
- *Time:* represents the probability that the learner has spent a sufficient time covering the case. We use time spent as an indication of effort (i.e., the more time spent the greater the potential for self-explanation). Time nodes are observed to low, medium and high according to the intervals between learner actions.
- *Coach Hint to SE:* indicates whether or not the learner's self-explanation action was preceded by a prompt from the coach.

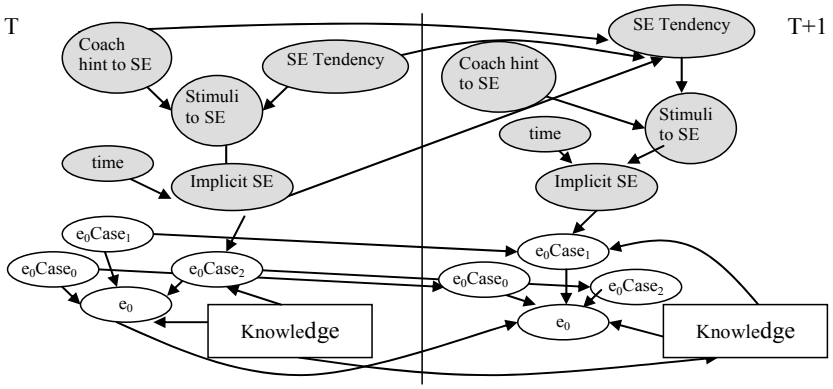

Fig. 4. Nodes Related to Implicit Self-Explanation Actions

We now discuss the impact of the above nodes on the model's assessment of the learner's exploratory behaviour. Whether or not a learner's action implies effective exploration of a given case (e.g., e_0Case_2) depends on the probability that: 1) the student self-explained the action and 2) s/he knows the corresponding concept, as assessed by the set of knowledge nodes in the model (summarized in Fig. 4 by the node *Knowledge*). In particular, the CPT for a case exploration node is defined so that low self-explanation with high knowledge generates an assessment of adequate exploration and, thus, does not trigger a Coach hint. This accounts for the fact that a student with high knowledge of a concept does not need to dwell on the related case to improve her understanding [3]. Note that the assessment of *implicit SE* is independent from the student's knowledge. We consider implicit self-explanation to have occurred regardless of correctness, consistent with the original definition of self-explanation [6].

5.2 Explicit Self-Explanation Slices

Usage of ACE's SE tools provides the model with additional information on the student's self-explanation behaviour. Self-explanation actions using these tools generate explicit self-explanation slices; two such slices are displayed in Figure 5. Compared to implicit SE slices, explicit SE slices include additional evidence nodes representing: 1) the usage of the SE tool (*SE Action* node in Fig. 5), and 2) the correctness of this action (*Correctness* node in Fig. 5). The *SE Action* node, together with the time the student spent on this action, influences the assessment of whether an explicit self-explanation actually occurred (*Explicit SE* node in Fig. 5). As was the case for the implicit SE slices, correctness of the SE action does not influence this assessment. However, correctness does influence the assessment of the student's corresponding knowledge, since it is a form of explicit evidence. Consequently, if the explicit SE action is correct, the belief that the student effectively explored the corresponding

case is further increased through the influence of the corresponding knowledge node(s).

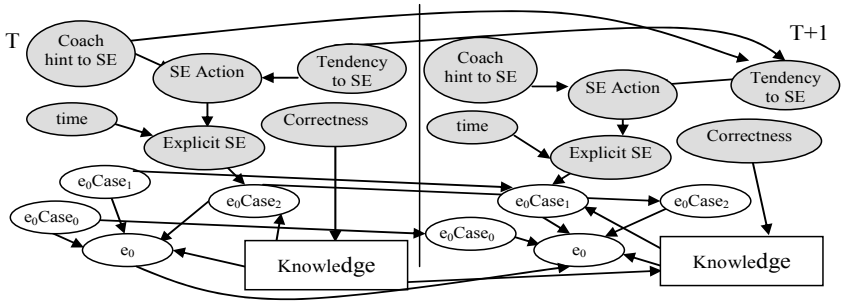

Fig. 5. Nodes Related to Explicit Self-Explanations Actions

5.3 Assessing Self-Explanation Tendency

One novel aspect of ACE's student model is its ability to assess how a student's tendency to self-explain evolves during the interaction with the system. In particular, the model currently represents the finding that explicit coaching can improve SE tendency [2]. Fig. 5 shows how the model assesses this tendency in the explicit SE slices.

