

***In vivo* experiments on whether supporting metacognition in intelligent tutoring systems yields robust learning**

Ken Koedinger, Vincent Aleven, Ido Roll, & Ryan Baker

Introduction

Interactive educational technologies, like intelligent tutoring systems, provide an excellent platform on which to perform research on metacognition. Such systems provide fine-grained tracking and assessment of students' cognitive and metacognitive behaviors. They also facilitate the implementation of tightly controlled experiments within the context of real classrooms. This chapter covers recent research on the impact of different types of metacognitive support in intelligent tutoring systems on both the learning of domain content and desired metacognitive behaviors. We describe results from four lines of controlled experimentation, mostly performed in the context of real classrooms, *in vivo*. Our results show clear benefits of metacognitive supports for domain content learning. We also demonstrate innovative methods for assessing real-time metacognitive behaviors as students are engaging in learning activities. Finally, we present some partial success in attempts to produce lasting improvements in metacognitive behaviors themselves and discuss why metacognitive behaviors may be resistant to change.

Following Brown (1983), we define metacognition as thinking about cognition (memory, perception, reasoning, etc.) itself, that is, reasoning about one's own thinking. Within this chapter, we follow Schoenfeld et al.'s (1987) framing of "metacognitive learning strategies" as

specific kinds/uses of metacognition that aid learning, including planning, checking, monitoring, selecting, revising, and evaluating. These strategies are important components of models of self-regulated learning in the classroom, such as the model put forward by Pintrich (2004). Students' self-reported use of such strategies has been shown to correlate positively with academic outcomes (Pintrich & de Groot, 1990),

Researchers have long been interested in finding ways to help students acquire more complete and adaptive metacognitive skill in order to achieve better learning outcomes across academic topic areas and even across disciplines. In an influential volume entitled *How People Learn* (Bransford, Brown, & Cocking, 2000), one of three broad recommendations is to focus on improving students' metacognitive skills. A number of instructional programs focused on improving metacognition have been shown to be successful in actual classrooms. Some notable examples are programs that have focused on reciprocal teaching of reading skills (Palincsar & Brown, 1984; Rosenshine & Meister, 1994), on self-assessment of application of a scientific inquiry cycle (White & Frederiksen, 1998), and on strategies for self-regulation of mathematical reasoning (Mevarech & Kramarski, 2003). Classroom programs with a metacognitive focus have also been created for students with learning disabilities, although their effectiveness has not always been established in controlled experiments (Butler, 1998; Guthrie, Wigfield, & vonSecker, 2000).

In recent years, there has been increasing research on deploying metacognitive support within *computer-based* learning environments, particularly within intelligent tutoring systems (Azevedo & Cromley, 2004a; Arroyo et al., 2007; Biswas, Leelawong, Schwartz, Vye, & CTG, 2005). The research presented in this chapter focuses on novel ways of supporting metacognition in real educational settings. This is done using computer-based tutoring of specific metacognitive

skills, as contrasted to a comprehensive program of metacognitive instruction. In discussing various mechanisms to engage students in utilizing appropriate metacognitive strategies, we distinguish between static and adaptive metacognitive support (cf., Azevedo, Cromley, & Seibert, 2004b). By *static support* for metacognition we mean prompts or scaffolds that are present throughout instructional activities and do not vary between students. Unlike Azevedo et al.'s (2004b) notion of "fixed" support, our notion of static support may also involve feedback to the student as to whether they are successfully following the prompt. A classic example of static metacognitive support are prompts to students to "self-explain" (Chi, Bassok, Lewis, Reimann, & Glaser, 1994), which may occur in a text, a computer interface, or may be consistently delivered by a human tutor. If delivered by a computer or human tutor, these prompts may be followed by feedback to the student as to whether or not they successfully executed the prompt. Even when combined with feedback of this nature, we consider them to be static support, because the decision whether and when to prompt is not situation-specific, and not specific to any aspect of student metacognition. *Adaptive support* for metacognition on the other hand involves an active instructional agent, such as a teacher, human or computer tutor, that implements instructional strategies that adapt to individual students' metacognitive behavior, for example, by fading prompts for metacognitive behaviors (i.e., gradually reducing the frequency of these prompts), or by providing feedback on meta-cognitive errors. This definition of adaptive metacognitive support is more specific than Azevedo et al.'s (2004b) in that it specifically requires adaptation to student *metacognition*, not just to any aspect of student behavior, It thus requires that the system *assesses* aspects of student metacognition.

This chapter reviews four examples of our work on computer-based tutoring systems that provide either static or dynamic support for metacognition:

1. static support for self-explanation (Alevén & Koedinger, 2002)
2. adaptive support for error self-correction (Mathan & Koedinger, 2005)
3. adaptive ways of reducing “gaming the system” (Baker et al., 2006)
4. adaptive tutoring of help-seeking skills (Roll, McLaren, Alevén, & Koedinger, 2007a)

In evaluating the effect of these forms of metacognitive support, we have focused our efforts on assessing improvements in “robust learning”. Robust learning produces a high level of conceptual understanding and/or procedural fluency so that students perform well not only on immediate post-tests with items highly similar to training but also on tests of transfer, long-term retention, and preparation for future learning (see learnlab.org/research/wiki/index.php/Robust_learning). As shown in Figure 1, we have defined four goals for robust learning of metacognitive skills, each of which builds on the previous one.

The first goal is for students to improve their metacognitive behavior within the learning environment while they receive metacognitive support. Ideally, this improvement will lead to better learning gains in the domain targeted by the supported environment, which is the second goal of metacognitive tutoring. The third goal is for students to internalize the metacognitive practices and thus to demonstrate better metacognitive behavior in subsequent instruction using a similar interface. Our fourth goal is that the metacognitive support will make students better at future domain-level learning (faster and more complete), based on the metacognitive practices they internalized.

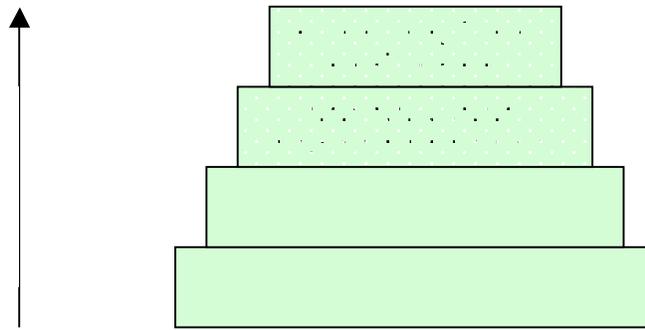


Figure 1. Goals of metacognitive tutoring. Toward the bottom are less ambitious and more proximal goals that support the more ambitious, ultimate goals above.

As our research has evolved, we have come to the view that an ideal experiment investigating the effect of metacognitive support will measure all four levels of learning. However, so far we have focused mainly on the first two goals; only in our most recent experiments on tutoring help seeking have we addressed the third goal. The fourth goal remains for future work. In the first and second lines of research, focused on the metacognitive strategy of *self-explanation* (Chi et al., 1989) and *error self-correction* (Nathan, Kintsch, & Young, 1992), we assessed the effect of the metacognitive support on domain learning but not on students' metacognitive activities. In the third and fourth lines of research, focused on discouraging *gaming the system* and encouraging better independent *help seeking*, we not only assessed changes in domain learning, due to the metacognitive support, but also in the use of appropriate and inappropriate metacognitive strategies while the intervention is in place. In the experiment on tutoring help seeking, we also evaluated whether students' help-seeking behavior had improved following the intervention.

The reported research emphasizes controlled *in vivo* experimentation (Koedinger & Corbett, 2006; learnlab.org/research/wiki/index.php/In_vivo_experiment), that is, experiments or quasi-experiments that strive to maximize both internal validity and ecological validity. With respect to

ecological validity, *in vivo* experiments use actual educational content and are run for realistic durations (hours rather than minutes of instruction) in the context of real courses and school settings. With respect to internal validity, *in vivo* experiments include a control condition and a treatment that involves a manipulation that targets a single causal principle (i.e., not full program variations as in typical randomized field trials) and *whenever possible* there is random assignment of students (or course sections) to condition. The “ecological” control condition used represents existing practice at the school setting.

1. Tutoring Self-Explanation

Our first line of research illustrating intelligent tutor support for metacognition focuses on the effect of supporting self-explanation. Of interest is a simple recurring metacognitive choice: After producing or being given a problem-solving step, students can either go on to the next step, or they can decide to self-explain the step they just completed, as a check of their understanding. Self-explanation has been studied extensively in the cognitive sciences, with one notable example being Chi et al. (1989), which reported an association between this metacognitive skill and learning. *Static* prompts for self-explanation have been shown in a number of lab studies to have a positive effect on student learning (Chi et al., 1994; Renkl et al., 1998; Siegler, 2002). Within the realm of intelligent tutoring systems, Conati and VanLehn (2000, 2001) implemented dynamic prompts for self-explanations, which were presented based on the system’s assessment of the student’s domain knowledge and the student’s understanding of the example that they were explaining. They found that these prompts, combined with feedback on self-explanations, were superior to a self-explanation control condition that used static prompts, although this control condition differed in a number of other ways as well. These researchers did not investigate, however, whether self-explanation support improved learning over a non-self-

explanation control condition, nor did they investigate whether students had become better self-explainers after the self-explanation support was removed. Our studies addressed the first limitation, whereas addressing the second remains for future work.

In two studies, we explored the effect of *static* support for self-explanation in an intelligent tutor that supports guided problem-solving practice. For each solved problem step, the tutor solicited simple self-explanations (namely, the name of the domain principle being applied, typically the name of a geometry theorem), and provided feedback on the correctness of these self-explanations. It did not, however, adapt to students' level of domain knowledge, or metacognition, and thus this type of support does not qualify as *adaptive* support.

Educational Problem: Too much Shallow Reasoning

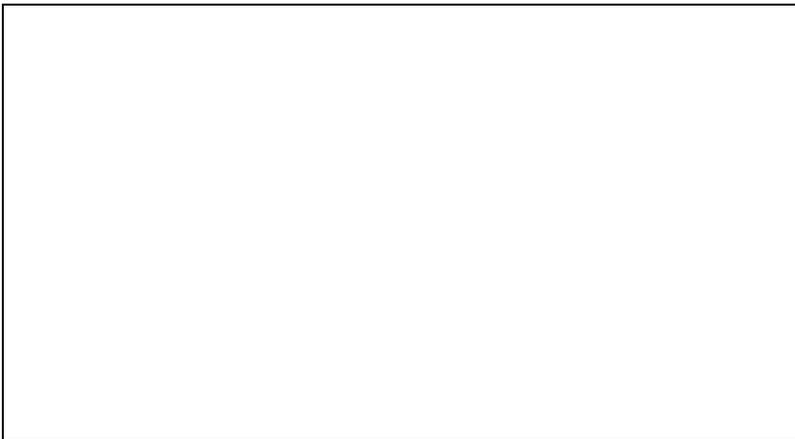


Figure 2: Shallow reasoning in geometry

Our research into self-explanation started with a practical and widespread educational problem: students often develop shallow reasoning strategies as they learn in a new domain or attempt to learn a new cognitive skill. In the high-school geometry course that we were developing at the time, we observed that students often appear to draw on shallow strategies such as that illustrated in Figure 2. In this example, the student erroneously infers that two angles that

look the same (namely, Angle 1 on the right and the un-numbered angle on the left) must have the same measures. We also found that the students were better at solving geometry problems than they were at justifying their reasoning steps by pointing out which theorems they were using (Aleven, Koedinger, Sinclair, & Snyder, 1998). This finding may be a manifestation of shallow reasoning, although other explanations are possible.

