

Toward Meta-cognitive Tutoring: A Model of Help-Seeking with a Cognitive Tutor

Vincent Aleven, Bruce McLaren, Ido Roll, Kenneth Koedinger, *Human-Computer Interaction Institute, Carnegie Mellon University*
{aleven, bmclaren}@cs.cmu.edu, {idoroll, koedinger}@cmu.edu

Abstract. The research reported in this paper focuses on the hypothesis that an intelligent tutoring system that provides guidance with respect to students' meta-cognitive abilities can help them to become better learners. Our strategy is to extend a Cognitive Tutor (Anderson, Corbett, Koedinger, & Pelletier, 1995) so that it not only helps students acquire domain-specific skills, but also develop better general help-seeking strategies. In developing the Help Tutor, we used the same Cognitive Tutor technology at the meta-cognitive level that has been proven to be very effective at the cognitive level. A key challenge is to develop a model of how students should use a Cognitive Tutor's help facilities. We created a preliminary model, implemented by 57 production rules that capture both effective and ineffective help-seeking behavior. As a first test of the model's efficacy, we used it off-line to evaluate students' help-seeking behavior in an existing data set of student-tutor interactions. We then refined the model based on the results of this analysis. Finally, we conducted a pilot study with the Help Tutor involving four students. During one session, we saw a statistically significant reduction in students' meta-cognitive error rate, as determined by the Help Tutor's model. These preliminary results inspire confidence as we gear up for a larger-scale controlled experiment to evaluate whether tutoring on help seeking has a positive effect on students' learning outcomes.

Keywords. Meta-cognition, cognitive modeling, help-seeking, tutor agents, educational log file analysis

INTRODUCTION

Meta-cognition is an important area of focus for researchers in the learning sciences. In a recent influential volume entitled *How People Learn*, in which leading researchers survey state-of-the-art research on learning and education (Bransford, Brown, & Cocking, 2000), one of three broad recommendations was to focus on improving students' meta-cognitive skills. A number of classroom studies have shown that instructional programs with a strong focus on meta-cognition can improve students' learning outcomes (Brown & Campione, 1996; Palincsar & Brown, 1984; White & Frederiksen, 1998). An important question therefore is whether intelligent tutoring systems can be effective in supporting meta-cognitive skills. A small number of studies have shown that indeed they can. For example, it has been shown that self-explanation, an important meta-cognitive skill, can be supported with a positive effect on the learning of domain-specific skills and knowledge (Aleven & Koedinger, 2002; Bunt, Conati, & Muldner, 2004; Conati & VanLehn, 2000; Mitrovic, 2003; Renkl, 2002; Trafton & Trickett, 2001). Another example is the work of Gama who showed advantages of having students self-assess their skill level (2004).

This paper focuses on a different meta-cognitive skill: help-seeking, that is, the ability to solicit help when needed from a teacher, peer, textbook, manual, on-line help system, or the Internet. Help-seeking has been studied extensively in social contexts such as classrooms (see e.g., Karabenick, 1998). In the classroom, it is considered an important self-regulatory strategy that, rather than signalling a student's dependence, can be instrumental in developing independent skill and ability (e.g., Nelson-LeGall, 1981; Newman, 1994; 2002). An important finding of this line of research is that those who need help the most are the least likely to ask for it (Karabenick & Knapp, 1988a; Puustinen, 1998; Ryan, Gheen, & Midgley, 1998).

Help-seeking has been studied to a lesser degree in the context of interactive learning environments. Given that many learning environments provide some form of on-demand help, it might seem that effective help use would be an important factor influencing the learning results obtained with these systems. However, there is evidence that students tend not to effectively use the help facilities offered by learning environments (for an overview, see Aleven, Stahl, Schworm, Fischer & Wallace, 2003). They often ignore the help facilities or use them in ways that are not likely to help learning. They frequently use the system's on-demand hints to get answers, without trying to understand how the answers are derived or the reasons behind the answers (Aleven & Koedinger, 2000). On the other hand, there is also evidence that when used appropriately, on-demand help in an interactive learning environment can have a positive impact on performance (Aleven & Koedinger, 2000; Bartholomé, Stahl, Pieschl, & Bromme, 2006) and learning (Renkl, 2000; Schworm & Renkl, 2002; Wood, 2001; Wood & Wood, 1999). Furthermore, there is evidence that different types of help (Dutke & Reimer, 2000) or feedback (McKendree, 1990; Arroyo et al., 2001) affect learning differently.

The research discussed in this paper focuses on the hypothesis that providing tutoring with respect to students' help-seeking behavior helps them to become better help seekers and, thus, better future learners (see e.g., Bransford & Schwartz, 2001). We know of no studies that have investigated whether students' learning gains can be improved through help-seeking instruction within computer-based tutoring, with the exception of a study by Luckin and Hammerton (2002), who report encouraging early evidence that meta-cognitive scaffolding can help learners and, in particular, the less able learners. However, no controlled experiment was conducted to evaluate the effect of such scaffolding. Nor do we know of any studies investigating the effectiveness of instruction on help-seeking outside the context of interactive learning environments. The work on teaching self-questioning strategies, reviewed in Rosenshine, Meister, and Chapman (1996), is perhaps the closest.

We are testing the effectiveness of instruction on help-seeking in the context of a Cognitive Tutor. This type of intelligent tutor is designed to support *learning by doing* and features a cognitive model of the targeted skills, expressed as production rules (Anderson et al., 1995). Cognitive Tutors for high-school mathematics have been highly successful in raising students' test scores (Koedinger, Anderson, Hadley, & Mark, 1997) and are being used in over 2000 schools across the United States. To address our research question, we have added a Help Tutor agent, a Cognitive Tutor in its own right that provides feedback to students on the way they use the tutor's help facilities. Our approach to tutoring help-seeking contrasts with a previous approach to overcoming some of the barriers to effective help-seeking, namely, a multi-agent brokerage system that uses student models to match students in need of help with peer helpers (Bull, Greer, & McCalla, 2003). We view the two approaches as largely complementary – even

with a brokerage system, students must still make decisions about when to seek help, and tutoring is likely to help in that regard.

In order for the Help Tutor agent to provide meaningful and appropriate feedback on students' help-seeking behavior, it must have an accurate model of adaptive help-seeking behavior. This model must be precise and detailed enough to be executed on a computer and to make predictions about whether a particular student should seek help at each step of solving a tutor problem. Following Cognitive Tutor design principles (Anderson *et al.*, 1995), we have implemented the model in the form of production rules. Identifying such a model is a contribution in its own right, since there are no standards, guidelines, or published cognitive task analyses that we can draw upon. A number of models of help-seeking have been published in the literature (e.g., Gross & McMullen, 1983; Nelson-LeGall, 1981; Newman, 1994). Typically, such models identify a number of decision steps in the help-seeking process as well as factors that influence these decisions. For example, the decision to seek help may be influenced by factors such as the perceived threat to one's self-esteem, fear of being seen as incompetent, and a reluctance to be in debt to the helper (Gross & McMullen, 1983). However, the existing models focus on help-seeking in social situations and, most importantly, they do not provide the specifics necessary for a computational model. Compared to the model presented in this paper, they cover a wider range of help-seeking situations and factors that influence help-seeking, but do so in significantly less detail.

Developing a computational model of help-seeking is a challenging task. Viewed superficially it might seem clear what judicious and deliberate use of a tutor's help facilities entails. A student should ask for help when she needs it and only then. For example, a request for help is appropriate when a student is stuck solving a tutor problem but not when she has not yet thought about the problem. Further, students should carefully read and interpret the help given by the system. But exactly how can these general requirements be operationalized in a cognitive model? Adding to the challenge is the fact that it can be difficult to tell whether particular help-seeking behavior is productive or not. For example, Wood and Wood (1999) describe an example of a student who, by all reasonable measures, appeared to be over-using the tutor's help facilities, yet ended up among the students with the highest learning gains.

Our approach has been first to develop a model based on our intuitions, formed over many years of developing Cognitive Tutors and observing students who use them. We then validated the model by running it off-line against an existing data set of student-tutor interactions. We used the results of this analysis to implement various refinements to the model. We then tried out the Help Tutor in a small-scale pilot study, which involved 4 students, and refined the model further based on the experience gained in this study. This paper discusses the development of the model and presents encouraging results from the pilot study.

INITIAL TEST BED: THE GEOMETRY COGNITIVE TUTOR

Although our model of help-seeking behavior is designed to work with any Cognitive Tutor, and possibly other intelligent tutors as well, we are initially testing it within the Geometry Cognitive Tutor, shown in Figure 1. This tutor was developed in our lab together with a full-year geometry high-school curriculum. The curriculum aims at engaging students in active learning-by-doing through problem solving. In courses based on this curriculum, students work with the Cognitive

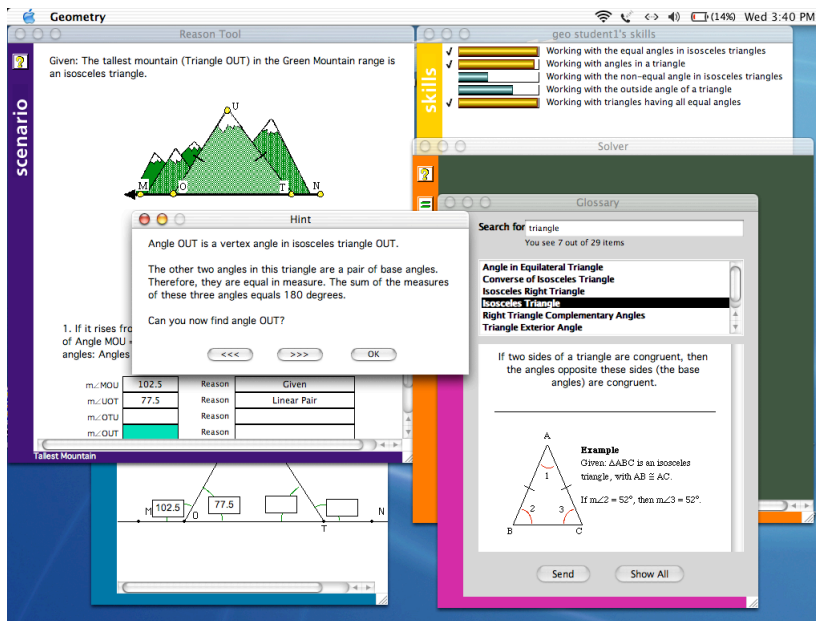


Fig. 1. The Geometry Cognitive Tutor.