This assessment consists of two stages. In the first stage, represented by the slice created in response to a student's explicit SE action (slice T in Fig. 5), evidence of a Coach hint to self-explain allows the model to refine its assessment of the student's SE tendency *before* s/he performed the SE action. The CPT for a *SE Action* node is designed so that the amount of credit for the SE action that goes to the student's *SE Tendency* in slice T depends upon whether the action was preceded by a hint. The occurrence of such a hint *explains away* much of the evidence, limiting the increase in the belief that the student's *SE Tendency* was the cause of the SE action.

The second stage models how a student's SE tendency evolves as a result of a Coach's hint to self-explain. A *Coach hint to SE* node at time *T* is linked to a *SE tendency* node at time *T+1*, so that the probability that the tendency is high increases after the hint occurs. The magnitude of this increase is currently set to be quite conservative, but we plan to refine it by performing user studies. A similar mechanism is used to assess *SE Tendency* in implicit SE slices.

6 Sample Assessment

We now illustrate the assessment generated by our model with two sample scenarios.

Scenario 1: Explicit SE Action. Suppose a student, assessed to have a low initial knowledge and a fairly high SE tendency, is exploring an exercise in the Plot Unit. She first performs an exploratory action, and then chooses to self-explain explicitly using the SE tools. Figure 6 (Top) illustrates the model's assessment of the relevant knowledge, SE tendency, and case exploration for the first three slices of the interaction. Slice 1 shows the model's assessment prior to any student activity. Slice 2 shows the assessment after one exploratory action with medium time has been performed, but not explicitly self-explained. Since the student's SE tendency is fairly high and medium time was spent performing the action, the assessment of case exploration increases. Slice 3 shows the assessment after the student performed an explicit SE action (corresponding to the same exploration case). Since the action was performed without a Coach hint, the appraisal of her SE tendency increases in that time slice. The self-explanation action was correct, which increases knowledge of the related concept. Finally, case exploration increases in Slice 3 because 1) the learner spent enough time self-explaining and 2) has provided direct evidence of her knowledge.

Scenario 2: Insufficient Time. Let's now suppose that our student moves on to a Plot Unit exercise involving the linear function, and that she has low knowledge of this function's behaviour. Our student tries various configurations of the function in the interface, but does each action quickly, leaving little time for self-explanation. Figure 6 (Bottom) illustrates the model's assessment of the exercise and SE tendency nodes for the first three slices of this interaction. With each quick action performed by the student, the model's belief in the student having explored the exercise adequately increases very slightly to indicate that actions were performed, but were not sufficiently self-explained. This belief is based on the model's assessment of the explored case nodes, for which the belief would be low in this scenario (since each action corresponds to a different case, we did not show case probabilities in the graph to avoid cluttering the figure). After these three actions, the exercise exploration is low, but the student's tendency to self-explain remains fairly high to account for the possibility that the student will eventually engage in self-explanation. This will lead the Coach to believe that although the student has not explored the exercise well so far, s/he may do so prior to moving on to a new one. The Coach remains silent unless the student actually tries to leave the exercise without providing any further evidence of self explanation. On the other hand, had the model assessed the student's SE tendency to be low, the Coach would have intervened as soon as a lack of self-explanation was detected.

The low probability for adequate exercise exploration illustrates the difference between this version of the model and the original ACE model [3]. The old model took into account only coverage of exploratory actions, without assessing whether the student had self-explained the implications of those actions. Thus, that model would have assessed our student to have adequately explored the cases covered by her actions.