It is important to note that superficial strategies occur in many domains and with many forms of instruction. In physics problem solving, for example, there are many common misconceptions, such as confusing mass and weight, or the direction of velocity and acceleration (Ploetzner & VanLehn, 1997). Novices often classify physics problems by superficial features that are not typically related to the solution method (Chi, Feltovich, & Glaser, 1981). In mathematics, students may learn procedures without ever grasping the underlying principles (Byrnes & Wasik, 1991; Fuson, 1990; Hiebert & Wearne, 1996). In the domain of writing, students often exhibit a “knowledge telling” strategy in which they simply list everything they know about a subject rather than constructing a well-organized argument (Scardamalia & Bereiter, 1985).

Studies to Evaluate Self-Explanation Support

We conducted two experiments to test the hypothesis that simple self-explanation support, requiring that students connect their answers to the underlying problem-solving principles, would help students to learn in a more robust manner. We investigated this hypothesis in the context of the Geometry Cognitive Tutor. We summarize the second of two experiments (Aleven & Koedinger, 2002).

The Geometry Cognitive Tutor is a form of intelligent tutoring software. It was developed in our lab and is being used in hundreds of schools across the United States. This software (as well as the Cognitive Tutors used in the studies reported in subsequent sections of this chapter) is

based on the ACT-R theory of cognition and learning (Anderson & Lebière, 1998) as well as on a set of principles for the design of Cognitive Tutors (Koedinger & Corbett, 2006). The Cognitive Tutor software supports students as they acquire complex cognitive skill. It selects problems for students on an individual basis, and provides step-by-step guidance as students solve problems. The tutor's guidance consists of contextual hints (given at the student's request) about what to do next, feedback on correctness, and just-in-time messages that provide help with common errors. Cognitive Tutors have been shown to significantly increase high-school students' math achievement, compared to more typical classroom instruction (see the summary in Koedinger & Alevan, 2007).

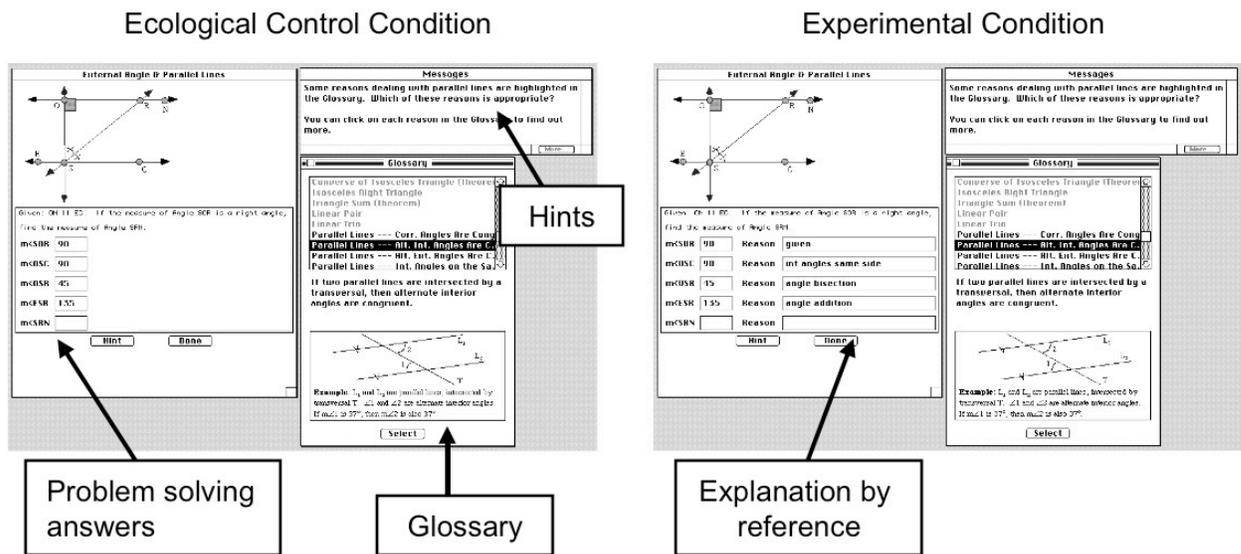


Figure 3. In the Experimental Condition, students explained their steps “by reference.” Otherwise, the tutor versions used in both conditions were the same.

The standard Geometry Cognitive Tutor served as the *ecological* control condition in the experiment, meaning that it represents standard practice in the field. We shall refer to it as the

Problem Solving Condition. For the treatment condition (Explanation Condition) we created a version of this tutor that was enhanced to support self-explanation but was otherwise the same (see Figure 3). For each problem-solving step, the tutor prompted the student to provide a reason, namely, the name of the theorem being used. The students could type the name of the reason if they remembered it, or they could select it from the tutor's on-line Glossary of geometry knowledge, a resource with information about the approximately 25 different theorems being targeted in the given tutor unit. For each theorem, there was a complete statement expressing the theorem and a simple example of how the theorem could be used in solving problems. Students could double-click on Glossary items in order to select them as the explanation for the current step. The tutor provided correctness feedback on the explanations, and if the student requested a hint, provided hints related to the correct explanation. The software required that a student get the explanations right before moving on to the next problem.

All students took a pre-test and post-test, with three kinds of test items. First, "Answer" items were designed to assess whether students could solve geometry problems of the same type that they encountered as they were working with the tutor. Second, "Reason" items were designed to assess whether students could explain their reasoning in the same manner they did when working with the tutor. The Answer and Reason items followed the same format as the corresponding steps in the tutor problems. (We note that the students in the Problem Solving condition did not practice explaining their reasoning as they worked with the tutor. For them, therefore, these post-test items were a form of transfer.) Third, "Not Enough Info" items tested whether students could recognize situations where there was insufficient information to infer the measure of a given quantity. These items were designed to expose shallow knowledge, because shallow strategies such as "if angles look the same, they are the same" will often lead to wrong answers on these

items. These items provide a measure of transfer, because the students in neither condition had encountered these types of items during their work with the tutor.

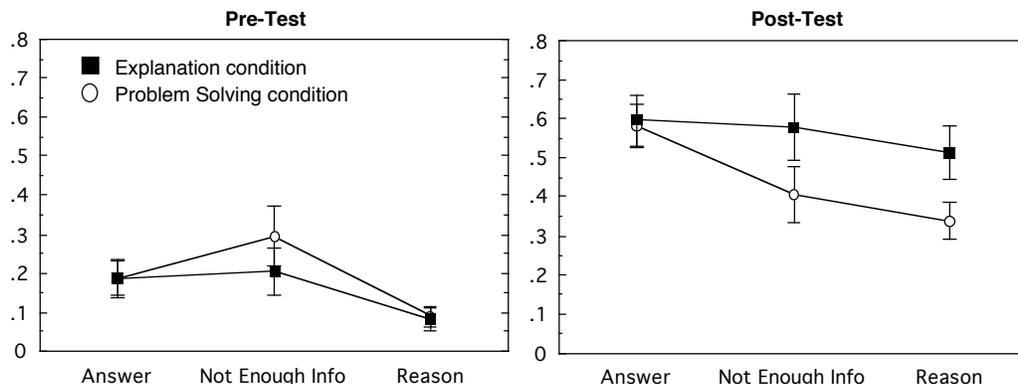


Figure 4. Pre- and post-test results for students in the second study, on each of the three types of test items. Multiple forms were used to counterbalance for test difficulty.

We found that students who had explained their steps during their work with the tutor showed greater gains with respect to Not Enough Info items and with Reason items than the students who solved problems without explaining their steps (see Figure 4). The performance on the Answer items did not differ between the two conditions, despite the fact that the students in the Explanation Condition had encountered only half as many of them during training as their counterparts in the Problem Solving Condition. (The students in the Explanation Condition spent a significant portion of their time explaining steps. Because we controlled for time-on-task, these students completed fewer problems as they worked with the tutor.)

Thus, the students who had explained their steps learned in a more robust manner, compared to students in an ecological control condition. They did better on two types of items that hinge on deeper understanding, namely, Reason items and Not Enough Info items. The Reason items tap conceptual knowledge, and the Not Enough Info are transfer items designed to expose

shallow knowledge. A further analysis suggested that students in the Explanation Condition acquired less shallow procedural knowledge and more flexible conceptual/declarative knowledge (Aleven & Koedinger, 2002).

Discussion of the Self-Explanation Support Studies

The experiment shows that supporting meta-cognition during instruction can improve robust learning of domain knowledge. We found that simple self-explanation prompting, with feedback on self-explanations, leads to robust learning. A limitation of the current study, addressed in some of the studies discussed below, is that we assessed domain (geometry) learning, not metacognitive improvement. We did not assess, for example, whether students were more likely to spontaneously self-explain at appropriate junctions during subsequent learning experiences. Nonetheless, the finding that even simple self-explanation support can be effective in intelligent tutoring software is of practical and theoretical interest, especially given that it shows improvement over a control condition that itself has been shown to improve upon typical classroom instruction (Koedinger & Aleven, 2007). Other researchers have since replicated the effectiveness of static menu-based support for self-explanation, combined with feedback on students' self-explanations (Atkinson, Renkl, & Merrill, 2003; Corbett, McLaughlin, Scarpinato, & Hadley, 2000; Corbett, Wagner, & Raspat, 2003).

Studies with interfaces that support free-form entry of self-explanations (i.e., students type in explanations) suggest that feedback on self-explanations may be an important factor. Aleven and colleagues created two versions of the Geometry Cognitive Tutor that requested free-form explanations from students, one that did not provide any feedback (Aleven & Koedinger, 2000b), and one that provided feedback through a natural language dialogue module (Aleven, Ogan, Popescu, Torrey, & Koedinger, 2004). In two studies, they compared these tutor versions against

menu-based self-explanation (i.e., against the experimental condition in the experiment described above). They found that prompts for free-form explanation without feedback are ineffective. Students largely ignore them and provide very few good explanations (Alevén & Koedinger, 2000b). With feedback, however, free-form explanations help students learn to state better explanations, with no detrimental effect on problem-solving skill or transfer, even though self-explanations take time away from solving problems per se (Alevén et al., 2004). In addition, Corbett and colleagues in a study with a Cognitive Tutor for Algebra II found that free-form explanations with feedback led to slightly (but reliably) better transfer (Corbett, Wagner, Lesgold, Ulrich, & Stevens, 2006).