Tutor in a computer lab two class periods a week. The software tutor poses complex, real-world problems and provides just-in-time, individualized feedback and on-demand advice to support students in solving the problems. Typically, students are allowed to work at their own pace. Some teachers give credit for each completed tutor unit. During the three class periods each week that are not devoted to tutor use, students work in the classroom with the course textbook, which supports whole-class instruction and small-group problem solving. The Cognitive Tutor curricula have been very successful in raising students' test scores, compared to more traditional math curricula. In multiple studies, students in Cognitive Tutor Algebra and Geometry courses scored higher on end-of-course objective tests and open-ended problem solving tests than students enrolled in standard courses (Koedinger *et al.*, 1997; Koedinger, Corbett, Ritter, & Shapiro, 2000). At the time of this writing, the Cognitive Tutor Geometry curriculum is in routine use in 350 schools across the United States. The Cognitive Tutor Algebra curriculum, which is based on the same principles, is in use in approximately 2000 schools.

Like other Cognitive Tutors, the Geometry Cognitive Tutor uses a cognitive model of the skills to be learned, expressed in the form of production rules. The tutor uses the model to follow students step by step through their (sometimes idiosyncratic) approach to each problem. It uses an algorithm called *model tracing* to evaluate the student's solution steps and provide feedback (Anderson *et al.*, 1995). After each problem-solving action by the student, the tutor evaluates the action and provides feedback. It searches the problem space defined by the model's production rules, generating possible next steps in the solution of the problem. If the student's step is among the predicted steps, the tutor accepts it as correct. Otherwise, the step is marked as incorrect. When the step is accepted as correct, the tutor updates the working memory used by its production system, by firing the rule or rules that led to the predicted action. When the step is identified as incorrect, by either falling outside of the model's predictions altogether or by

Table 1
Sample hint sequence of the Geometry Cognitive Tutor

1. As you can see in the diagram, Angles UOT and MOU are adjacent angles. Together they form segment MT. How can you use this fact to find the measure of Angle UOT?
2. Look in the Glossary for reasons dealing with adjacent angles.
3. Some rules dealing with adjacent angles are highlighted in the Glossary. Which of these reasons is appropriate? You can click on each reason in the Glossary to find out more.
4. The sum of the measures of a linear pair of angles is 180° . Angle UOT and Angle MOU form a linear pair.
5. The sum of the measures of angles UOT and MOU is equal to 180 degrees.
6. The measure of Angle UOT is equal to 180 degrees minus the measure Angle MOU.
7. $m\angle UOT = 180^\circ - m\angle MOU$.

matching the prediction of a “bug rule,” working memory is not updated. Thus, working memory always mirrors the state of problem solving in the student interface, a necessary condition for the model-tracing algorithm to be able to follow the student step-by-step through the problem.

The Geometry Cognitive Tutor offers two types of on-demand help, which are quite typical of intelligent tutoring systems: context-sensitive hints and a context-insensitive Glossary. At the student’s request, the tutor provides context-sensitive help at multiple levels of detail (see the pop up window in the middle of Figure 1). This multi-level help is tailored toward the student’s specific goal within the problem at hand, with each hint providing increasingly more specific advice. In the current version of the tutor, students can move through the hint levels both in forward and backward direction. In the tutor version used in the off-line evaluation of the help-seeking model described below, however, students could only move forward. Only one level is displayed at any given time. That is, when a new hint level is displayed, it replaces the previous level. (The specifics of the tutor’s hint display mechanism will turn out to be important later on when we look at students’ tendencies to click through the hint levels.) As illustrated in Table 1, the hints typically identify a problem-solving principle to be used (a geometry theorem or definition) and explain how it applies to the problem at hand (levels 4-6 in Table 1). The last hint in each sequence (dubbed the “bottom-out hint”) comes very close to providing the answer. In the Angles unit, which deals with the geometric properties of angles and is the focus of the study reported in this paper, the bottom-out hint provides an arithmetic expression that can be evaluated to find the quantity sought (e.g., “ $m\angle UOT = 180^\circ - m\angle MOU$ ”). In order to determine the hint to provide to the student, the tutor first runs its cognitive model to generate possible next problem-solving steps. It then selects one of them as the preferred next step and collects hint templates that are attached to the matching production rules. Finally, it fills out the templates with problem-specific information collected during the matching of the rules, such as the names of the quantities (e.g., “ $m\angle UOT$ ”).

The Geometry Cognitive Tutor also provides a second source of help, a Glossary, shown at the bottom-right of Figure 1. Unlike hints, the information in the Glossary is not tailored to the user’s goals. Rather, at the student’s request, the Glossary displays information about a selected

geometry rule (i.e., a theorem or definition): a statement of the rule plus a simple example diagram illustrating the use of the rule in a problem-solving situation. It is up to the student to search for relevant rules and to decide which rule is applicable. In order to encourage the students to use the Glossary, we added hints to the tutor's hint sequences that identify a crucial feature of the problem (often, a kind of geometric configuration like "adjacent angles") and recommend that students search the Glossary for rules dealing with that kind of configuration or feature (see hint levels 1-3 in Table 1). The Glossary was added not just because it is likely to help students learn geometry better, but also because it gives students an opportunity to gain experience with a source of help that is like many others found in the real world, where information often does not come tailored to one's particular problem. For example, reference manuals or Internet searches typically enable the learner to search for relevant information in a larger body of knowledge. We view the use of a resource like the Glossary as being part of *doing* geometry (and math and science more generally), not just of *learning* geometry. The goal of instruction in geometry should not be merely for students to develop fluency with a particular set of geometry theorems but also to deal with unfamiliar theorems found in reference materials like the Glossary. Such a viewpoint is in line with current theories about transfer (e.g., Bransford & Schwarz, 2001; Schwartz & Taylor, 2004), although not necessarily with standardized tests, which rarely provide novel definitions for students to read and apply. Therefore, we try to get students to use the Glossary but try not to focus them exclusively on Glossary use.

Cognitive Tutors keep track of a student's knowledge growth over time by means of a Bayesian algorithm called *knowledge tracing* (Corbett & Anderson, 1995). At each problem-solving step, the tutor updates its estimates of the probability that the student knows the skills involved in that step, based on whether the student was able to complete the step without errors and hints. The probabilities of skill mastery are displayed in the skill meter window, shown at the top right in Figure 1. Skills for which the probability is .95 or greater are considered to be mastered and are shown in gold with a tick mark. A Cognitive Tutor uses the estimates of skill mastery to implement a mastery-learning regime, selecting problems that involve un-mastered skills until the student has mastered all skills. Thus, students tend to pay attention to their "skill bars" displayed in the skill meter window and watch with satisfaction as their skill bars turn gold. The knowledge mastery estimates also play a role in the model of help-seeking, presented below.

A MODEL OF DESIRED HELP-SEEKING BEHAVIOR

Design

We have developed a preliminary model of adaptive help-seeking behavior with a Cognitive Tutor. The model specifies how a student ideally would use the help facilities of a Cognitive Tutor, namely, multi-level context-sensitive hints and a Glossary. The model forms the basis of the Help Tutor agent, which serves as an adjunct to a Cognitive Tutor. That is, the Cognitive Tutor provides tutoring at the cognitive level (i.e., with respect to domain-specific skills and knowledge), the Help Tutor provides tutoring at the meta-cognitive level, with respect to the student's help-seeking behavior. The Help Tutor agent is a Cognitive Tutor in its own right. It uses its model of help-seeking behavior in the same way that a regular Cognitive Tutor uses its model of domain-specific skills and knowledge, namely, to track students' problem-solving steps

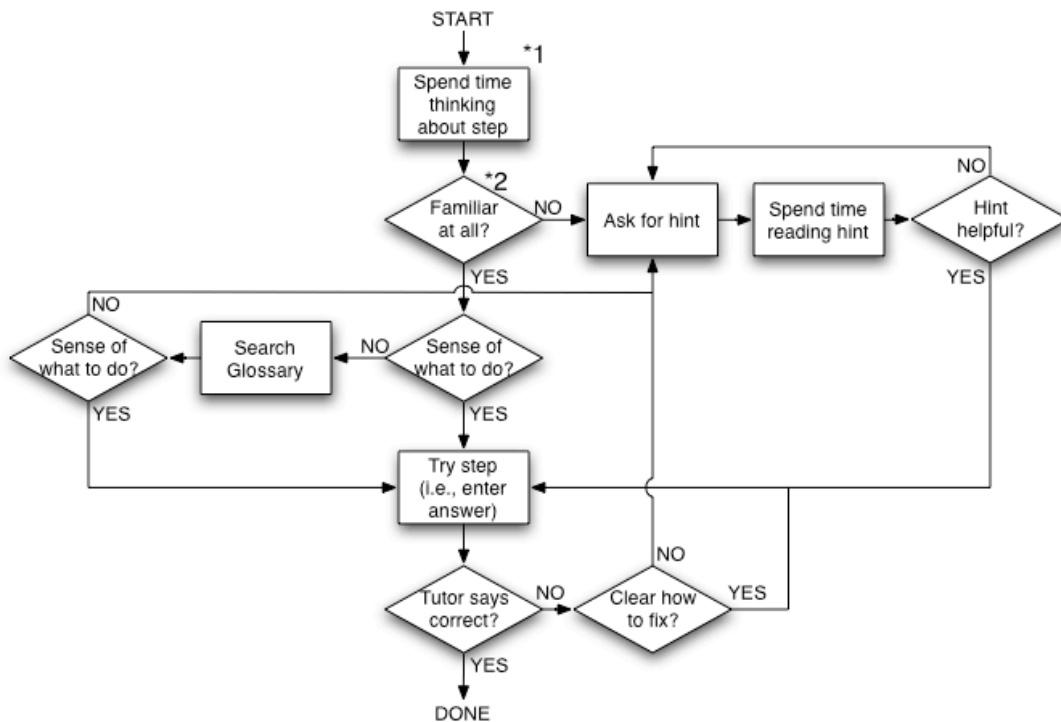


Fig.2. A model of desired help-seeking behavior. The asterisks “*” indicate examples of where violations of the model can occur, discussed later in the paper.

and provide feedback and guidance. The initial design of the model is shown in Figure 2. The model was modified later, based on its evaluation against data of student-tutor interactions and on the comments from students who participated in a small-scale pilot. The changes are described in a later section of the paper.

The model is based on our experience and intuition with intelligent tutors – in particular, the Geometry Cognitive Tutor – and shares some general traits with models of social help-seeking put forward by Nelson-LeGall (1981), Gross and McMullen (1983), and Newman (1994), a point that is discussed further below. We believe the model is a contribution to the literature on help-seeking because it is far more fine-grained than existing models, which are not computational in nature. In addition, to the best of our knowledge, it is the first help-seeking model that has been computationally realized in an interactive learning environment. However, the current model is not as broad as existing models, which are intended to cover a broad range of help-seeking situations. The current model captures the use (as well as the non-use) of the kinds of help facilities that are typical of intelligent tutoring systems. We expect the model will eventually clarify poorly understood relations between help-seeking and learning.