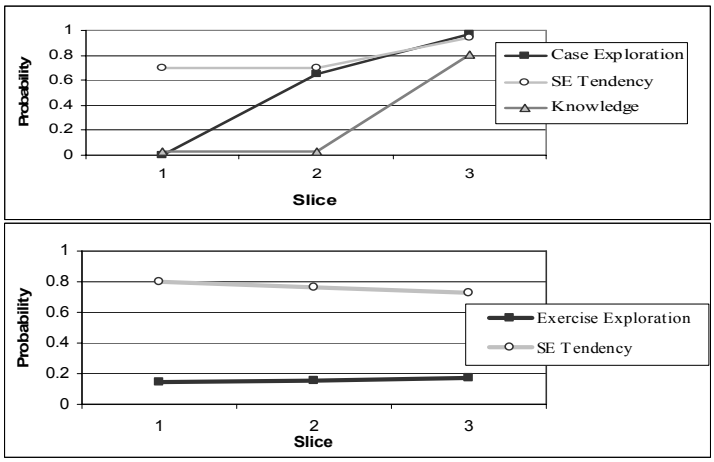

Fig. 6. Scenario 1 (Top) and Scenario 2 (Bottom)

7 Summary and Future Work

In this paper, we described how we augmented ACE, an open learning environment for mathematical functions, to model and support self-explanation during a student's exploration process. To provide this support in a timely fashion, ACE relies on a rich model of student self-explanation behaviour that includes an explicit representation of 1) a student's SE tendency and 2) how this tendency evolves as a consequence of ACE coaching. Having a framework that explicitly models the evolution of a student's SE tendency not only allows for a more accurate assessment of student behaviour, but also provides a means to empirically test hypotheses on a phenomenon whose details are still open to investigation.

The next step will involve evaluating ACE's student model and the support for self-explanation with human participants, allowing us to validate a number of assumptions currently in our model, including the role of time in assessing implicit self-explanation and the impact coach hints on self-explanation tendency. In addition, we plan to investigate the most appropriate interface options for presenting the self-explanation hints and tools. We are also examining ways to improve the assessment of implicit SE through the use of eye tracking to track students' attention.

References

1. Aleven, V. and K.R. Koedinger, An Effective Meta-cognitive Strategy: Learning by Doing and Explaining with a Computer-Based Cognitive Tutor. Cognitive Science, 2002. **26**(2): p. 147-179.
2. 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.
3. Bunt, A. and C. Conati. Probabilistic Student Modelling to Improve Exploratory Behaviour. Journal of User Modeling and User-Adapted Interaction, 2003. **13**(3): p.269-309.
4. Bunt, A., C. Conati, M. Huggett, and K. Muldner, On Improving the Effectiveness of Open Learning Environments through Tailored Support for Exploration. in AIED 2001, 10th World Conference of Artificial Intelligence and Education. 2001. San Antonio, TX.
5. Chi, M.T.H., Self-Explaining Expository Texts: The Dual Processes of Generating Inferences and Repairing Mental Models. In Advances in Instructional Psychology, 2000, p. 161-237.
6. Chi, M.T.H., M. Bassok, M. Lewis, P. Reimann and R. Glaser, Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. Cognitive Science, 1989. **15**: p. 145-182.
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. and K. VanLehn, Toward Computer-based Support of Meta-cognitive Skills: A Computational Framework to Coach Self-Explanation. International Journal of Artificial Intelligence in Education, 2000. **11**.
9. Conati, C. and K. VanLehn. Providing Adaptive Support to the Understanding of Instructional Material. in IUI 2001, International Conference on Intelligent User Interfaces. 2001. Santa Fe, NM, USA.
10. Mitrovic, A. Supporting Self-Explanation in a Data Normalization Tutor. in Supplementary proceedings, AIED 2003. 2003.
11. Shute, V.J. and R. Glaser, A Large-Scale Evaluation of an Intelligent Discovery World. Interactive Learning Environments, 1990. **1**: p. 51-76.
12. van Joolingen, W.R. and T. de Jong, Supporting Hypothesis Generation by Learners Exploring an Interactive Computer Simulation. Instructional Science, 1991. **20**: p. 389-404.