Researchers have begun to experiment with simple adaptive forms of self-explanation support, but it is too early to tell what works and what does not. The adaptive support in Conati and VanLehn's (2000) system was effective but the study conducted did not isolate the effect of the adaptive nature of the support. Hausmann and Chi (2002) reported that computer-based prompts are as effective as human prompts when students type self-explanations into a computer interface without feedback. The prompts were yoked across the conditions, so the study suggests that adaptive prompts may not be necessary, at least when students do not receive feedback on their self-explanations. A study by Weerasinghe and Mitrovic (2005) reported that self-explanation prompts selected based on students' domain knowledge improve domain-level learning. Conati, Muldner, and Carenini (2006) point the way toward truly adaptive support. They developed a system that decides when to present prompts for self-explanation, and what type of self-explanation to prompt for, by trying to predict how much a given student will benefit from each type of self-explanation. In summary, a number of researchers are actively

investigating whether adaptive self-explanation support can help students learn more robustly at the domain level and become better future learners.

2. Tutoring Error Self-Correction

Our second line of research illustrating intelligent tutor support for metacognition focuses on the effect of supporting students in error self-correction. This research program grew out of a reframing of the debate over feedback timing. Some researchers have demonstrated and argued for benefits of delayed feedback over immediate feedback in instruction and training (Schmidt & Bjork, 1992). An intuition behind this argument is that delayed feedback leaves room for the student to engage in self-detection and correction of their own errors. Other research (Corbett & Anderson, 2001) has shown benefits of immediate feedback, particularly in enhancing the efficiency of learning while maintaining learning outcomes. Mathan and Koedinger (2005) reframed this feedback debate by suggesting that perhaps the more relevant difference should be in the nature of the “model of desired performance” or the instructional objectives. One may want to produce error-free efficient expertise or, alternatively, one may want to produce flexible problem solvers (also called “intelligent novices” – Bruer, 1993) who may make some initial errors but who, unlike novices, eventually detect and correct these errors. In the context of an intelligent tutoring system, the immediate feedback given relative to an intelligent novice model of desired performance will actually be similar to delayed feedback given relative to an expert model of desired performance, as both will wait to give feedback on errors. However, there is one key difference. Namely, unlike delayed feedback, tutoring with the goal of producing intelligent novices involves providing assessment/monitoring, feedback, and hints at the metacognitive level that guide, as needed, the process of error detection and correction. That is, this kind of tutoring constitutes an *adaptive* form of metacognitive support, as defined above.

Error self-correction requires students to *reflect* on their performance, on the outcomes of that performance, how those outcomes are different from desired outcomes, and, most importantly, on the rationale or reasoning for the initial performance attempt and how that reasoning might be modified in order to revise the performance and achieve the desired outcome. Other researchers have also focused on the metacognitive processes of monitoring one's performance and progress, to check for errors or evaluate level of understanding (Flavell, 1979; Palinscar & Brown, 1984; Pintrich, 2004; Schoenfeld, 1983; White & Frederiksen, 1998). This project focused particularly on supporting reflective reasoning after an error has been identified. During such reflection learners think about their thinking – about *why* they decided to do something and about what might be wrong (and thus can be fixed) in that reasoning process. This study illustrates this process in the context of Excel spreadsheet programming and in particular the use of relative and absolute references (explained in Figure 5).

	A	B
1		Hourly Wage
2	Hours Worked	\$10
3	20	=A3*B2
4	30	=A4*B3
5	40	=A5*B4

(a)

	A	B
1		Hourly Wage
2	Hours Worked	\$10
3	20	\$200
4	30	6000
5	40	240000

(b) ✗

	A	B
1		Hourly Wage
2	Hours Worked	\$10
3	20	=A3*B\$2
4	30	=A4*B\$2
5	40	=A5*B\$2

(c)

	A	B
1		Hourly Wage
2	Hours Worked	\$10
3	20	\$200
4	30	\$300
5	40	\$400

(d) ✓

Figure 5. Two different attempts at the task of writing a formula to multiply a fixed hourly wage (in cell B2) by different hours worked (in cells A3, A4 and A5) to produce the amounts earned (cells B3, B4, and B5). Panel (a) shows the entry of an incorrect formula in cell B3 and what the resulting formulas look like after the original is copied into cells B4 and B5. When a formula in Excel is copied and then pasted in new cells, the cell references are updated “relative” to the original position. This produces the wrong result as seen in panel (b). However, if a “\$” is placed in front of a column letter or row number (panel c), the reference is “absolute” and will stay the same when copied (when a reference to B\$2 is copied from B3 to B4, it does not change to B\$3, but remains B\$2). The result shown in panel (d) is now correct.

Notice in Figure 5a and 5b how when the formula is copied in cell B4, the formula produces \$6000 rather than the desired result of \$300. Excel performs “relative referencing” such that, for instance, if a referenced cell is above the formula cell in the original formula (as B2 is in the formula in B3), then it will remain above it in the copied cell (the reference will become B3 when the formula is copied into cell B4). In contrast, in Figure 5c, a correction to the formula is made employing “absolute referencing” by adding “\$” in front of the 2 in B2. Thus, the correct result is achieved as shown in 5d. How might a learner go about reasoning reflectively about the error shown in 5b?

	A	B
1		Loan Amount
2		\$10,000
3		
4	Interest Rate	Interest Owed
5	1%	\$100
6	5%	
7	10%	
8	15%	
9		

(a)

Question 2 of 3

What type of references change when you paste a formula into cells in other rows?

- Column References will change (B in B2, A in A5)
- Row References will change (2, 5)
- Both column and row references will change
- None of the references will change
- I dont know

(b)

	A	B	C	D
1		Loan Amount		
2		\$10,000		
3				
4	Interest Rate	Interest Owed		
5	1%	\$100		
6	5%	\$0		
7	10%	#VALUE!		
8	15%	\$15		
9				

(c)

What went wrong with the copy and paste?

- 1 The formula in the highlighted cell is multiplying the bright red cells!
- 2 Which cells should your formula be multiplying instead?

You got =A6*B3 in B6 Select the incorrect parts of the formula:

- A
- 6
- B
- 3

OK

(d)

Figure 6. At the point of initial error (a), the Expert tutor provides hints as shown in (b). In the Intelligent Novice tutor, however, it is at the later point after copying and pasting the formula (c) that the Intelligent Novice hints are given (d). Note the different character of the hint messages (b) and (d). The Intelligent Novice hints (d) help students understand the reasoning behind a correct formula by analyzing what went wrong with incorrect formula entry.

Experiments on Tutoring Self-Correction

Figure 6 illustrates the difference both in timing and content of the feedback and hints between the Expert (EX) tutor condition (see panels a and b) and the Intelligent Novice (IN) tutor condition (see panels c and d). The experiment performed, while not run in the context of a real course, had a number of features of *in vivo* experimentation including ecological validity factors: real content, appropriate students (temporary employment workers whose skills could be enhanced through participation), realistic duration (nearly 3 hours of instruction), and internal validity factors (single principle variation and random assignment). In addition to using a post-test immediately following instruction and using items isomorphic to the training, robust learning measures were also used including measures of long-term retention (one week later) and of conceptual and procedural transfer. The results demonstrated benefits for the Intelligent Novice tutor condition on all measures (Mathan & Koedinger, 2005). For instance, Intelligent Novice (IN) students performed significantly (and statistically reliably) better (at 74% correct) on the delayed transfer test than students in the Expert (EX) condition (at 60% correct). IN students were also significantly better (88% vs. 79%) on an immediate test involving Excel formula programming items that were isomorphic to those used in the instruction. Learning curve analysis of log data showed that the effect of the treatment made a difference quite early in the instruction (Martin, Koedinger, Mitrovic, & Mathan, 2005). This result suggests that error correction support was more relevant to students' initial declarative understanding than later refinement through practice.

Discussion of Tutoring Error Self-Correction

The early differences in favor of the IN condition (nominally delayed feedback) are inconsistent with predictions of Bjork's (1994) "desirable difficulties" notion that suggests a trade-off

whereby delayed feedback should hurt immediate performance (relative to immediate feedback), but enhance long-term learning. This inconsistency may be attributed to the fact that the Intelligent Novice intervention is not simply delaying feedback but providing direct assistance for error correction reasoning when needed. There may also be differences between Bjork's prior results and ours because of the simpler, motor-skill oriented domains in that work in contrast to the more complex and semantically-rich domain addressed here. The effectiveness of the intervention presented here appears to be in how it helps students reason to a better understanding of the required inferences and not in improving a general student ability to self-detect and fix errors. Metacognitive support from the tutor helps students to reason about their errors. By reasoning about correct procedures in contrast to incorrect ones, students appear to develop a better conceptual understanding of those procedures. We suspect that if students had simply been given delayed feedback, without the support for reasoning about error correction (as illustrated in Figure 6d), we would not see the positive outcomes we did. We do note that detecting and making sense of errors in this task domain appears relatively easy for students (even if learning the procedures is not). Intelligent Novice error correction feedback may not work as well in other domains where error detection or error understanding is more difficult.

The Intelligent Novice feedback approach has interesting similarities with support for self-explanation in that it directs students toward the deep relevant features of the domain and away from the shallow irrelevant features (see learnlab.org/research/wiki/index.php/Features). Students are supported to think about what aspects of the formula may change in unwanted ways if an absolute reference (\$) is not used. Such a deeper encoding leads to better transfer than a surface or shallow encoding, for instance, thinking in terms of row and column movements and whether to put the \$ before the letter or the number. Helping students make the contrast between

incorrect and correct performance seems to support deep encoding. It is similar to variations on self-explanation in which students are given an incorrect performance example (ideally a common or likely one) and asked to explain why it is wrong (Grosse & Renkl, in press; Siegler 2002).

3. Tutoring to Reduce Gaming the System

Tutoring metacognition does not only require guiding students to learn to choose appropriate learning and performance strategies. It also requires guiding students to learn to not choose inappropriate learning and performance strategies. Consider a student using learning software, who does not know how to correctly solve the current problem step or task. This student might adopt the appropriate metacognitive strategy of seeking help from the software, the teacher, or another student, and then self-explaining that help. However, the student may instead choose a less appropriate strategy: “gaming the system.”