According to the model, the decision to seek help, when working on a step in a tutor problem, depends to a large degree on how well the skills involved in the step are mastered. To summarize the model briefly, if the step looks familiar and the student has a sense of what to do,

she should try the step without seeking help. Otherwise, she should hold off and use an appropriate source of help first: the Glossary if the step is at least somewhat familiar, context-sensitive hints if the student is new to the given kind of step. More specifically, according to the model, the ideal student behaves as follows: If, after spending some time thinking about a problem-solving step, the step does not look familiar, the student should ask the tutor for a hint, rather than rely on guessing strategies and tutor feedback to get the answer right. After reading the hint carefully, she should decide whether the hint was helpful, that is, whether it is now clear how to solve the step. If so, she should go ahead and try it. If not, she should request a more detailed hint. Although this is not indicated in the flow chart of Figure 2, if a student is not able to solve a step after exhausting all sources of help, the model prescribes that she ask the teacher. If the step looks familiar from the start, but the student does not have a clear sense of what to do, she should try to use the Glossary to find out more. Thus, the model steers students towards the Glossary in situations where they are likely to have some level of knowledge that they can bring to bear in deciding whether information found is helpful. If the student does have a sense of what to do, she should try to solve the step. If the tutor feedback indicates that the step is incorrect, the student should ask for a hint unless it was clear how to fix the problem. In general, the student should carefully deliberate on her actions (i.e., take time to make a decision) before deciding on the next move.

The strategy for deciding between hints and the Glossary balances two concerns: on the one hand, context-sensitive hints provide tailor-made help and thus make the student's job easier, compared to the Glossary. As mentioned, effective Glossary use requires search and judgments about the relevance of what was found. Thus, for students faced with an unfamiliar situation, context-sensitive hints may be more effective. On the other hand, we want students to develop skill in working with the Glossary, since we view the use of this type of resource as an integral part of doing math or science. Given that Glossary use is more challenging than hint use and may lead to higher cognitive load, it makes sense to prescribe context-sensitive hints in unfamiliar situations and Glossary use in situations where students have at least some minimal amount of knowledge.

The recommendation that students ask for a hint in unfamiliar situations is based on the assumption that the students come to the tutor with sufficient background knowledge to learn from the tutor's hints. In the Angles unit, for example, it is assumed that students know basic concepts such as angles and angle measures. The tutor's hints will not tell students what angles are nor that angles have measures whose unit is degrees. Such an assumption is reasonable in standard Cognitive Tutor courses, which involve not just tutor use but also classroom and small-group instruction, aimed at helping students learn basic concepts. Under this assumption it makes sense for students to request a hint from the tutor when a step is not familiar, rather than ask the teacher right away. The recommendation that students request a more specific hint level only if the previous level did not help, rather than, for example, follow every hint sequence till the end, is based on the assumption that it is a good thing if students get only as much help as needed and generate as much of the necessary information for themselves as they can (see e.g., Anderson, 1993, p.241).

For implementation, we had to refine and make concrete some of the abstract elements of the flowchart. Our strategy was to try simple approaches first and to expend the effort to create something sophisticated only after it was clear that a simpler approach would not work. For example, to implement the tests *Familiar at all?* and *Sense of what to do?*, which are essentially

self-monitoring actions by the student, we needed a way to assess how well a particular student knows a particular skill at a particular point in time. Item response theory (Hambleton & Swaminathan, 1985) is not well suited, since it does not track the effect of learning over time. Instead, we use the estimated probabilities of an individual student's skill mastery derived by the Cognitive Tutor's knowledge-tracing algorithm. As mentioned above, by means of this algorithm, the tutor maintains up-to-date estimates of the probability that the student knows each skill targeted in the instruction. The tests *Familiar at all?* and *Sense of what to do?* compare these probability estimates against pre-defined thresholds. If the probability of skill mastery is above .3, the test *Familiar at all?* succeeds. Similarly, the test *Sense of what to do?* succeeds if the probability of skill mastery is above .6. These values are intuitively plausible, given our past experience with Cognitive Tutors, but need to be validated empirically. One of the goals of our experiments with the model, described below, is to evaluate and refine the thresholds, and to determine whether fixed thresholds are a viable approach.

The tests *Clear how to fix?* and *Hint helpful?* also had to be rendered more concrete. For the *Clear how to fix?* test, the help-seeking model prescribes that a student with a higher probability of having mastered the skill involved in the step, should re-try a step after missing it once, but that mid or lower skilled students, according to the probability estimates, should ask for a hint. This is clearly a simplification, since the distinction between errors that are easy to fix and errors that are hard to fix hinges not only on the student's skill level, but also on the particular error made. However, it may be a practically useful simplification. In the future we plan to elaborate the *Clear how to fix?* test by using heuristics that catch some of the common types of easy-to-fix slips that students make. Our implementation of *Hint Helpful?* assumes that the amount of help a student needs on a particular step depends on the probability that she has mastered the skill(s) involved in that step. Thus, a high-skill student, after requesting a first hint, is predicted to need 1/3 of the available hint levels, a mid-skill student 2/3 of the hints, and a low-skill student *all* of the hints. Again, our initial implementation of this test is a simplification. Whether or not a hint is helpful may depend not only on the given student's mastery of the skill that is explained in the hints, but also on their (mathematical) reading comprehension ability. The step *Spend time reading hint*, finally, was implemented using a fixed threshold of 8 seconds.

The model presented here has some high-level overlap with models focused on help-seeking in social contexts, such as that one presented by Nelson-LeGall (1981) and later elaborated by Newman (1994). Their model has the following steps:

1. Become aware of a need for help
2. Decide to seek help
3. Identify potential helper(s)
4. Use strategies to elicit help
5. Evaluate help-seeking episode

For each step, Nelson-LeGall and Newman elaborate a number of factors that influence the step. Thus, these models are primarily descriptive, whereas our model is both prescriptive and descriptive. It is prescriptive in the sense that it is meant to capture good help-seeking behavior that, it is hypothesized, is beneficial for students to learn. It is descriptive in the sense that it also captures actual help-seeking behavior that is hypothesized to be ineffective, in the form of "bug rules," described further below. The bug rules enable the tutor to comment on less than optimal help-seeking behavior.

In the Nelson-LeGall/Newman model, the student must first become aware that she needs help, for example, by assessing task difficulty, monitoring task progress, or evaluating her own comprehension or skill. In the next step, she must consider all available information and decide whether to seek help. This decision may involve a range of factors besides self-assessment of progress or skill, such as threats to self-esteem, fear of embarrassment, reluctance to be indebted to the helper, etc. In the next step, the learner must find a suitable helper. In most classrooms, the teacher or a fellow student could serve this role. In a social context, the choice of helper may depend on the age of the learner and on the perceived competence and sensitivity of the helper (Nelson-LeGall, 1981; Knapp & Karabenick, 1988). Next, the student must decide how to request help, based on her knowledge and skills of discourse (Newman, 1998). Essentially, the request must match the task demands. Finally, the student reflects upon the help-seeking event to decide if it was helpful and to determine whether further help is required.

Our computational model, presented in Figure 2, is of course narrower in scope: it is meant to cover help-seeking with a Cognitive Tutor rather than all of social help-seeking. Our model however, provides far greater detail and specificity, as is needed for a model that is to be run on a computer. The model covers all steps of the Nelson-LeGall/Newman model. With respect to the first two steps, in our model, as in the Nelson-LeGall/Newman model, the decision to seek help depends heavily on self-monitoring and/or self-assessment by the student, captured by the tests *Familiar at all?* and *Sense of what to do?*. In addition, feedback from the tutor may help students become aware of their need for help, captured in the test *Tutor says correct?*. With respect to Step 3 of the Nelson-LeGall/Newman model (Identify potential helper(s)), in our model the students have two potential choices: context-sensitive hints and the Glossary. The factors that should influence this choice have been described above. It is interesting to note that a number of social factors that may negatively influence the decision to seek help, such as fear of embarrassment or reluctance to be indebted to the helper, may not come into play in the context of computer tutors, quite possibly an important advantage of computer tutors (see e.g., Karabenick & Knapp, 1988b).

In step 4 (Using strategies to elicit help) the model requires that students give sufficient attention (modeled as time) to the hints and ask only for as many hints as needed. The model does not capture glossary-usage strategies, however. When using the Glossary, students must decide what search term to use and how to refine their search if it turns up too few or too many items. Currently, this type of strategy is not modeled. All the model does in situations where Glossary use is appropriate (step *Search Glossary*) is put a cap on the number of Glossary items that the student may look at, on the assumption that a productive search strategy does not lead to an extended search process. A student who inspects many Glossary items without finding what she is looking for is probably better off asking for a hint. In the future, we plan to elaborate the model to capture search strategies in greater detail, so that the Help Tutor can help students learn to search the Glossary effectively. Finally, with respect to step 5 (Evaluate help-seeking episode), our model prescribes that students evaluate the usefulness of hints (*Hint helpful?*) before deciding whether or not to request the next hint level.

In addition to covering most of the steps of the Nelson-LeGall/Newman model, our model prescribes that the student works deliberately, captured in the steps *Spend time thinking about step* and *Spend time reading hint*. While an emphasis on avoiding haste and guessing perhaps takes us outside the realm of help-seeking per se, such meta-cognitive behaviors are clearly important when working with a computer tutor.

Implementation

We have implemented an initial version of the help-seeking model of Figure 2. (As mentioned, subsequent modifications to the model are described later in the paper.) The current model consists of 57 production rules, which capture both correct (or adaptive) and incorrect (or maladaptive) help-seeking behavior. Thirty-two of the rules are bug rules, which reflect deviations from the ideal help-seeking behavior (or “meta-cognitive bugs”). These rules enable the Help Tutor agent to provide feedback to students when their help-seeking behavior is less than ideal. In deciding what action to take next (i.e., try the step, request a hint, or inspect a Glossary item), three key pieces of information are considered:

- (1) The time needed for deliberate action,
- (2) The skill(s) involved in the step and the probability that the student masters this skill(s)
- (3) What the student has already done with respect to this step (e.g., the number of previous – and unsuccessful – attempts and the number of hints).

This information enables the model to generate predictions about the correct meta-cognitive behavior: what type of action to take (try the step, request hint, or inspect a Glossary item) and how much time should – at minimum – be spent on that action, in order to carry it out deliberately. The student’s help-seeking behavior is evaluated by comparing it against these predictions. If the student’s action matches one of the behaviors generated by the “good rules,” the Help Tutor remains silent. If the student’s action matches one of the bug rules, however, an error message associated with the bug rule is printed. In designing the help-seeking model, we have made the assumption that if the student gets the answer right at the cognitive level (as determined by the Cognitive Tutor), the Help Tutor should not complain even if the action seemed suboptimal at the meta-cognitive level, for example, if it was carried out too fast or if help use would have been more appropriate. Negative meta-cognitive feedback immediately following a correct action is probably not a good idea.

As an example of how the model operates, let us consider a student faced with an unfamiliar problem-solving step in a tutor problem. Without spending much time thinking about the step, she ventures an answer and gets it wrong. In doing so, the student deviates from the help-seeking model in two ways: she does not spend enough time thinking about the step (a meta-cognitive error marked as “*1” in Figure 2) and in spite of the fact that the step is not familiar to her, she does not ask for a hint (marked as “*2”). The students’ errors match bug rules that capture these forms of unproductive help-seeking behavior, allowing the tutor to provide feedback.

Figure 3 shows the tree of rules explored by the model-tracing algorithm as it searched for rules matching the student’s help-seeking behavior (or in this situation, lack thereof). A number of paths in the tree contain applicable rules that did not match the student’s behavior (marked with “***”), including a rule that represents the “right” meta-cognitive behavior in the given situation (“think-about-step-deliberately”). This rule implements the step *Spend time thinking about step*, shown in the flow chart of Figure 2. The prediction made by this rule with respect to the amount of time needed for the deliberate execution of the step did not match the student’s behavior. Therefore, the branch containing this rule was abandoned during the search. If it had been explored further, it would, for example, have contained the rule “ask-hint-due-to-low-skill.”

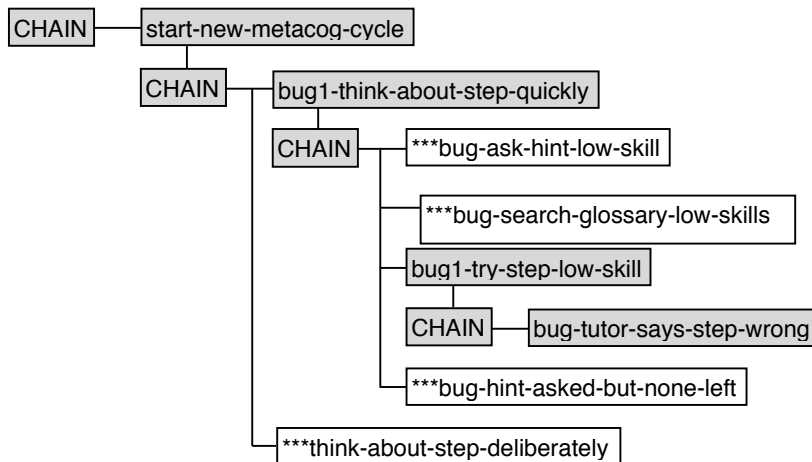


Fig.3. Space of rules explored by the Help Tutor as it tries to find a rule sequence that predicts the student's help-seeking behavior in the example described in the text, namely, to try an unfamiliar step quickly without help use.

The conditions of this rule test that the student's skill level is below the threshold discussed above (i.e., this rule implements the test *Familiar at all?*)

The rule chain that matched the students' behavior is highlighted. This chain includes an initial rule that starts the meta-cognitive cycle ("start-new-metacog-cycle"), a subsequent bug rule that represents the meta-cognitive error of acting too quickly ("bug1-think-about-step-quickly"), a second bug rule that represents the meta-cognitive error of trying the step, given her low mastery of the skill at that point in time ("bug1-try-step-low-skill"), and, finally, a rule that reflects the fact that the student answered incorrectly ("bug-tutor-says-step-wrong"). The fact that the answer was incorrect is not in itself a meta-cognitive bug; this rule merely confirms that the other meta-cognitive errors should be commented upon. The feedback message in this case, compiled from the two bug rules identified in the chain, is: "Slow down, slow down. No need to rush. Perhaps you should ask for a hint, as this step might be a bit difficult for you." The bug rules corresponding to the student acting too quickly and trying the step when they should not have are shown in Table 2.

The help-seeking model uses information passed from the cognitive model to perform its reasoning. For instance, the skill involved in a particular step, the estimated probability that a particular student masters that skill, the number of hints available for that step, and whether or not the student got the step right, are passed from the cognitive to the meta-cognitive model. Meta-cognitive model tracing takes place after cognitive model tracing. In other words, when a student enters a value to the tutor, that value is evaluated at the cognitive level before it is evaluated at the meta-cognitive level.

An important consideration in the development of the Help Tutor was to make it modular and useable in conjunction with a variety of Cognitive Tutors, that is, to make it a plug-in tutor agent (Rich, Lesh, Rickel, & Garland, 2002; Ritter, 1997) applicable to a range of Cognitive Tutors with limited customization. We have attempted to create rules that are applicable to any Cognitive Tutor, not to a specific tutor. However, there will be some need for customization, as

Table 2
Example bug rules matching unproductive help-seeking behavior

<p>Rule: Bug1-think-about-step-quickly If the student is engaged in a meta-cognitive problem And the current subgoal is to think about the step And the student spent less than <i>min-thinking-time</i> to think about the step Then Remove the subgoal (next subgoal is to decide what action to take)</p> <p>Bug message: "Slow down, slow down. No need to rush."</p>	<p>Rule: Bug1-try-step-low-skill If the student is engaged in a meta-cognitive problem And the current subgoal is to decide what action to take And the students' estimated mastery level for the skill involved in the current step is less than <i>min-familiarity-level</i> And the student has not seen all the hints yet for the current step Then Try-Step Set a subgoal to evaluate the result</p> <p>Bug message: "Perhaps you should ask for a hint, as this step might be a bit difficult for you."</p>
--	---

optional supporting tools (of which the Glossary is but one example) are available in some tutors and not others.

A TAXONOMY OF HELP-SEEKING BUGS

In order to compare students' help-seeking behavior against the model, we have created a taxonomy of errors (or bugs) in students' help-seeking behavior, shown in Figure 4. The taxonomy includes four main categories. First, the *Help Abuse* category covers situations in which the student misuses the help facilities provided by the Cognitive Tutor or uses them unnecessarily. Recall from the flow chart in Figure 2 that a student with a high probability of mastery for the skill in question should first try the step, a student with medium mastery should use the Glossary, and a student with low mastery should ask for a hint. Also, students must read hints carefully. Thus, we distinguish the following subcategories of *Help Abuse*: spending too little time with a hint – that is, moving to the next hint level without spending sufficient time to read the current hint level (*Clicking Through Hints*), requesting a hint (after some deliberation) when knowledgeable enough to either try the step (*Ask Hint when Skilled Enough to Try-step*) or use the Glossary (*Ask Hint when Skilled Enough to Use Glossary*), or using the Glossary when skilled enough to try the step (*Glossary Abuse*).

Second, situations in which the student could benefit from asking for a hint or inspecting the Glossary, but chose to try the step instead, are categorized as *Help Avoidance*. There are three bugs of this type, all of which involve trying a step when it is not familiar, that is, without sufficient mastery of the skill involved. When the student's estimated skill mastery is low, or after repeated errors, we distinguish between situations in which the step was tried quickly (*Guess Quickly*) and those in which the student spent a sufficient amount of time (*Try Unfamiliar*

Step Without Hint Use). Situations in which the student's skill mastery is medium, and hence where the Glossary should have been consulted, is covered by *Try Vaguely Familiar Step Without Glossary Use*.

Third, the category *Try-Step Abuse* represents situations in which the student attempts to hastily solve a step and gets it wrong, when sufficiently skilled to try the step (*Try-Step Too Fast*). Finally, the category of *Miscellaneous Bugs* covers situations not represented in the other high-level categories. The *Read Problem Too Fast* error describes hasty reading of the question, when first encountered followed by a rapid help request. *Ask for Help Too Fast* describes a similar situation in which the student asks for help too quickly after making an error. The *Used All Hints and Still Failing* bug represents situations in which the student has seen all of the hints, yet cannot solve the step (i.e., the student has failed more than a threshold number of times). In our implemented model, the student is advised to talk to the teacher in this situation. In general, if the student gets the step right at the cognitive level, we do not consider a meta-cognitive bug to have occurred, regardless of whether the step was hasty or the student's skill level was inappropriate.

COMPARING THE MODEL TO STUDENTS' ACTUAL META-COGNITIVE BEHAVIOR

We conducted an empirical analysis with two inter-related goals: (a) to get a sense of how close the preliminary model was to being usable in a tutoring context and of any modifications that were needed and (b) to get a sense of students' help-seeking behavior. In this analysis we used an existing set of logs of student-tutor interactions, covering the Angles unit, one of the units that make up the curriculum of the Geometry Cognitive Tutor. We replayed these logs to compare students' actual step-by-step help-seeking behavior, without any tutoring on help-seeking, with the predictions made by the help-seeking model. This methodology might be called "meta-cognitive model tracing after the fact" – it is not the same as regular model tracing, since one does not see how students change their help-seeking behavior in response to the tutor's feedback. We determined the extent to which students' help-seeking behavior conforms to the model and the extent to which such conformance is predictive of their learning results. We then used the results to improve the model.

More specifically, on the assumption that the model was correct, we determined the frequency of the categories of meta-cognitive bugs defined by the model (and described above). We then computed correlations between the frequency of these bugs and students' learning outcomes. We then considered, in light of these data, whether and where our temporary assumption that the model was adequate was tenable. For example, if a particular type of meta-cognitive error occurred with "reasonable" frequency and was negatively correlated with learning, we took this as evidence that the error had been modeled adequately. Where errors were highly frequent or were not correlated with learning, we considered how the model should be modified. This would make it possible for the Help Tutor to intervene with appropriate frequency and when meta-cognitive errors occur that are correlated (negatively) with learning.

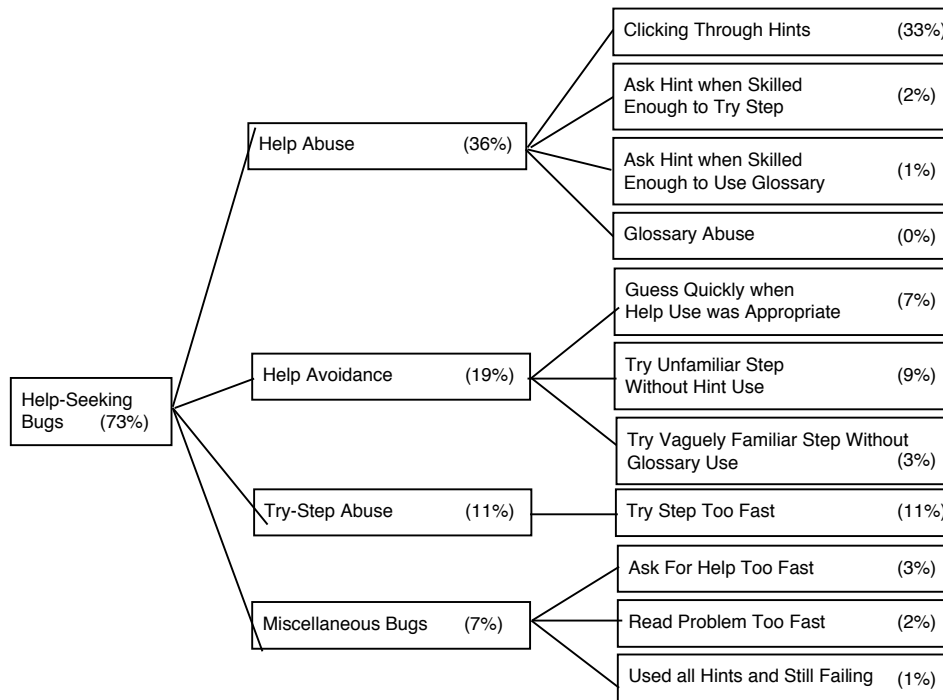


Fig.4. A taxonomy of help-seeking bugs. The percentages indicate how often each bug occurred in a data set of student-tutor interactions collected during an earlier experiment.