We define gaming the system as “attempting to succeed in an interactive learning environment by exploiting properties of the system rather than by learning the material” (Baker et al, 2006). Gaming behaviors have been observed in a variety of types of learning environments, from intelligent tutors (Schofield, 1995), to educational games (Magnussen & Misfeldt, 2004) to online course discussion forums (Cheng & Vassileva, 2005). Gaming behaviors have been found to be associated with significantly poorer learning in intelligent tutors (Baker, Corbett, Koedinger, & Wagner, 2004). Additionally, there is some evidence that gaming the system on steps that the student does not know (termed “harmful gaming”) is associated with poorer learning outcomes than gaming the system on steps the student already knows (termed “non-harmful gaming”) (Baker, Corbett, Roll, & Koedinger, 2008). Within intelligent tutoring systems, gaming the system generally consists of the following behaviors:

1. quickly and repeatedly asking for help until the tutor gives the student the correct answer (Wood & Wood, 1999), and
2. inputting answers quickly and systematically. For instance, systematically guessing numbers in order (1,2,3,4...) or clicking every checkbox within a set of multiple-choice answers, until the tutor identifies a correct answer and allows the student to advance.

In this section, we discuss a pedagogical agent, Scooter the Tutor, designed to reduce gaming and increase gaming students' learning (Baker et al, 2006). Scooter the Tutor was built around a machine-learned detector of gaming behavior, which was shown to accurately determine how much each student games the system and on which problem steps the student games (Baker et al., 2008). This detector was able to distinguish between the two types of gaming behavior mentioned earlier, harmful and non-harmful gaming, enabling Scooter to focus solely on responding to harmful gaming, the behavior associated with poorer learning. This detector has also been shown to effectively transfer between different tutor lessons with little reduction in effectiveness (Baker et al., 2008).

Scooter was designed to both reduce the incentive to game, and to help students learn the material that they were avoiding by gaming, while affecting non-gaming students as minimally as possible. Scooter was built using graphics from the Microsoft Office Assistant, but with modifications to enable Scooter to display a wider range of emotion. During the student's usage of the tutoring system, Scooter responds to gaming behavior in two ways: via emotional expressions and supplementary exercises.

When the student is not gaming, Scooter looks happy and occasionally gives the student positive messages (see the top-left of Figure 7). Scooter's behavior changes when the student is

detected to be gaming harmfully. If the detector assesses that the student has been gaming harmfully, but the student has not yet obtained the answer, Scooter displays increasing levels of displeasure (culminating in the expression shown on the bottom-left of Figure 7), to signal to the student that he or she should now stop gaming, and try to get the answer in a more appropriate fashion.

If the student obtains a correct answer through gaming, Scooter gives the student a set of supplementary exercises designed to give the student another chance to cover the material that the student bypassed by gaming this step, shown in figure 7. Within supplementary exercises, the student is asked to answer a question. This question may require understanding one of the concepts required to answer the step the student gamed through, or may require understanding the role the gamed-through step plays in the overall problem-solving process. If the student tries to game a supplementary exercise, Scooter displays anger.

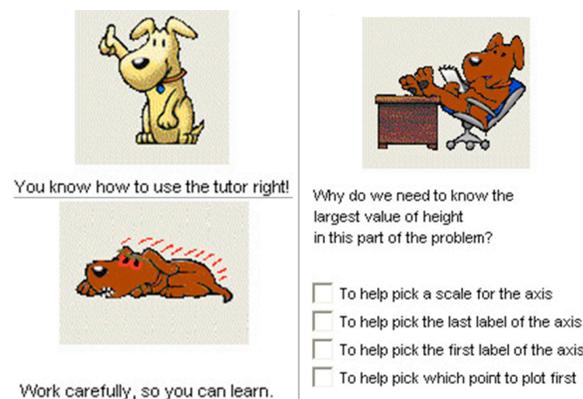


Figure 7. Scooter the Tutor – looking happy when the student has not been gaming harmfully (top-left), giving a supplementary exercise to a gaming student (right), and looking angry when the student is believed to have been gaming heavily, or attempted to game Scooter during a supplementary exercise (bottom-left).

Study on Effects of Tutoring Students not to Game the System

We studied Scooter's effectiveness in an *in vivo* experiment in the context of a year-long Cognitive Tutor curriculum for middle school mathematics (Koedinger & Corbett, 2006), within 5 classes at 2 schools in the Pittsburgh suburbs. The study was conducted in the spring semester, after students had already used the Cognitive Tutor for several months. Each student used a tutor lesson on scatterplots. The control condition and experimental condition occurred during different weeks – students not using the scatterplot tutor used a different tutor lesson on another subject. 51 students participated in the experimental condition for the scatterplot lesson (12 were absent for either the pre-test or post-test, and thus their data will not be included in analyses relevant to learning gains); 51 students participated in the control condition for the scatterplot lesson (17 were absent for either the pre-test or post-test).

Before using the tutor, all students first viewed instruction on domain concepts, delivered via a PowerPoint presentation with voiceover and simple animations. In the experimental condition, a brief description of Scooter, and how he would respond to gaming, was incorporated into the instruction. Then students completed a pre-test, used the tutor lesson for 80 minutes across multiple class periods, and completed a post-test. Test items were counter-balanced across the pre-test and post-test. Log files were used to distill measures of Scooter's interactions with each student, including the frequency with which Scooter got angry, and the frequency with which Scooter gave a student supplementary exercises.

Observational data was collected to determine each student's frequency of gaming, using quantitative field observations (systematic observations of student behavior by field observers – cf. Baker, Corbett, Koedinger, & Wagner, 2004), in order to analyze Scooter's effects on gaming

frequency. Another potential measure of how much each student gamed, the gaming detector, was not used because of risk of bias in using the same metric both to drive interventions and as a measure of the intervention's effectiveness.

Results of Intervention on Metacognitive Behavior and Learning

The Scooter intervention was associated with a sizeable, though only marginally significant, reduction in the frequency of observed gaming. 33% of students were seen gaming in the control condition (using quantitative field observations), while 18% of students were seen gaming in the experimental condition. However, although fewer students gamed, those students who did game did not appear to game less (14% in the experimental condition, 17% in the control condition).

In terms of domain learning, there was not an overall between-condition effect. However, only a minority of students received a substantial number of supplemental exercise interventions from Scooter (because only a minority of students gamed the system, as in previous studies of gaming). There is some evidence that the intervention may have had an effect on these specific students. In particular, the supplemental exercises appeared to be associated with significantly better domain learning. The third of students (out of the overall sample) that received the most supplementary exercises had significantly better learning than the other two thirds of the students, as shown in Figure 8. Students who received the most supplementary exercises started out behind the rest of the class, but caught up by the post-test (see Figure 9 Left). By contrast, in both the control condition (see Figure 9 Right) and in prior studies with the same tutor, frequent harmful gaming is associated with starting out lower than the rest of the class, and falling further behind by the post-test, rather than catching up.

The emotional expressions, on the other hand, were not associated with better or worse learning. Students who received more expressions of anger did not have a larger average learning gain than other students.

There was no evidence that students reduced their degree of gaming after receiving either type of intervention (according to the quantitative field observations). Hence, the observed reduction in gaming may have been from Scooter's simple presence. Students who chose to game knowing that Scooter was there did not appear to reduce their gaming.

Given the connection between receiving the supplementary exercises and learning, it is surprising that there was not an overall learning effect for Scooter. One potential explanation is that students who ceased gaming chose other ineffective learning strategies instead. The lack of connection between reduced gaming and improved learning may indicate that gaming is not directly causally related with learning. The apparent benefits of supplementary exercises may be from the "variable encoding" (Paas & Van Merriënboer, 1994) that students experienced in the different kinds of presentation of the target knowledge in the original tutor and the supplementary exercises.

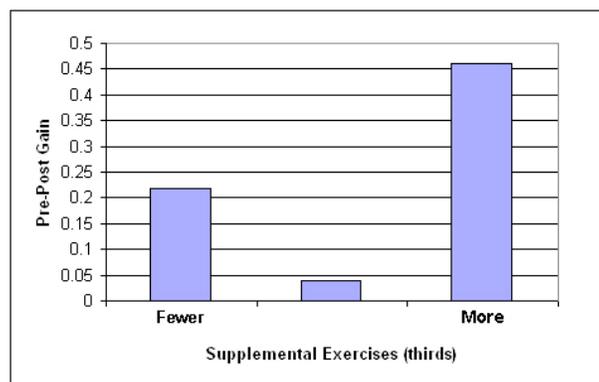


Figure 8. The learning gains associated with receiving different levels of supplemental exercises from scooter

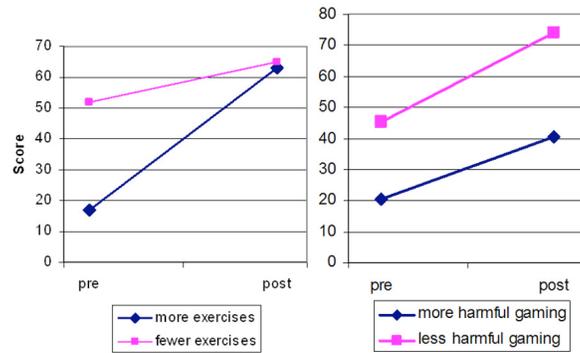


Figure 9. Left: The pre-test and post-test scores associated with receiving different levels of supplemental exercises from scooter (top third versus other two thirds). Right: The pre-test and post-test scores associated with different levels of harmful gaming in the control condition (top half of harmful gaming versus other students)

Recently, there have been two other systems designed to reduce gaming and improve gaming students' learning. Walonoski and Heffernan (2006) displayed sophisticated visualizations about gaming behavior to both students and teachers, on the student's screen. Arroyo et al.'s (2007) ProgressTips gave detailed meta-cognitive messages to gaming students about appropriate meta-cognitive behavior between problems. Walonoski and Heffernan's gaming visualizations, like Scooter, reduced gaming. Gaming was even reduced in future units where gaming visualizations were not given, showing some durability of the result. However, Walonoski and Heffernan did not measure domain learning. Arroyo et al.'s ProgressTips did not reduce gaming – instead, they caused students to shift gaming behavior (personal communication, Ivon Arroyo), as in Murray and VanLehn (2005). Domain learning, however, was improved by ProgressTips. ProgressTips also improved students' attitudes towards the tutor and learning domain, a positive effect not obtained with Scooter, who was generally disliked by students who received his interventions (Baker, 2005). One possible explanation is that ProgressTips disrupted students' learning

experiences less than Scooter, who interrupted student behavior as soon as a student completed a step through gaming.

In general, the results of studies on these three systems to respond to gaming suggest that interventions given for differences in students' meta-cognitive behaviors can reduce the incidence of inappropriate behaviors and improve learning, whether they are given at the time of behavior or shortly afterwards.