The data used in the analysis were collected during an earlier study in which we compared the learning results of students using two tutor versions, one in which they explained their problem-solving steps by selecting the name of the theorem, principle, or definition that justifies it and one in which the students solved problems without explaining (Aleven & Koedinger, 2002). For purposes of the current analysis, we group the data from both conditions together. It is appropriate to do so, since there was no significant difference in the help-seeking error rate of students between conditions. Students spent approximately 7 hours working on this unit of the tutor. The protocols from interaction with the tutor include data from 49 students, 40 of whom completed both the pre- and post-tests. The students performed a total of 58,732 actions related to skills tracked by the tutor.

The logs of the student-tutor interactions were replayed with each student action (either an attempt at answering, a request for a hint, or the inspection of a Glossary item) checked against the predictions of the help-seeking model. Actions that matched the model's predictions both in type (e.g., hint request) and in manner (i.e., were they done deliberately?) were recorded as "correct" help-seeking behavior. On the other hand, actions that did not match the model's predictions were recorded as incorrect help-seeking behavior and were classified automatically with respect to the bug taxonomy of Figure 4, based on the bug rules that were matched. We computed the frequency of each bug category. We also computed each category's correlation

Table 3

Correlation coefficients for the relation between the frequencies of meta-cognitive bug categories and post-test scores, with pre-test scores partialled out. Correlations that are significant at the 0.05 level are marked with an asterisk (“*”).

Help Abuse	Help Avoidance	Try-Step Abuse	Misc. Bugs	Total Meta-cognitive Bugs
-0.48*	-0.12	0.265	-0.38*	-0.65*

with learning, by computing a linear regression between post-test score and bug rate while controlling for pre-test score.

Help-seeking error rate

The overall percentage of help-seeking errors was 73%. That is, 73% of the students’ actions did not conform to the help-seeking model. The most frequent errors at the meta-cognitive level were *Help Abuse* (36%), with the majority of these being *Clicking Through Hints* (33%). The next most frequent category was *Help Avoidance* (19%), which covers situations in which help is not used even though it is likely to be beneficial. The category *Try-Step Abuse* – quick attempts at answering steps – was also quite frequent (11%), especially if one takes into account that the 2nd-level category *Guess Quickly When Help Use was Appropriate*, can also be seen as a form of *Try-Step Abuse* (in addition to being *Help Avoidance*). Counting it as *Try-Step Abuse* would increase the frequency of this category to 18%.

The frequency of help-seeking bugs correlated strongly with the students’ post-test performance, with pre-test score partialled out (partial $r = -0.65$, $t(2,38)=5.2$, $p<0.0005$), as shown in Table 3. The model therefore is a good predictor of learning gains – the more help-seeking bugs students make, the less they learn. The correlation between students’ frequency of success at the cognitive level (computed as the percentage of problem steps that the student completed without errors or hints from the tutor) and learning gain is about the same (partial $r = 0.65$, $t(2,38)=5.2$, $p<0.0005$) as the correlation between help-seeking bugs and learning. Finally, success in help-seeking and success at the cognitive level were highly correlated ($r=0.81$, $t(1,38)=73.8$, $p<0.0005$). In a multiple regression, the combination of pre-test, cognitive success rate, and meta-cognitive success rate explains 58% of the variance in the data, but only the contribution of the cognitive success rate is significant.

We also looked at how the bug categories correlated with learning (shown in Table 3). Both *Help Abuse* and *Miscellaneous Bugs* were negatively correlated with learning with $p < 0.01$. On the other hand, *Try-Step Abuse* and *Help Avoidance* were not correlated with learning.

The high correlation between errors at the cognitive level and learning results at the cognitive level is probably not surprising. It is a rather typical result in many learning scenarios that the students who experience greater difficulties during learning (and high hint use is a clear sign of difficulty) tend to come away with lower learning outcomes (e.g., Wood & Wood, 1999).

We see several likely reasons for the strong correlation between cognitive and meta-cognitive success. First, students who frequently clicked through hints, a very frequent meta-cognitive error, were likely to have both (a) a high rate of help-seeking bugs, since each click counts as a separate meta-cognitive error, and (b) a low cognitive success rate – since a step was

correct only if it was completed without help or errors. Thus, steps with clicking through episodes were considered to be incorrect at the cognitive level. This explanation would be true in particular if clicking through were a likely way of reacting (perhaps out of frustration) to hints that had been seen before. Frequent clicking through may also contribute to a high correlation between cognitive and meta-cognitive success (failure, rather) in that a student who clicks through hints is not likely to learn much from the hints and therefore is likely to continue to have trouble with the skill at issue (leading to further cognitive errors and possibly, click-through episodes and thus strengthening the correlation). A second reason for the high correlation between cognitive and meta-cognitive errors may be that the meta-cognitive model prescribes that students, when confronted with a step for which their skill mastery is low, start out by requesting a hint rather than by trying the step. Performance-oriented students, however, may be inclined to try steps even if they are not very confident they know how to do them, and rely on tutor feedback to tell them if they need to request hints. That strategy is more in line with the popular (and rational) test-taking strategy that says that one should guess even when not sure. Clearly, attempts at steps for which one has low skill are likely to fail (at the cognitive level). Since such attempts are also considered meta-cognitive errors, they contribute to a high correlation between cognitive and meta-cognitive success.

The high correlation between success at the meta-cognitive level and learning outcomes at the cognitive level is somewhat difficult to interpret, given that success at the meta-cognitive level and the cognitive level are highly correlated, and given that the meta-cognitive performance does not explain any additional variance in learning outcomes at the cognitive level above and beyond the cognitive performance. It is tempting to see the high correlation as evidence that better help-seeking *causes* better learning. In particular, it is hard *not* to think that frequent guessing or frequent clicking through hints is detrimental to learning, compared to a more deliberate approach. The data however are correlational and do not confirm such an interpretation. On the other hand, they do not rule out a causal link. The causal link (or absence thereof), which is ultimately what we are after, will be addressed in the next phase of the project, where (after modifying the meta-cognitive model as described below) we evaluate whether the Help Tutor has a positive influence on student learning. If the Help Tutor helps students to learn better, a fair conclusion is that help-seeking is an important influence on learning. Perhaps it is relevant to point out that in the literature on another meta-cognitive phenomenon, self-explanation, papers that reported correlations with learning (e.g., Chi, Bassok, Lewis, Reimann, & Glaser, 1989) preceded papers that provided causal evidence, namely, better learning outcomes due to manipulation of students' self-explanation behavior by means of prompts, up-front instruction, or feedback on explanations (e.g., Alevén & Koedinger, 2002; Bielaczyc, Pirolli, & Brown, 1995; Chi, de Leeuw, Chiu, & Lavancher, C. 1994; Renkl, Stark, Gruber, & Mandl, 1998).

In the next sections, we consider to what degree the results from our off-line analysis of students' help-seeking behavior confirm that our meta-cognitive model is on the right track and where they indicate the need for improvement. Where the model is deemed accurate, we evaluate what the analysis tells us about students' help-seeking behavior.

Table 4
(Cognitive) success rate on attempts at solving a step

	All attempts	First attempts	Other attempts
Deliberate attempts	62%	62%	59%
Hasty attempts	60%	77%	43%

Try-step Abuse: Fast solution attempts

The category *Try-Step Abuse* was not correlated with learning, suggesting that we had not modeled this phenomenon accurately. This category includes (overly) fast attempts at solving a step, which, as mentioned, were (somewhat coarsely) defined as attempts on which the student spent less than 7 seconds before acting. These meta-cognitive bugs fall under two categories, depending on whether the student was sufficiently skilled to try the step or not, according to the model's criterion. First, if the student was sufficiently skilled, her only error was the fact that the attempt was too fast. This bug is referred to as *Try-Step Too Fast* and falls under the broader category of *Try-Step Abuse*. Second, if the student tries a step quickly without being sufficiently skilled to attempt the step without help, she makes a double meta-cognitive error: in addition to being too quick, she avoids the use of help. This bug is referred to as *Guess Quickly when Help Use was Appropriate* and is classified as a form of *Help Avoidance* (reflecting our preference for using as the basis for aggregation the action that the students *should* have taken, rather than the one they *actually* took). Our intuition was that fast attempts are evidence of guessing and are likely to have a detrimental effect on learning. However, the data did not confirm this intuition. As shown in Table 3, there was no correlation between a student's frequency of *Try-Step Abuse* and the learning outcomes. Further, the data indicated that haste did not harm a student's chances of getting the answer right. As shown in Table 4, 60% of fast solution attempts were correct, versus 62% of deliberate attempts. (By correct we mean correct at the cognitive level, that is, with respect to geometry. By deliberate we mean not fast, that is, attempts on which the student spent more than 7 seconds.) Thus, we further investigated the data in an attempt to identify more appropriate definitions for these bug categories.

A closer look reveals that the student's skill level is an important factor distinguishing fast attempts that are negatively correlated with learning and those that are not. Trying fast on steps for which the student has a probability of high skill is not negatively correlated with learning. In fact, when controlling for pre-test, the frequency of this behavior is positively correlated with learning ($r = .28$, $p < .1$), although the effect is only marginally statistically significant. On the other hand, fast attempts on steps for which the student does not have high skill mastery are negatively correlated with learning ($r = -.32$, $p < .05$).

In addition, the context in which the solution attempt occurred reveals a clear difference between deliberate and fast attempts. When looking only at answer attempts that occurred as the student's first action on a given step (i.e., without previous errors, hint requests, or Glossary use on the same step), the overall success rate on deliberate attempts (defined as attempts on which the student spent more than 7 seconds) is 62%. On fast first attempts, the success rate climbs to

Table 5
Success rate on meta-cognitively correct answer attempts and on answer attempts classified as *Help Avoidance*

	After an error	After a hint	First action
Metacogn. Correct	51%	84%	78%
Help Avoidance	29%	62%	39%

77%. On the other hand, on attempts that are not the first action on a step (i.e., attempts that follow an error, hint request, or Glossary inspection), the results are reversed. Deliberate attempts have a higher success rate than fast attempts: 59% v. 43%. In other words, when students do not know the answer immediately, quick attempts at solving it after an error or hint have a substantially lower chance of success. These differences are consistent across all skill levels, meaning that fast attempts after an error or a hint are not successful at any skill level.

It may seem surprising that on their first attempt, students had a higher chance of getting a step right when they worked quickly, but upon reflection, it is reasonable. Quick performance may indicate that the student is highly skilled. That is, the fact that a first attempt is quick may indicate that the student already knows how to solve this step or solved it in her head while working on previous steps. The data indicate however that on other attempts (i.e., attempts to solve a step that follow one or more earlier unsuccessful attempts), quick action is undesirable. Thus, combining the results of the analyses, a better way to distinguish between incorrect quick answers that are guesses and incorrect quick answers that are slips in skilled performance may be to consider the skill level of the student and whether or not the attempt is the first action on the given step. When the student's skill level is high, quick first attempts should not be considered a meta-cognitive bug. We subsequently changed the model accordingly.

Help Avoidance

Like *Try-Step Abuse*, the category *Help Avoidance* did not correlate with learning, a reason to investigate this category further. As it turned out, the frequency of *Help Avoidance* depended on the context in which it occurred. *Help Avoidance* happened mainly on first actions on a step and after errors and happened only very infrequently after hint requests: 29% of all first actions and 45% of all responses to errors were classified as being of this type. After asking for a hint, on the other hand, students usually did not avoid the use of help, but instead kept asking for more hints. Only 4% of the actions following a hint were of type *Help Avoidance*. This is not surprising, given students' tendency to click through hints, discussed below. The frequency of *Help Avoidance* on first actions and after errors is even higher if one considers that by definition, this bug cannot occur when students have a high probability of mastering the skill involved in the step – the help-seeking model prescribes trying the step in such situations (unless multiple errors were made already). When students had low or medium skill level, they avoided help on 78% of their first attempts, on 63% of their actions after an error, and on 4% of actions following a hint.

There was evidence that avoiding help led to lower performance with the tutor (see Table 5). The (cognitive) success rate on answer attempts classified as *Help Avoidance* (which were mostly attempts at solving a step, given the low frequency of Glossary use) was much lower than on

actions deemed meta-cognitively correct, especially after an error (29% v. 51% correct) and on first actions (39% v. 78% correct). Avoiding help after a hint also appears to have a negative influence on performance, although the effect is smaller (62% v. 84% correct). As mentioned, the frequency of this meta-cognitive behavior was very low.

The data on performance with the tutor suggest that the model captures the category *Help Avoidance* quite well. Why, then, was there no (negative) correlation between *Help Avoidance* and learning? One possible answer is that some students got help from their peers or the teacher.

We have anecdotal data that at times students prefer to ask their friends or teachers over the system, in order not to affect their estimated skill level. An alternative explanation may be that even if the model accurately captures situations where students needed help but did not get it, it may still be that those who *did* opt to request help from the tutor in such situations had little advantage over those who did not, perhaps because the help was unclear or inaccurate, or because the students used the tutor's help in an unproductive manner. The prevalence of *Help Abuse* (as mentioned, 36% of students' actions) lends support to the latter interpretation. Put differently, given that in this data set, *Help Avoidance* tended to be "*Help Abuse Avoidance*," it is perhaps not surprising that there is no correlation with learning. That explanation is bolstered by the analysis of the most frequent form of *Help Abuse*, presented next.

Help Abuse: Clicking Through Hints

As mentioned, *Clicking Through Hints* was by far the most frequent bug category. It captures students' moving to the next hint level without spending adequate time to read the current hint level. The threshold was initially set to 5 seconds for all hints. Since the next level replaces the current contents in the tutor's hint window, the chance to read the current level is gone. (There was no option for students to move backward in a hint sequence – that capability was added only later to the tutor.) More than 25,000 actions were classified under the *Clicking Through Hints* category. Further analysis revealed that these actions tended to appear in clusters. Overall, there were just under 4,700 such clusters, with an average cluster length of 5.4 clicks. There was considerable evidence of rapid, automatic clicking: 28% of the clicking through clusters contained more clicks than there were hint levels for the given step, conjuring up images of furious, somewhat mindless clicking. Many students had these extended click sequences, at least on occasion. Of the 49 students, 14 had at least one sequence of "clicking through" that was longer than 10 actions, whereas no hint sequence was longer than 8 levels, so that at most 7 clicks were needed to display the bottom-out hint. Five students had clicking through clusters whose length was longer than 20 occurrences. Of the *Clicking Through Hints* actions, 87% were done in less than 2 seconds, 39% in less than 0.5 seconds. Thus, there was strong evidence of frequent use of a strategy by which students try to reach the bottom-out hint as fast as they can, skipping all other hints, affirming the prevalence of a behavior first reported in Alevan and Koedinger (2000). Given that the decision to click through seems to be made up front, it is appropriate to view each clicking through episode as a single mental action, rather than counting each rapid click as a separate action, as we did in the analysis of meta-cognitive bugs reported above. Thus, we recalculated the frequencies of the bug categories, this time counting each clicking through episode as a single action.

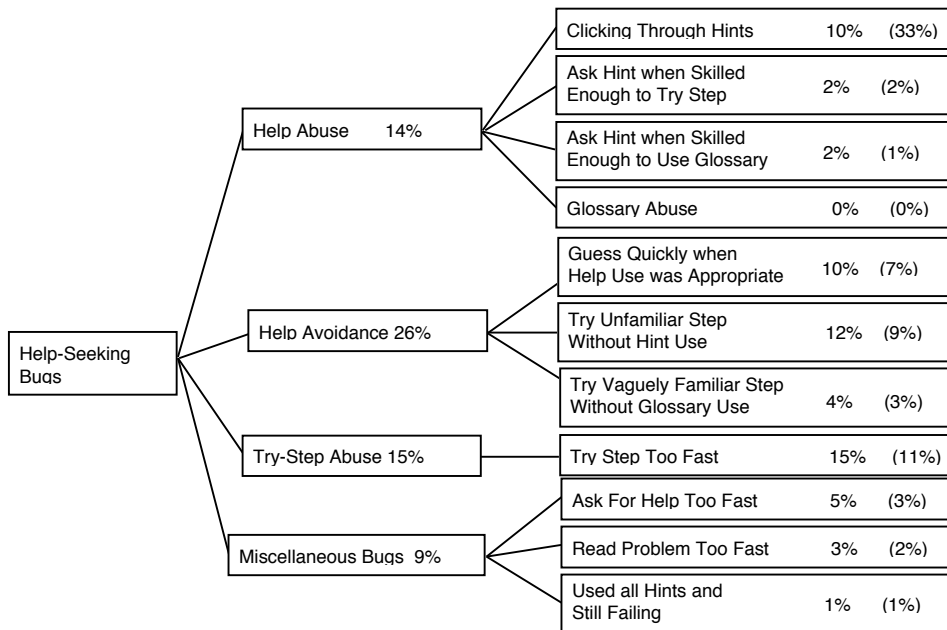


Fig.5. Frequency of help-seeking bugs when clusters of *Clicking Through Hints* bugs are viewed as a single action rather than one action per click – the frequencies shown in parentheses repeat the error frequencies reported in Figure 4.

As can be seen in Figure 5, under this assumption, the overall meta-cognitive error rate goes down to 64% and the frequencies of the error categories change rather dramatically. *Clicking Through Hints* is no longer the most common meta-cognitive error – its rate is now down to 10%, which is lower than those of *Try-Step Too Fast* (15%) and *Try Unfamiliar Step Without Hint Use* (12%). When counting each click as a single action, 82% of all hint requests were of type *Clicking Through Hints*. However, when counting each clicking through cluster only once, only 46% of all hint requests were clicking through clusters. Of the broader categories, *Help Avoidance* (26%) is now far more frequent than either *Help Abuse* (14%) or *Try-Step Abuse* (15%). Further, as shown in Table 6, *Help Avoidance* is now significantly negatively correlated with post-test (when controlling for pre-test), whereas before it was not.

Combined with the data presented above showing the negative association between *Help Avoidance* and performance with the tutor, this analysis lends support to the way *Help Avoidance* is captured in the model.

Table 6

Correlations between the bug categories and post-test, with pre-test partialled out, when counting clusters of *Clicking Through Hints* either as single actions (“collapsed”) or as multiple actions (“not collapsed”). The “Not collapsed” row repeats Table 3 for ease of comparison. Correlations that are significant at the 0.05 level are marked with an asterisk (“*”).

	Help Abuse	Help Avoidance	Try-Step Abuse	Misc. Bugs	Total Meta-cognitive Bugs
<i>Clicking Through Hints</i> clusters collapsed	-0.46*	-0.38*	0.111	-0.54*	-0.66*
Not collapsed	(-0.48*)	(-0.12)	(0.265)	(-0.38*)	(-0.65*)

DISCUSSION

The comparison of the help-seeking model against data of student-tutor interactions sheds light on the validity of the model and the adjustments that were needed in order to use it for “live” tutoring. The fact that the main bug categories of the model, with the exception of *Try-Step Too Fast*, correlate negatively with learning provides some measure of confidence that the model is on the right track. Before we comment on the changes to the model that were made as a result of the analysis, we review the findings about help-seeking that emerged. Given the state of development of the help-seeking model, these findings should be taken with a grain of salt, until it is clear that the model can be used successfully for tutoring. Nevertheless, we see this analysis as a step toward a better understanding of help-seeking and how to tutor it.

Implications with respect to students’ help-seeking behavior

A striking finding is the high frequency of help-seeking errors: 73% and 64% in two different analyses. While this error rate is too high to be useful for tutoring, a point that is taken up below, it confirms earlier findings that students’ help-seeking behavior is far from ideal (see e.g., Aleven et al., 2003). We also found that help-seeking errors correlate negatively with learning, underscoring the importance of helping students to improve help-seeking behavior by means of instruction. The finding that students often abuse hints confirms earlier work (Aleven & Koedinger, 2000; Aleven, McLaren, & Koedinger, 2006; Baker, Corbett, & Koedinger, 2004). The current analysis extends that finding by showing that help abuse is frequent relative to other kinds of help-seeking bugs and that it correlates negatively with learning. The relative frequency of this behavior is reduced when repeated rapid actions of clicking through hints are collapsed into a single mental action, but the negative correlation with learning remains. A new finding is the high *Help Avoidance* rate and the fact that *Help Avoidance* correlates negatively with learning. It is important therefore that the Help Tutor is able to comment on such behavior, as illustrated in an extended example in this paper.

The data suggests that we were not very successful at modeling the phenomenon of *Try-Step Abuse*, which involves fast attempts at answering/guessing a step in a tutor problem. Our analysis of this phenomenon reveals that answering quickly is likely to be appropriate when students have high skill, at least as an initial action on a step, before hint use or errors.

Changes made to the model based on the meta-cognitive error analysis

The analysis made it clear that a number of changes to the model were necessary, in order to use it for tutoring. For one thing, it had to be made more lenient. The error rate of 73% implies that the Help Tutor would intervene (i.e., present an error feedback message) approximately 3 out of every 4 actions taken by a student. In practical use, this is likely to be annoying and distracting to the student. Therefore, we have updated and refined the model in a number of ways, prompted in part by the analysis results and in part by our findings in early pilot tests of the Help Tutor with real students, which are described in more detail below.