4. Tutoring Help Seeking

The ability to seek help adaptively is a key metacognitive skill that figures prominently in theories of self-regulated learning (Newman, 1994), specifically as an important resource management strategy (Pintrich, 1999). Research on help seeking in social settings (e.g., classrooms) shows that adaptive help-seeking behavior can be an effective route to independent mastery of skills (Karabenick & Newman, 2006). Intelligent tutoring systems are an interesting context to investigate help seeking since they typically have sophisticated on-demand help facilities and they offer an opportunity for very detailed analysis of the usage patterns of these facilities. As part of the step-by-step guidance that these systems offer, students can typically request multiple levels of tailored help on what to do next at any point during their problem-solving activities. This form of help is thought to be beneficial for learning (e.g. Wood & Wood, 1999). As mentioned, the particular system that we worked with, the Geometry Cognitive Tutor, also provides an on-line Glossary of geometry knowledge that lets students browse descriptions and examples of the geometry theorems that they are learning.

Upon closer scrutiny, however, the assumption that students' learning benefits from using these facilities turns out to be problematic. We found that students often do not use help facilities effectively (Alevan & Koedinger, 2000a; Alevan, Stahl, Schworm, Fischer, & Wallace, 2003).

This finding mirrors findings by researchers who study help seeking in social settings (Arbreton, 1998; Newman 2002.) With surprising frequency, students abuse the tutor's help, focusing on help levels that essentially give away the next step, all but ignoring the help levels that provide explanations of why the answer is the way it is.

This behavior appears to generally be maladaptive, metacognitively, and has been shown to be associated with poorer learning (Alevan, McLaren, Roll & Koedinger, 2006; Baker, Corbett, Koedinger, & Wagner, 2004; Baker, Corbett, Roll & Koedinger, 2008). However, it appears that for a small number of students the rapid clicking through hints represents a way of efficiently turning a problem step into an example step. The student may then self-explain the step, in the best case reconstructing the tutor's explanations. Providing support for this explanation, a recent data mining study examining tutor logs found that spending large amounts of time on a bottom-out hint is positively correlated with learning (Shih, Koedinger, & Scheines, 2008).

Another form of maladaptive help-seeking behavior seen in our research is when students resist using help even when they clearly seem to need it, such as after multiple errors on a step. For example, Alevan and Koedinger (2000a) found that after several consecutive errors on a step with no hints, students were more likely to try again rather than ask for a hint. In addition, the students used the tutor's glossary very rarely. On the other hand, when requesting a hint, students asked to see the bottom-out hint on 82% of their hint episodes. To summarize, students either ask for 'instant' help or no help at all, but tend to avoid more complex help seeking episodes.

Studies to Evaluate Tutoring of Help Seeking

Given the high frequency of maladaptive help seeking, we embarked on research to test the hypothesis that a Cognitive Tutor agent that provides guidance with respect to students' help-seeking behavior can help students to both learn better at the domain level and become better

help seekers. In other words, this fourth project focuses on improving one aspect of students' meta-cognitive abilities, their ways of seeking help, and thereby their current and future learning. In this sense, this project is more ambitious than the other three projects described above: the goal is not just to “channel” students into a particular meta-cognitive behavior with the tutor, but to help students internalize a meta-cognitive behavior that transfers to future tutor use and even to other learning environments. Furthermore, the help-seeking project avoids using any domain-specific assumptions or concepts, and thus maintains a metacognitive character, making it possible to apply the underlying model of help seeking to different domains without much adaptation. That is, this project offers externally regulating tutoring with the goal of helping students internalize the productive strategies and better self-regulate their learning in the supported environment and beyond. In doing so, it targets all four goals as described in Figure 1.

As a first step, we developed a model that aims to capture both effective and ineffective help-seeking behavior (Alevin et al., 2006). In contrast to the previous project (Scooter), in which a model was built using machine learning, the model was built by hand, and implemented as a set of production rules. Lacking the guidance of a detailed normative theory of help seeking, we made extensive use of student-tutor log data and theoretical cognitive task analysis to design the model. For example, the initial version of our model prescribed that students should always take their time before attempting a step. When mining log data of student-tutor interactions, we noticed that fast actions by students on steps on which they are skilled correlate with large learning gains, a finding clearly at odds with our initial model. We therefore updated the model to allow fast attempts (including incorrect ones) on steps where a student is skilled, subject to limitations such as number of overall errors on this step. The model, (as summarized in flow-chart format in Figure 10), stipulates that students should work deliberately, spending adequate

time reading problem statements and hints, and that they should use help in one of three cases: when steps are not familiar to them; when they do not have a clear sense of what to do; and when they have made an error (as indicated by the tutor's domain-level feedback) that they do not know how to fix. The choice of what source of help to use, the tutor's on-demand hints or the glossary, should be driven by the student's self-assessed level of relevant knowledge. The less familiar a step is, the more contextualized the help requested should be. (We view the glossary as decontextual help and the hints as highly contextual help.)

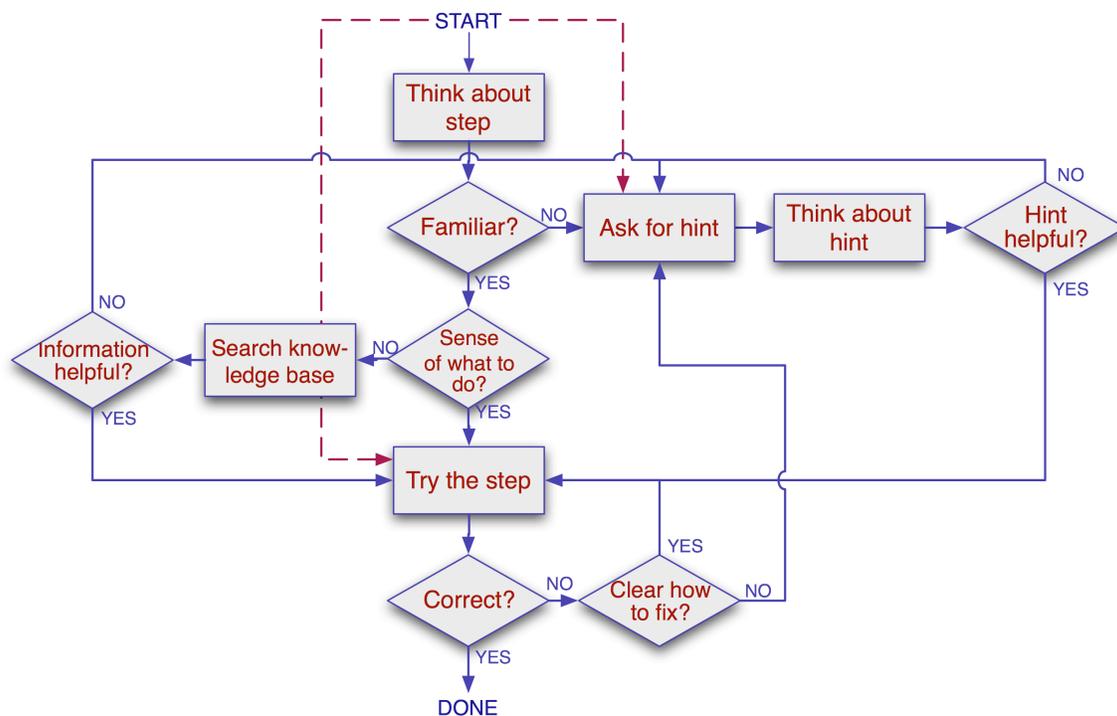


Figure 10. Schematic view of the Help Seeking Model

Essentially, this model is a detailed normative theory of help seeking with an intelligent tutoring system. It specifies in a detailed, moment-by-moment manner what a student should do in any given situation. Notably, the model does not limit the student to a single learning trajectory. Rather, it allows for a wide variety of reasonable actions at each point, while

excluding actions that were clearly shown to be ineffective in our modeling and data mining activities.

Data from our studies suggest that the help-seeking behavior captured by the model is associated with higher learning gains. In several analyses we have conducted, students who demonstrated poorer help-seeking behavior (i.e., students whose help-seeking behavior conformed to the model) were found to have poorer pre-post learning gains, though the result was not entirely stable across data sets (Roll et al., 2005).

There is evidence that the model captures help-seeking behavior that is independent of the specific content or group of students. The model correlates well across different cohorts of students and tutors within the Cognitive Tutor family (Roll et al., 2005). We found that students tend to make the same types of help-seeking errors between Cognitive Tutors – the correlation between students' frequency of different types of help-seeking errors in an angles unit (in geometry) and a scatterplot unit (in data analysis) was 0.89.

Next, we created the “Help Tutor” agent, a Cognitive Tutor at the meta-cognitive level. Driven by the help-seeking model, it provides context-sensitive feedback on students' help-seeking behavior, as they work with a Cognitive Tutor. Student behavior that, in a given context, matches the normative predictions of the metacognitive model specific to that context, is deemed metacognitively correct. Student behavior that matches any of the many metacognitive “bug rules” in the model (which capture what is believed to be inappropriate help-seeking behavior) were deemed metacognitively incorrect. The Help Tutor displays a feedback message in response to such errors, as shown in Figure 11, “Perhaps you should ask for a hint, as this step might be a bit difficult for you,” or “It may not seem like a big deal, but hurrying through these steps may lead to later errors. Try to slow down.” Perhaps somewhat unfortunately, in retrospect,

while the Help Tutor pointed out help-seeking behavior that was not likely to be productive, it never praised students for being good help seekers.

The screenshot shows a software interface with a window titled "Scenario". Inside, there is a text box describing a fuel tank problem: "The fuel tank of an automobile has a capacity of 12 gallons. The fuel gauge is shown here. As the indicator moves from the empty position to the full position, it sweeps a 120 degree angle." Below the text is a fuel gauge diagram with an arrow pointing to a position between 'E' and 'F'. A "Hint" dialog box is open, containing the text: "No need to hurry so much. Take your time and read the hint carefully. Consider trying to solve this step without another hint. You should be able to." Below the hint are navigation buttons: "<<<", ">>>", and "OK".

1. If the indicator sweeps 30 degrees, how many gallons of gasoline have been added to the fuel tank?

angle	30	Reason	Given
gas	3	Reason	

Figure 11. Example of a help seeking error message.

We integrated this metacognitive tutor agent into the Geometry Cognitive Tutor, so that students received guidance both with respect to geometry and with respect to their help-seeking behavior. The cognitive and metacognitive agents were not always in agreement; for example, an answer may constitute an error at the domain level, but attempting the step was still a metacognitively-appropriate action; likewise, a student may obtain a correct answer by guessing, metacognitively-inappropriate behavior. A prioritizing algorithm was implemented to choose the

more informative feedback in cases where both cognitive and metacognitive feedback was appropriate.