First, we changed the model so that it would be less persistent in asking the student to modify her behavior. Previously, the tutor would comment on the same meta-cognitive bug each time it came up on a single step, across a range of meta-cognitive bugs. In the updated model, the

Help Tutor typically does not comment on repetitions of the same meta-cognitive bug. For example, where previously the Help Tutor would refuse to give a hint if it deemed the student to be skilled enough to solve a step without a hint, no matter how often the student repeated the request, in the updated model, after an initial refusal a hint will be forthcoming if the hint request is repeated. Previously, when the student clicked through a hint, the Help Tutor would present a message and then re-display the same hint, requiring that the student again spend the same amount of reading/comprehending time as initially required. In the updated model, the time threshold for re-reading a hint is reduced.

Second, since the Glossary is used as a mechanism to choose a theorem, principle or definition to explain steps (that is, in the particular tutor unit under study, the students are required to explain their numeric answers by indicating which theorem or definition justifies the step – they can do so by selecting from the Glossary; thus, the Glossary serves as a menu of “Reasons” in addition to its usual information lookup function), we changed the Help Tutor so that it no longer displays messages related to Glossary abuse on Reason steps. This change illustrates that it is important to take into account the context in which help facilities are used. While ordinarily we would discourage abuse of the Glossary, we needed to be more lenient because of the dual functions of the Glossary.

Third, we fine-tuned the hint reading time, so that the Help Tutor would respond more appropriately to an estimate of actual reading time and skill level. We used the Card, Moran, and Newell (1983) reading research as a baseline and then assumed that high skill students would need 1/3 of the time to read a hint as a mid or low level student would need.

Finally, we decided that it is not a meta-cognitive error to try a step quickly when the likelihood of skill mastery for that step is high. The error analysis showed that this kind of meta-cognitive behavior was not related to lower learning outcomes. Moreover, students were frequently successful when trying steps quickly. Perhaps this finding can be explained by students’ thinking or planning ahead, which high-skill students may be more likely to do or simply by the fact that their high skill enabled them to be quick and accurate at the same time. To further reduce the likelihood of false alarms, we reduced the time threshold for deciding whether or not an attempt was “too quick” from 7 seconds to 2 seconds (see also Roll, Baker, Aleven, McLaren, & Koedinger, 2005). This change reduces the number of attempts deemed fast by a factor of 8. Thus, the Help Tutor may not catch all fast attempts, but when it does point out to a student that she is working too fast, the message has a lot of face validity. Distinguishing further between quick, incorrect steps that are likely to be guesses and those that are slips in skilled performance will probably remain a difficult problem to address.

While these kinds of changes may appear to be rather detailed, they are likely to have a major impact on the student’s acceptance of the Help Tutor and hence on its success. In particular, it seemed important that students would often agree with the Help Tutor’s assessment of their help-seeking behavior. Otherwise, the Help Tutor might lose credibility and end up being ignored.

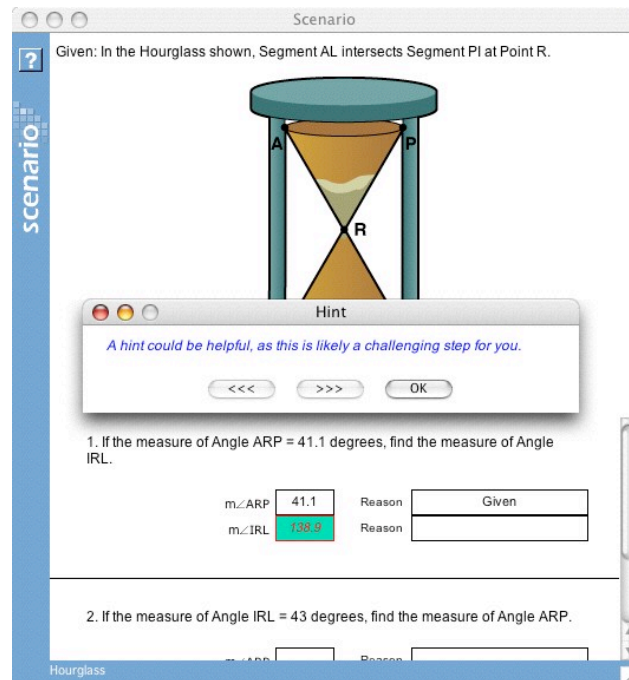


Fig.6. The Help Tutor provides feedback on students' help-seeking behavior, including situations in which students avoid asking for help.

A PILOT STUDY OF THE HELP TUTOR

The error analysis reported above helped in validating the model, but did not produce any information about how students react to the Help Tutor's advice. Therefore, we conducted a small-scale pilot study to find out (a) whether students perceive the Help Tutor in a positive light, (b) whether and how the Help Tutor influences their behavior, and (c) whether the Help Tutor intervenes with appropriate frequency. Four high-school students from a public school in a suburban area worked with the Help Tutor, shown in Figure 6. Three of them worked with the tutor for two sessions, one week apart. The fourth student worked with the tutor for the second session only. The students were accustomed to working with the Geometry Cognitive Tutor, as they use it regularly in their classroom, but they were not familiar with the particular curriculum unit involved in the study. The Help Tutor sessions took place during class periods in which the students normally used the Cognitive Tutor, but in a different classroom, separate from the other students in the class, who did not participate in the pilot study. The Help Tutor was modified between the sessions, to improve it based on the experience gained during the first session. Some of the changes are described above.

The results presented here relate to the second session only. Students completed a total of 685 actions (defined as answer attempts, hint requests, or Glossary inspections). The overall ratio of help-seeking errors (i.e., deviations from the Help Tutor's model) was 16%, ranging from 9%

to 24% for the different students (see Table 7). Thus, the Help Tutor intervenes once for every six student actions, which seems to be a very reasonable frequency (i.e., frequent enough to have an impact on students' help-seeking behavior but not too frequent to annoy the students). Clearly, we were successful in bringing down the error rate of 73% that we found during the error analysis. Even more encouraging was the fact that the rate of help-seeking errors declined during the session. It dropped from 18% during the first half to 14% during the second half. The reduction in error rate is statistically significant (paired- $t=4.0$, $p<0.03$), evidence that the students adapted their behavior to the tutor. Interestingly, the reduction in the error rate cannot be attributed to the students' getting more fluent with the geometry material, since it occurred irrespective of the student's skill level for the given step (high skill: from 16% to 10%; low skill: from 33% to 29%). Of course, it should be kept in mind that the pilot study was of short duration and we did not measure the influence that the Help Tutor may have had on students' learning of the domain-related skills and knowledge (i.e. geometry).

At the end of each session, the students filled out a questionnaire in which they were asked whether they welcomed tutor feedback suggesting that they work slower, ask for a hint, or try without using a hint. They were asked also whether the tutor made these suggestions at appropriate times and with reasonable frequency. One of the four students, though fond of the Help Tutor after the first session, was quite annoyed by it after the second. She did not like the tutor's suggestions that she reduce the number of hint requests. During the two sessions, this student received more than twice the number of error messages following her hint requests than the other students, due to her faulty use of help. The other three students had a positive opinion about the tutor. All three wanted the tutor to offer suggestions that they work slower and they thought that the tutor presented them at appropriate moments. Two of the three welcomed suggestions from the tutor that they try a step by themselves and thought the tutor presented them with appropriate frequency. The third student thought that these messages are unnecessary.

In sum, these answers are encouraging. They seem to indicate that the Help Tutor's advice was perceived as appropriate and that the Help Tutor did establish some credibility with the students. This is not to say that they always reacted positively at the moment that they received feedback from the Help Tutor: witness the remarks of one of the students discussed above. Particularly the messages that denied a student's help request, suggesting that the student try a step without asking for a hint, were not very popular. After such a message, one student said: "I hate this tutor!" and another replied: "Because it makes you do the work yourself..." Such comments should probably not be taken as a sign that the tutor was ineffective. It is not unusual for students to complain when working with Cognitive Tutors, even though on the whole, there is clear evidence that the tutors are motivating (Schofield, 1995). Furthermore, if it is true that the Help Tutor makes students work harder, as suggested by the second student's comment, it may well be that students' learning outcomes are affected positively.

Table 7
Frequency of help-seeking errors during the pilot study

	Student 1	Student2	Student 3	Student 4	Overall
1st half	20%	27%	10%	15%	18%
2nd half	18%	21%	7%	12%	14%
Overall	19%	24%	9%	13%	16%

It will be interesting to see whether such positive attitudes toward the Help Tutor will persist when students use the Help Tutor over extended periods of time. It may be that students are put off by the absence of some of the rewards that they are accustomed to as they work with the regular Cognitive Tutor. For example, the Help Tutor does not display meta-cognitive skill bars that would indicate the student's mastery of help-seeking skill (although there is no reason why it could not display them – we choose not to display them to keep things simple). On the other hand, the Help Tutor may help students to be more deliberate and reflective learners, which may lead to higher levels of understanding and, hence, motivation.

CONCLUSION

In order to test the hypothesis that meta-cognitive tutoring can help students to learn better, we have developed a Help Tutor agent, designed to be seamlessly added to existing Cognitive Tutors. This tutor agent provides feedback with respect to students' help-seeking behavior. In particular, it comments when students do not use the Cognitive Tutor's help facilities effectively. It is hypothesized that this feedback will help students to become better help seekers and better learners.

In order to implement the Help Tutor, we are using, at the meta-cognitive level, an approach to tutoring that has been very successful at the cognitive level, namely, the Cognitive Tutor technology (Anderson et al., 1995). The Help Tutor uses a model of adaptive help-seeking behavior, expressed as production rules, to monitor students' meta-cognitive behavior. It does so in exactly the same way that Cognitive Tutors for, say, high-school mathematics use their cognitive models to provide guidance with algebra or geometry. The development of the model is a contribution in its own right, since no existing model of help-seeking that we know of is detailed enough to run on a computer.

To validate the model, we have run it against pre-existing tutor data. The analysis revealed that the model is generally on track, since its major bug categories, with one exception, correlated negatively with learning. The analysis revealed a high overall meta-cognitive error rate – too high for practical use in a tutor. To the extent that the model is reasonable, this analysis confirms earlier findings that high-school students tend not to seek help effectively when working with computer-based tutors. Consistent with earlier work, we found a proliferation of hint abuse. Novel findings of the analysis were the negative correlations between various categories of meta-cognitive bugs and students' learning results, which inspire confidence that the model is on the right track. Further, students' avoidance of help emerged as a serious concern: it was both frequent and (under the reasonable assumption that episodes of rapid clicking through hints

should be considered a single mental action), negatively correlated with learning. The analysis of meta-cognitive errors led to a number of refinements in the model, aimed primarily at bringing down the overall error rate of 73% and also, at making the Help Tutor less “persistent” when the student repeats a meta-cognitive bug. Such changes we believe will be important for the Help Tutor’s acceptance by students.