We conducted two classroom experiments to evaluate the effects of the kind of metacognitive feedback generated by the Help Tutor (Roll et al., 2007a). The first study compared the traditional Geometry Cognitive Tutor to a version that included the Help Tutor, integrated with the Cognitive Tutor as described above. In the second study the Help Tutor was the main component of a broader metacognitive instructional package that also included declarative classroom instruction on help-seeking principles and short self-assessment activities supported by automated tutors. The classroom instruction included short video segments illustrating how to use the tutor's help facilities effectively. In the self-assessment activities, the students were asked to rate their ability to apply a new geometry theorem prior to solving their first problem involving that theorem, and then were asked to reflect on the correctness of their prediction (Roll et al., 2007a). In both experiments, the students in the control condition worked with the standard Geometry Cognitive Tutor. In both experiments, we tested the hypotheses that the respective metacognitive instruction (the Help Tutor in the first experiment, and the metacognitive instructional package in the second experiment) would lead to more desirable help-seeking behavior, both during practice with the tutor (goal 1 in Figure 1) and in a paper-and-pencil transfer environment after tutor usage (goal 3 in Figure 1), and that it would lead to better domain-level learning (goal 2 in Figure 1).

The studies showed mixed results. The evidence seemed to confirm the first hypothesis, that help-seeking behavior in the tutor would be improved: there was evidence that students sought help more appropriately under the Help Tutor's tutelage, as measured by the percentage of actions that conformed to the help-seeking model. However, the improvement was seen with

respect to only a subset of help-seeking action types. There was no difference between the conditions in terms of help-seeking choices made on the first action on a new problem step before any Help Tutor feedback was seen. This finding may suggest that the improvement in students' metacognitive behavior was mainly the result of complying with the Help Tutor messages rather than of students' internalizing the metacognitive support and making proactive choices.

With respect to the second hypothesis, we found no lasting effect on students' help-seeking procedural knowledge, as measured by post-test scores on items with embedded hints, compared to performance on items with no embedded hints. We did, however, find that students who used the Help Tutor had a better declarative understanding of help-seeking principles, as measured by the quality of their answers to hypothetical help-seeking scenarios.

The evidence did not support the third hypothesis, that the metacognitive instruction would lead to better domain-level learning. Although students in both conditions improved significantly from pre- to post-test in both studies with respect to their geometry knowledge and skill, we did not find any differences between the conditions.

Discussion of Studies of Tutoring Help Seeking

In sum, only one of the three hypotheses was confirmed, namely, that the Help Tutor would lead to more productive help-seeking behavior. However, these studies have several limitations. For one, we did not measure help-seeking behavior in the subsequent unit within the same tutoring environment. That is, we did not address the fourth of the goals of metacognitive tutoring, depicted in Figure 1. While the improved help-seeking behavior did not transfer to the paper-and-pencil environment, it may have transferred to the subsequent unit within the Geometry

Cognitive Tutor, which uses similar interface elements, problem types, learning goals, and requires similar learning strategies (as in Walonoski & Heffernan, 2006).

Interpretation of the results should take into account the context of the metacognitive instruction. Our studies focused on a specific type of learning environment, one that supports step-by-step problem solving with immediate feedback and hints available on demand. Such environments may encourage certain types of metacognitive behaviors and discourage others. For example, in the Cognitive Tutors we used, domain-level knowledge tracing (i.e., the Bayesian method by which the tutor tracks the student's detailed knowledge growth over time – Corbett & Anderson, 1995) uses only information from students' first action on each problem step. Therefore, students may be tempted to always enter the step before requesting help, even if they are essentially guessing, because they will receive full credit for a correct answer, and if their answer is wrong they will not be penalized more than if they had requested a hint. These factors may promote an interaction style on the part of students that is specific to the particular learning software. It may be that in other environments, in which the advantages of productive help-seeking strategies are more apparent to students, help-seeking instruction will be more readily internalized by students.

Students' attitudes towards the Help Tutor may also help explain their behavior. Based on several interviews we conducted, and based on informal feedback, it appears that students did not like the system's commenting on their help-seeking behavior, even though they often agreed with the feedback. (A similar pattern was also observed with Scooter the Tutor, described above.) Even though students believed that the Help Tutor's suggestions were probably correct, they did not see its advice as valuable, and thus they may have merely complied with it, rather than internalizing the advice. It could be said that the students applied their existing help-seeking

strategies to the Help Tutor itself. They were used to ignoring intermediate hints, and thus tended to ignore the Help Tutor. (This explanation is consistent with the lack of metacognitive improvement on first attempts before any Help Tutor feedback.)

The results of our experiment strongly suggest that many students, in spite of being aware of appropriate help-seeking strategies, choose not to apply those strategies. The students appeared to know the difference between ideal and faulty help-seeking behavior: they reported agreeing with the Help Tutor comments, and, compared with the control group students, demonstrated better conceptual understanding of help seeking following usage of the Help Tutor and receiving the declarative help-seeking instruction. But their actual help-seeking behavior, on the post-test, was no better than the control condition students' help-seeking behavior. It did not reflect their superior declarative knowledge of help-seeking. Apparently, students did not *want* to apply this knowledge. Hence, in order to create a learning environment that effectively promotes help seeking, we need to better understand the motivational and affective factors that shape students' help-seeking behavior and their desire to be (or become) effective help seekers and effective learners.

Relations between Gaming and Affect, Motivation, and Attitudes

Why do students choose to engage in inappropriate learning behaviors or strategies? In particular, why do students choose to game the system, which is clearly an ineffective learning strategy? Answering this question may help in the development of future interventions that address gaming, in a more effective fashion than work up to this point has.

This broad question led us to investigate the relationship between gaming the system and several potential factors that could influence the choice to game, including affect, motivation, attitudes, and differences between tutor lessons. In this section, we briefly summarize six studies

that we have conducted, in collaboration with our colleagues. These studies are reported in full detail in (Baker, 2007; Baker, Walonoski, Heffernan, Roll, Corbett, & Koedinger, 2008; Rodrigo et al, 2007, 2008). In these studies, we correlated the frequency of gaming the system (measured either by behavioral observation or the gaming detector) and data from questionnaires and affect observations, with junior high school and high school students in the USA and Philippines. Five of six studies were conducted using intelligent tutoring systems – the sixth study involved an educational game.

One of the most common hypotheses for why students game (e.g. Martinez-Miron, du Boulay, & Luckin, 2004; Baker, Corbett, Koedinger, & Wagner, 2004) is that students game because they have performance goals rather than learning goals. Two studies investigated this hypothesis – both found no such relationship, a surprising result especially given the finding by Pintrich (1999) that performance goals (termed extrinsic goals in this work) are associated with self-reports of not using self-regulated learning strategies. Items that indicated that the student had relatively low desire to persist in educational tasks were weakly associated with gaming, with r^2 under 0.05.

Another popular hypothesis among teacher (Baker et al., 2008) is that gaming the system is associated with anxiety – however, two studies found no relationship between gaming and anxiety.

Negative attitudes towards computers, the learning software, and mathematics, were each found to be correlated with gaming in one to three of the studies. These effects were statistically significant, but were fairly weak, with r^2 under 0.05.

The affective states of boredom and confusion were associated with the future choice to game. In two studies, a student's affect and behavior were repeatedly observed. In both studies,

there was evidence that a bored student was more likely (about twice as likely) to begin gaming in the next three minutes, particularly among students who were frequently bored. There was a trend towards more gaming after confusion within one study, and a trend in the opposite direction in the other study. Gaming the system's future probability was not significantly increased by other affective states, including frustration, delight, surprise, and the flow state.

Hence, there were some fairly solid relationships between momentary affect and gaming, but only weak relationships between relatively stable attitudinal/motivational constructs and gaming. This finding led us to study, using data mining, how predictive are these semi-stable student attributes. The detector of harmful gaming was applied to each student's behavior in each tutor unit in an entire year of middle school-level tutor data, and the amount of variance in the student terms predicted as a class was used as a proxy, and an upper bound, for the total amount of variance that could be predicted by the sum of all student attributes that remain stable over a year. The result indicated that differences between lessons were much better predictors of how much a student would game ($r^2 = 0.55$) than differences between students ($r^2 = 0.16$).

Overall, then, these results suggest that several factors contribute to a student's decision to game the system. Semi-stable student characteristics, in particular attitudes towards the domain, play a small role in the choice to game the system. A larger role appears to be played by affect, with a student's momentary experience of boredom or confusion leading students in many cases to game the system shortly afterwards. This suggests that affective learning companions (Kort, Reilly, & Picard, 2001) that respond effectively to these affective states may in some cases be able to prevent gaming the system from occurring – it remains for future work to determine which affective responses might be effective in this regard. Another important factor is differences between tutor lessons – studying in greater depth how the differences between

lessons increase or decrease gaming, and whether this effect is mediated by affect, may also help us to develop tutor lessons that students do not choose to game.

General Discussion and Conclusion

Within this chapter, we have discussed our research group's work to build tutorial support for metacognition, giving examples of interventions in the following four areas:

1. Self-explanation
2. Error self-correction
3. Reduction of gaming the system
4. Help-seeking skills

While we explored these metacognitive abilities within the context of intelligent tutoring systems, the results of the experiments may be relevant to other forms of instructional support for metacognition. The two first systems have achieved reliable domain-level effects on robust learning measures. The last two projects, on the other hand, had more modest or unstable effects on learning. In this discussion, we raise several hypotheses to explain the differences found between the outcomes of the given support and highlight key achievements and challenges.

Analysis of the Interventions

The four interventions differ along a number of dimensions, most notably in terms of how adaptive their metacognitive support is. As mentioned earlier, *static* support does not change depending on the students' behavior or knowledge; instead, it occurs during certain fixed stages in the learning task, regardless of student metacognition. In contrast, *adaptive* metacognitive support assesses aspects of students' metacognitive behavior and adapts the software's behavior or response, based on its assessment of the student's metacognitive behavior. Systems that offer adaptive support for metacognition typically allow students the freedom to commit

metacognitive errors, and respond to these errors when they recognize them. Such systems may of course also respond when students engage in metacognitive behaviors deemed to be desirable or effective.

Under this definition, the approach to tutoring self-explanations discussed above is a form of static metacognitive support. In this intervention, the key metacognitive decision, namely, when/whether to self-explain one's problem-solving steps, is performed by the system not by the student, and consequently the system cannot assess students' metacognitive decisions. The system *does* assess the correctness of students' self-explanations, but once the decision to self-explain is made, the self-explanations themselves may well be viewed as domain-level behavior. Nevertheless, this tutor should still be viewed as supporting metacognition. Because it scaffolds students in engaging in positive metacognitive behavior, there is an opportunity for students to internalize this behavior and, perhaps more importantly, benefit from it at the domain level.