In order to validate the model and tutor further, we conducted a pilot study with the Help Tutor, involving four students. The results of this pilot are cause for cautious optimism. The students seemed to adapt to the Help Tutor, as suggested by the fact that over the limited time that they used the Help Tutor, their meta-cognitive error rate went down. Further, in their questionnaires, three of the four students reported that they welcomed the Help Tutor’s input and found that the Help Tutor gave appropriate feedback. Thus, the Help Tutor seemed to have established some credibility in the eyes of these students. However, these results should be treated with caution. The pilot study was of short duration, involved only a small number of students, and did not evaluate any influence on students’ learning outcomes that the Help Tutor may have had.

The next step in our research will be to conduct a controlled experiment, in actual classrooms, to evaluate whether students’ help-seeking skill improves as a result of feedback from the Help Tutor and whether they obtain better learning outcomes. We will also evaluate whether better help-seeking behavior persists beyond the tutor units in which the students are exposed to the Help Tutor and whether students learn better in those units as a result. That is, we will evaluate whether the Help Tutor helps to prepare students for better future learning.

ACKNOWLEDGEMENTS

We would like to thank Matthew Welch, Michele O’Farrell, and Eunjeong Ryu for their assistance. This research is sponsored by NSF Award IIS-0308200 and NSF Award SBE-0354420 to the Pittsburgh Sciences of Learning Center. The contents of the paper are solely the responsibility of the authors and do not necessarily represent the official views of the NSF.

REFERENCES

- Alevén, V. & Koedinger, K. R. (2002). An effective meta-cognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*, 26(2), 147-179.
- Alevén, V. & Koedinger, K. R. (2000). Limitations of Student Control: Do Students Know when they need help? In G. Gauthier, C. Frasson & K. VanLehn (Eds.) *Proceedings of the 5th International Conference on Intelligent Tutoring Systems, ITS 2000* (pp. 292-303). Berlin: Springer Verlag.
- Alevén, V., McLaren, B. M., & Koedinger, K. R. (2006). Towards Computer-Based Tutoring of Help-Seeking Skills. In S. Karabenick & R. Newman (Eds.) *Help Seeking in Academic Settings: Goals, Groups, and Contexts*. Mahwah, NJ: Erlbaum.
- Alevén, V., McLaren, B., Roll, I., & Koedinger, K. (2004). Toward tutoring help seeking: Applying cognitive modeling to meta-cognitive skills. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.) *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 227-239). Berlin: Springer Verlag.

- Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R.M. (2003). Help Seeking in Interactive Learning Environments. *Review of Educational Research*, 73(2), 277-320.
- Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale, New Jersey: Lawrence Erlbaum.
- Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4, 167-207.
- Arroyo, I., Beck, J. E., Beal, C. R., Wing, R., & Woolf, B. P. (2001). Analyzing students' response to help provision in an elementary mathematics intelligent tutoring system. In R. Luckin (Ed.) *Papers of the AIED-2001 Workshop on Help Provision and Help Seeking in Interactive Learning Environments* (pp. 34-46).
- Baker, R. S., Corbett, A. T., & Koedinger, K. R. (2004). Detecting Student Misuse of Intelligent Tutoring Systems. In J. C. Lester, R. M. Vicario & F. Paraguaçu (Eds.) *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 531-540). Berlin: Springer Verlag.
- Bartholomé, T., Stahl, E., Pieschl, S., & Bromme, R. (in press). What matters in help-seeking? A study of help effectiveness and learner-related factors. *Computers in Human Behavior*.
- Bielaczyc, K., Pirolli, P. L., & Brown, A. L. (1995). Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem Solving. *Cognition and Instruction*, 13, 221-252.
- Bransford, J. D. Brown, A. L., & Cocking, R. R. (Eds.) (2000). *How People Learn: Brain, Mind, Experience, and School*. Washington, CD: National Academy Press.
- Bransford, J. D. & Schwartz, D. L. (2001). Rethinking transfer: a simple proposal with multiple implications. *Review of Research in Education* 24(3), 61-100.
- Brown, A. L. & Campione, J. C. (1996). Guided Discovery in a Community of Learners. In K. McGilly (Ed.) *Classroom Lessons: Integrating Cognitive Theory and Classroom Practice* (pp. 229-270). Cambridge, MA: The MIT Press.
- Bull, S., Greer, J., & McCalla, G. (2003). The Caring Personal Agent. *International Journal of Artificial Intelligence in Education*, 13(1), 21-34, 2003.
- Bunt, A., Conati, C., & Muldner, K. (2004). Scaffolding self-explanation to improve learning in exploratory learning environments. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.) *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 656-667). Berlin: Springer Verlag.
- Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Mahwah, NJ: Erlbaum.
- Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science*, 13, 145-182.
- Chi, M. T. H., de Leeuw, N., Chiu, M., & Lavancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18, 439-477.
- Conati, C. & VanLehn, K. (2000). Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education*, 11, 398-415.
- Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4, 253-278.
- Dutke, S. & Reimer, T. (2000). Evaluation of two types of online help information for application software: Operative and function-oriented help. *Journal of Computer-Assisted Learning*, 16, 307-315.
- Gama, C. (2004). Metacognition in interactive learning environments: the reflection assistant model. In J. C. Lester, R. M. Vicario & F. Paraguaçu (Eds.) *Proceedings 7th International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 668-677). Berlin: Springer.
- Gross, A. E. & McMullen, P. A. (1983). Models of the help-seeking process. In J. D. Fisher, N. Nadler & B. M. DePaulo (Eds.) *New directions in helping* (Vol. 2, pp. 45-61). New York: Academic Press.

- Hambleton, R. K. & Swaminathan, H. (1985). *Item Response Theory: Principles and Applications*. Boston: Kluwer.
- Karabenick, S. A. (Ed.) (1998). *Strategic help seeking. Implications for learning and teaching*. Mahwah: Erlbaum.
- Karabenick, S. A. & Knapp, J. R. (1988a). Help seeking and the need for academic assistance. *Journal of Educational Psychology*, 80, 406-408.
- Karabenick, S. A. & Knapp, J. R. (1988b). Effects of computer privacy on help seeking. *Journal of Applied Social Psychology*, 18, 461-472.
- Knapp, J. R. & Karabenick, S. A. (1988). Incidence of formal and informal academic help seeking in higher education. *Journal of College Student Development*, 29, 223-227.
- Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30-43.
- Koedinger, K. R., Corbett, A. T., Ritter, S., & Shapiro, L. (2000). Carnegie Learning's Cognitive Tutor: Summary Research Results. White paper. Available from Carnegie Learning Inc., e-mail: info@carnegielearning.com, web: <http://www.carnegielearning.com>
- Luckin, R. & Hammerton, L. (2002). Getting to know me: Helping learners understand their own learning needs through meta-cognitive scaffolding. In S. A. Cerri, G. Gouardères & F. Paraguaçu (Eds.) *Proceedings of Sixth International Conference on Intelligent Tutoring Systems, ITS 2002* (pp. 759-771). Berlin: Springer.
- McKendree, J. (1990). Effective feedback content for tutoring complex skills. *Human Computer Interaction*, 5, 381-413.
- Mitrovic, A. (2003). Supporting Self-Explanation in a Data Normalization Tutor. In V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo & K. Yacef (Eds.) *Supplemental Proceedings of the 11th International Conference on Artificial Intelligence in Education, AIED2003* (pp. 565-577). School of Information Technologies, University of Sydney.
- Nelson-LeGall, S. (1981). Help-seeking: An understudied problem-solving skill in children. *Developmental Review*, 1, 224-246.
- Newman, R. S. (1994). Adaptive help seeking: a strategy of self-regulated learning. In D. H. Schunk & B. J. Zimmerman (Eds.) *Self-regulation of learning and performance: Issues and educational applications* (pp. 283-301). Hillsdale, NJ: Erlbaum.
- Newman, R. S. (1998). Adaptive help seeking: A role of social interaction in self-regulated learning. In S. A. Karabenick (Ed.) *Strategic Help Seeking. Implications for Learning and Teaching* (pp. 13-37). Mahwah: Erlbaum.
- Newman, R. S. (2002). How Self-Regulated Learners Cope with Academic Difficulty: The Role of Adaptive Help Seeking. *Theory Into Practice*, 41(2), 132-138.
- Palincsar, A. S. & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension monitoring activities. *Cognition and Instruction*, 1, 117-175.
- Puustinen, M. (1998). Help-seeking behavior in a problem-solving situation: Development of self-regulation. *European Journal of Psychology of Education*, 13(2), 271-282.
- Renkl, A. (2002). Learning from worked-out examples: Instructional explanations supplement self-explanations. *Learning and Instruction*, 12, 529-556.
- Renkl, A., Stark, R., Gruber, H., & Mandl, H. (1998). Learning from worked-out examples: the effects of example variability and elicited self-explanations. *Contemporary Educational Psychology*, 23, 90-108.
- Rich, C., Lesh, N.B., Rickel, J., & Garland, A. (2002). A plug-in architecture for generating collaborative agent responses. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS* (pp. 782-789).

- Ritter, S. (1997). Communication, cooperation and competition among multiple tutor agents. In B. du Boulay & R. Mizoguchi (Eds.) *Artificial Intelligence in Education, Proceedings of AI-ED 97 World Conference* (pp. 31-38). Amsterdam: IOS Press.
- Roll, I., Baker, R. S., Alevan, V., McLaren, B. M., & Koedinger, K. R. (2005). Modeling students' metacognitive errors in two intelligent tutoring systems. In *Proceedings of User Modeling 2005* (pp. 379-88). Berlin: Springer-Verlag.
- Rosenshine, B., Meister, C., & Chapman, S. (1996). Teaching students to generate questions: A review of the intervention studies. *Review of Educational Research*, 66 (2), 181-221.
- Ryan, A. M., Gheen, M. H., & Midgley, C. (1998). Why do some students avoid asking for help? An examination of the interplay among students' academic efficacy, teachers' social-emotional role, and the classroom goal structure. *Journal of Educational Psychology*, 90(3), 528-535.
- Schworm, S. & Renkl, A. (2002). Learning by solved example problems: Instructional explanations reduce self-explanation activity. In W. D. Gray & C. D. Schunn (Eds.) *Proceedings of the 24th Annual Conference of the Cognitive Science Society* (pp.816-821). Mahwah, NJ: Erlbaum.
- Schwartz, D. & Taylor, M. (2004). Inventing to prepare for future learning: the hidden efficiency of encouraging original student production in statistics instruction. *Cognition and Instruction*, 22(2), 129-84.
- Trafton, J.G. & Trickett, S.B. (2001). Note-taking for self-explanation and problem solving. *Human-Computer Interaction*, 16, 1-38.
- White, B. & Frederiksen, J. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction*, 16(1), 3-117.
- Wood, D. (2001). Scaffolding, contingent tutoring, and computer-supported learning. *International Journal of Artificial Intelligence in Education*, 12.
- Wood, H. & Wood, D. (1999). Help seeking, learning and contingent tutoring. *Computers and Education*, 33, 153-169.