By contrast, the other three systems are adaptive to students' metacognitive behaviors. All three of them assess aspects of students' metacognitive behavior and respond according to their respective metacognitive assessment of the student, although they differ substantially in the range and type of metacognitive behaviors to which they respond. The Error Correction tutor recognizes a single desirable metacognitive behavior or lack thereof, namely, whether students correct their own (domain-level) errors as soon as the negative consequences of those errors are apparent. Further, this error correction recognition was only implemented for a limited (though carefully chosen) subset of the domain skills, namely formula writing errors. Scooter distinguishes a single broad undesirable metacognitive category, namely, harmful gaming. This category encompasses a range of more specific behaviors, such as guessing answers or help abuse, but Scooter does not "diagnose" them separately. It assesses metacognition at a more

aggregate level than the other two systems. The Help Tutor, on the other hand, assesses students' help-seeking behavior (and also how deliberately students work with the Cognitive Tutor) in a highly fine-grained manner. It recognizes several different specific metacognitive errors, unlike the other three systems.

The systems differ further with respect to the content of the feedback they provide on students' metacognitive behaviors. In any given system, the content of the feedback may relate primarily to the domain level, it may be primarily metacognitive, or it may involve both. As a further distinction, the feedback may comprise correctness information only, meaning that it indicates only whether the student engages in metacognitive behaviors deemed to be productive or not (without explaining why), or it may comprise more elaborate messages, for example, messages relating the student's behavior to specific principles of metacognition (e.g., help-seeking principles) or problem-solving principles at the domain level.

The intervention designed to tutor self-explanations provides feedback on the correctness of students' self-explanations. As mentioned, we consider this to be domain-level feedback. The intervention designed to tutor error correction reacts to metacognitive errors (namely, students' not fixing a domain-level error when the consequence of that error is apparent) mainly in a domain-related way. It helps the student to generate correct behavior through a domain-specific discussion of what went wrong. It does not give students meta-cognitive level instruction, for instance, by suggesting a general strategy for checking for errors like attempting the problem again with a different strategy. Likewise, Scooter's main reaction to gaming the system is to assign remedial problems that address domain-level topics that the student is struggling with, according to its student model. These remedial problems constitute a form of domain-related feedback. Scooter also provides correctness feedback at the metacognitive level, both through

brief messages and the animated animal cartoon character's emotional expressions. Of the four systems, the Help Tutor is the only one whose feedback is entirely metacognitive in content. It provides specific metacognitive error feedback messages, relating students' behavior to desired help-seeking principles.

Thus, the four interventions exhibit significant variability in their design. They provide case studies exploring different points in a design space for metacognitive support, defined by the dimensions discussed above, such as (a) whether the support for metacognition is static or adaptive, (b) the range and type of metacognitive behaviors that they respond to, (c) the level of detail at which they analyze students' metacognitive behaviors, (d) whether the feedback they provide in response to students' metacognitive behaviors is domain-related or metacognitive in nature, and (e) whether it is correctness feedback only or whether it is elaborate and explanatory. We expect that further dimensions in this design space for metacognitive support will be identified as researchers continue to develop and evaluate methods for automated tutoring of metacognition.

Although we cannot be said to have systematically explored the space defined by these dimensions, and it is therefore not possible to draw definitive conclusions about the effectiveness of any individual dimension or design feature, it is nonetheless interesting to ask to what hypotheses we can "extract" about the kinds of design features that may be effective in metacognitive tutors. We emphasize, however that the purpose of the discussion is to generate hypotheses, rather than to formulate definitive lessons learned.

The reader may recall the four levels of goals with respect to metacognitive tutoring, namely, to improve students' metacognitive behaviors, and their domain-level learning, both *during* the intervention and *afterwards*. Table 1 summarizes the results of our experiments with respect to

the four goals. Three of the four studies show improved domain learning as a result of metacognitive support; two yield evidence that such support can improve metacognitive behavior during the intervention (see Table 1).

Table 1: Relating the four studies to four goals of metacognitive support given in Figure 1.

	During Intervention		After Intervention	
	Improved Metacognitive Behavior	Improved Domain Learning	Improved Metacognitive Behavior	Improved Domain Learning
Static	Cannot tell without online assessment of metacognitive behavior	Tutoring self-explanations leads to improvement	Did not assess	Did not assess
Adaptive	Scooter and the Help Tutor led to improved metacognitive behavior	Tutoring error correction and Scooter led to improvement; Help Tutor did not	Help Tutor did not improve metacognitive behavior in a transfer environment	Did not assess, but unlikely given lack of metacognitive improvement

One might wonder whether metacognitive support that adapts to students' metacognitive behavior (i.e., what we have termed adaptive support), is more effective than static metacognitive support. Given that in our experience, adaptive metacognitive support is much more difficult to implement in computer tutors than static support, one might wonder whether it provides a good return on investment. The pattern of results obtained does not provide support for the hypothesis that adaptive support leads to better domain level learning than static support. The Self-Explanation Tutor, a form of static metacognitive support, improved students' domain-level learning, whereas the Help Tutor, with its detailed assessment of students' metacognition and highly specific metacognitive feedback, did not. However, we reiterate that in our experiments the systems differ in multiple ways, not just whether the support is static or

adaptive, and that therefore it is hard to draw a firm conclusion. An ideal experiment would contrast this dimension, static versus adaptive support, for a single metacognitive skill, like self-explanation, in the same domain(s), like geometry.

We note further that our experiments do not at all rule out that adaptive metacognitive support may be better for long-term retention of students' metacognitive abilities (the third of the four goals in Figure 1). Our experiments are largely silent on this issue. We tested long-term improvement only with respect to help seeking, and did not find a significant improvement in students' help-seeking *behavior* (although we did find an improvement in their declarative knowledge of help seeking). It may well be, then, that across a broader range of metacognitive skills, adaptive metacognitive support will, upon further research, turn out to be desirable to better help students exhibit metacognitive behavior in the long-term.

The pattern of results depicted in Table 1 seems to suggest another surprising trend: in the four systems we tested, the narrower the scope of metacognitive behaviors that the tutor targets (either through static or adaptive support), the stronger the effect on domain learning. Again, we note that our experiments were not designed to test this pattern in a rigorous manner; the systems differ along many dimensions in addition to this one. We found that the Self Explanation and Error Correction tutors, which each target a single metacognitive skill, were most clearly effective at the domain level. The Help Tutor on the other hand, which targets a wide range of metacognitive skills, with many detailed error categories corresponding to a comprehensive set of principles of help seeking, essentially did not have an effect on domain learning. Scooter falls somewhere in between these two "extremes," both in terms of the range of the metacognitive behaviors that it targets and its beneficial effect on domain-level learning. It may seem counterintuitive that targeting a single significant metacognitive skill might be more beneficial

(to domain learning) than providing feedback on a wide range of metacognitive behaviors.

Perhaps students have difficulty telling the forest from the trees when receiving a wide range of metacognitive feedback. Alternatively, systems that target a broad set of metacognitive skills may violate a key principle of both cognitive and metacognitive tutoring, that is, to communicate the goal structure underlying the problem solving (Roll, Alevan, McLaren & Koedinger, 2007b).

The patterns of results shown in Table 1 further suggests that metacognitive support may be more effective when the students perceive the metacognitive steps that they are asked to perform as being a natural part of the task at hand and not as extra work. In the Self-Explanation and Error Correction tutors (which as mentioned were most clearly successful at the domain level), the targeted metacognitive behaviors were essential to successful completion of the task presented to the student. Students could not finish the Geometry problems assigned to them without successfully explaining their steps, and could not complete the Excel training without successfully correcting their errors. This tight integration with the task makes students engage in the target metacognitive skill as a matter of course, and was shown to achieve the most improvement in learning.

This tight integration was not the case for Scooter and the Help Tutor, which were less successful at improving domain level learning. Unlike the previous two examples, in the cases of Scooter and the Help Tutor, students' metacognitive errors did not prevent them from completing the task at hand. On the contrary, one might even argue that the given tasks could be solved more easily by gaming the system or abusing its help facilities – that is, desired metacognitive behavior may be viewed as being at odds with fast completion of the tasks as given (even if it is hypothesized to improve long-term learning). In particular, some students complained that Scooter prevented them from completing the problems.

Thus, with the caveat that there were multiple differences between the systems that might help explain the pattern of results in Table 1, in the systems that were more successful at the domain level, the metacognitive behaviors appeared to be a regular (natural?) part of task performance and so may have been perceived by students as being *beneficial* for the problem-solving process.

The Role of Affect in Metacognitive Choices

Another possible explanation for the differences in domain learning observed between studies is the extent to which the meta-cognitive process has connections with students' affect. It may be that the more connected the metacognitive process is to affective issues (e.g., gaming the system, and the gaming-related errors in help-seeking may be more strongly related to affective issues than self-explanation or error correction), the less chance that an approach centered on scaffolding, monitoring, and tutoring will work. A source of suggestive evidence comes from the studies of gaming and affect that identified associations between negative affect and high levels of gaming (e.g., bored students are more likely to begin gaming). To confirm this hypothesis we will need corresponding evidence that self-explanation and error correction are not associated with negative affect. However, the association between gaming and affect already suggests that it may be valuable to either scaffold students' affect, in order to enable them to learn key metacognitive skills, or to assist students in developing affective self-regulation strategies that support better metacognition.

Tutoring and Assessing Metacognition During Learning

Beyond the direct effects on domain learning, the projects discussed in this chapter demonstrate the potential of intelligent tutoring systems to support the exercise of metacognitive skills. Through a rigorous process of design and evaluation, using existing knowledge on

domain-level tutoring, we created interventions that support students' metacognition within intelligent tutoring systems.

We have made some progress toward the goal of testing whether metacognitive support leads to measurable improvements in students' metacognition. The three adaptive systems presented in this paper each depended on the development of sophisticated models that automatically monitor and assess differences in students' metacognitive behaviors, as they used the tutoring software. The Help Tutor used a rational production-rule model, as did the Error Correction tutor (although it was much simpler in its metacognitive components). Scooter used a statistical model developed through machine learning. These models have been one of the key innovations underpinning our work, enabling us to study students' metacognition more effectively by applying the models to existing data (e.g., to study how gaming varies across tutor lessons) as well as enabling automated interventions. An important future step will be to use these detectors to assess student metacognitive behavior after the intervention has been removed and the experimenters have returned to the laboratory. By assessing students' future metacognition, we will be in a position to test the hypothesis that adaptive metacognitive support will lead to better learning – and to find out how durable any effects are.

While much work is still to be done, we would like to emphasize two main lessons learned with respect to evaluation and assessment of metacognition:

- a. Evaluation of interventions: Using the *in vivo* experimentation paradigm, we tested these interventions in ecologically valid classroom environments. *In vivo* experimentation is especially important in evaluating metacognitive instruction, which is tightly related to aspects of motivation, goal orientation, socio-cultural factors, etc. A metacognitive intervention that is effective in the laboratory may not be effective in

more ecologically valid settings. Furthermore, evaluating the systems in students' 'natural habitat' allows for collecting fine-grained data of students' metacognitive behaviors in the classroom.

- b. Automated assessment of metacognitive behavior: Each of the four systems used automated assessment of students' actions at the domain level. Furthermore, three of the systems used models of students' metacognitive behavior in order to detect metacognitive errors, independently from domain errors in the cases of the Help Tutor and Scooter. Each of these models/detectors has been shown to be able to transfer, to at least some degree, across domain contents and student cohorts. This automation creates the potential that formative metacognitive assessment can be embedded within a wide variety of learning environments; a step that, when combined with appropriate instruction, may have the potential to achieve wide-spread improvements to student metacognition, with lasting effects on learning.

To summarize, the projects described above demonstrate advancements and possibilities in using intelligent tutoring systems to support, teach, and assess metacognition. At the same time, they suggest important links between affect and metacognition.

References

- Aleven, V., & Koedinger, K. R. (2000a). 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.
- Aleven, V., & Koedinger, K. R. (2000b). The need for tutorial dialog to support self-explanation. In C. P. Rose & R. Freedman (Eds.), *Building Dialogue Systems for Tutorial Applications*,

Papers of the 2000 AAAI Fall Symposium (pp. 65-73). Technical Report FS-00-01. Menlo Park, CA: AAAI Press.

Aleven, 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.

Aleven, V., K. R. Koedinger, Sinclair, H. C., & Snyder, J. (1998). Combatting shallow learning in a tutor for geometry problem solving. In B. P. Goettl, H. M. Halff, C. L. Redfield, & V. J. Shute (Eds.), *Intelligent Tutoring Systems, Fourth International Conference, ITS '98* (pp. 364-373). Lecture Notes in Computer Science 1452. Berlin: Springer Verlag.

Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence and Education*, 16, 101-128.

Aleven, V., Ogan, A., Popescu, O., Torrey, C., & Koedinger, K. (2004). Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.), *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 443-454). Berlin: Springer Verlag.

Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R.M. (2003). Help seeking and help design in interactive learning environments. *Review of Educational Research*, 73(2), 277-320.

Anderson, J. R., & Lebière, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Arbreton, A. (1998). Student goal orientation and help-seeking strategy use. In S. A. Karabenick (Ed.), *Strategic Help Seeking: Implications For Learning And Teaching* (pp. 95-116). Mahwah, NJ: Lawrence Erlbaum Associates.

- Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Meheranian, H., Fisher, D., Barto, A., Mahadevan, S., Woolf, B.P. (2007) Repairing Disengagement with Non-Invasive Interventions. *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, 195-202.
- Atkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning from studying examples to solving problems: Combining fading with prompting fosters learning. *Journal of Educational Psychology*, 95, 774-783.
- Azevedo, R., & Cromley, J. G. (2004a). Does Training on Self-Regulated Learning Facilitate Students' Learning With Hypermedia? *Journal of Educational Psychology*, 96(3), 523-35.
- Azevedo, R., Cromley, J. G., & Seibert, D. (2004b). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia? *Contemporary Educational Psychology*, 29, 344-70.
- Azevedo, R., Moos, D., Greene, J., Winters, F., & Cromley, J. (2008). Why is externally-facilitated regulated learning more effective than self-regulated learning with hypermedia? *Educational Technology Research and Development*, 56(1), 45-72.
- Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., Koedinger, K. (2008) Why Students Engage in "Gaming the System" Behavior in Interactive Learning Environments. *Journal of Interactive Learning Research*, 19 (2), 185-224.
- Baker, R.S.J.d. (2007) Is Gaming the System State-or-Trait? Educational Data Mining Through the Multi-Contextual Application of a Validated Behavioral Model. *Complete On-Line Proceedings of the Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling*, 76-80.

- Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Evenson, S.E., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., Beck, J. (2006) Adapting to When Students Game an Intelligent Tutoring System. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 392-401.
- Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". *Proceedings of ACM CHI 2004: Computer-Human Interaction*, 383-390.
- Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R. (2008) Developing a Generalizable Detector of When Students Game the System. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 18, 3, 287-314..
- Beck, J., Chang, K-m., Mostow, J., Corbett, A. (2008) Does Help Help? Introducing the Bayesian Evaluation and Assessment Methodology. *Proceedings of Intelligent Tutoring Systems 2008*, 383-394.
- Biswas, G., Leelawong, K., Schwartz, D., Vye, N. & The Teachable Agents Group at Vanderbilt (2005). Learning By Teaching: A New Agent Paradigm for Educational Software, Applied Artificial Intelligence, vol. 19, (pp. 363-392).
- Bjork, R.A. (1994). Memory and metamemory considerations in the training of human beings. In J. Metcalfe and A. Shimamura (Eds.), *Metacognition: Knowing about knowing*. (pp.185-205). Cambridge, MA: MIT Press.
- Bransford, J.D., Brown, A.L., Cocking, R.R. (2000) *How People Learn: Brain, Mind, Experience, and School*. Washington, DC: National Research Council.
- Brown, A. L., Bransford, J. D., Ferrara, R. A., & Campione, J. C. (1983). Learning, remembering, and understanding. In *Handbook of child psychology* (pp. 77-166). New York: Wiley.

- Bruer, J. (1993). The Mind's Journey from Novice to Expert. *American Educator*, 6-46.
- Butler, D. L. (1998). The strategic content learning approach to promoting self-regulated learning: a report of three studies. *Journal of Educational Psychology*, 90(4), 682-97.
- Byrnes, J. P., & Wasik B. A. (1991). Role of conceptual knowledge in mathematical procedural learning. *Developmental Psychology*, 27, 777-786.
- Cheng, R., Vassileva, J. (2005) Adaptive Reward Mechanism for Sustainable Online Learning Community. *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 152-159.
- 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.
- Chi, M. T. H., P. J. Feltovich, & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5, 121-152.
- 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.
- Conati C., & VanLehn, K. (2001). Providing adaptive support to the understanding of instructional material. In C. Sidner & J. Moore (Eds.), *Proceedings of IUI 2001, the Sixth International Conference on Intelligent User Interfaces* (pp. 41-47). New York: ACM Press.

- Conati, C., Muldner, K., & Carenini, G. (2006). From example studying to problem solving via tailored computer-based meta-cognitive scaffolding: hypotheses and design. *Technology, Instruction, Cognition, and Learning*, 4(2), 139-190.
- Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4, 253–278.
- Corbett, A. T., Anderson, J. R. (2001) Locus of feedback control in computer-based tutoring: impact on learning rate, achievement and attitudes. Proceedings of CHI 2002, Human Factors in Computing Systems (March 31 - April 5, 2001, Seattle, WA, USA), ACM, 2001 245-252
- Corbett, A.T., McLaughlin, M.S., Scarpinato, K.C. & Hadley, W.S. (2000). Analyzing and Generating Mathematical Models: An Algebra II Cognitive Tutor Design Study. *Intelligent tutoring systems: Proceedings of the Fifth International Conference, ITS'2000* (pp. 314-32)3.
- Corbett, A.T., Wagner, A., Lesgold, S., Ulrich, H. & Stevens, S. (2006). The impact on learning of generating vs. selecting descriptions in analyzing algebra example solutions. *Proceedings of the 7th International Conference of the Learning Sciences* (pp. 99-105).
- Corbett, A. T., Wagner, A., & Raspat, J. (2003). The Impact of analyzing example solutions on problem solving in a pre-algebra tutor. *Proceedings of AIED 2003: The 11th International Conference on Artificial Intelligence and Education*, (pp. 133-140).
- D'Mello, S. K., Picard, R. W., and Graesser, A. C. (2007) Towards an Affect-Sensitive AutoTutor. Special issue on Intelligent Educational Systems – *IEEE Intelligent Systems*, 22 (4), 53-61.
- Flavell J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34(10), 906-911

- Fuson, K. C. (1990). Conceptual structures for multiunit numbers: Implications for learning and teaching multidigit addition, subtraction, and place value. *Cognition and Instruction*, 7, 343-403.
- Grosse & Renkl (in press). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and Instruction*.
- Hausmann, R. G. M., & Chi, M. T. H. (2002). Can a computer interface support self-explaining? *International Journal of Cognitive Technology*, 7(1), 4-14.
- Hiebert, J., & Wearne, D. (1996). Instruction, understanding, and skill in multidigit addition and subtraction. *Cognition and Instruction*, 14, 251-283.
- Karabenick, S. & Newman, R. (2006). *Help Seeking in Academic Settings: Goals, Groups, and Contexts*. Mahwah, NJ: Erlbaum.
- Koedinger, K. R. & Corbett, A. T. (2006). Cognitive Tutors: Technology bringing learning science to the classroom. In K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences*. Cambridge University Press.
- Koedinger, K. R. & Nathan, M. J. (2004). The real story behind story problems: Effects of representations on quantitative reasoning. *The Journal of the Learning Sciences*, 13 (2), 129-164.
- Koedinger, K. R., & Alevan V. (2007). Exploring the assistance dilemma in experiments with Cognitive Tutors. *Educational Psychology Review*, 19(3), 239-264.
- Kort, B., Reilly, R., & Picard, R.D. (2001). An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy - building a Learning Companion. *Proceedings of IEEE Intl. Conf. on Advanced Learning Technologies*, 43-46.

- Lovett, M. C. (1998). Cognitive task analysis in service of intelligent tutoring systems design: A case study in statistics. In B. P. Goettl, H. M. Half, C. L. Redfield, & V. J. Shute (Eds.) *Intelligent Tutoring Systems*, Lecture Notes in Computer Science Volume 1452 (pp. 234-243). New York: Springer.
- Magnussen, R., Misfeldt, M. (2004) Player Transformation of Educational Multiplayer Games. *Proceedings of Other Players*.
- Martin, B., Koedinger, K.R., Mitrovic, A., and Mathan, S. (2005) On Using Learning Curves to Evaluate ITS. *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 419-428.
- Martínez Mirón, E.A., du Boulay, B., Luckin, R. (2004) Goal Achievement Orientation in the Design of an ILE. *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, 72-78.
- Mathan, S. A. & Koedinger, K. R. (2005) Fostering the Intelligent Novice: Learning from errors with metacognitive tutoring. *Educational Psychologist*. 40(4), 257-265.
- Mevarech, Z. R., & Kramarski, B. (2003). The effects of metacognitive training versus worked-out examples on students' mathematical reasoning. *British Journal of Educational Psychology*, 73, 449-71.
- Murray, R.C., VanLehn, K. (2005) Effects of Dissuading Unnecessary Help Requests While Providing Proactive Help. *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 887-889.
- Nathan, M. J., Kintsch, W., & Young, E. (1992). A theory of algebra word problem comprehension and its implications for the design of computer learning environments. *Cognition and Instruction*, 9(4), 329-389.

- Newman, R. S. (1994). Academic 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. 293-301). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Newman, R. S. (2002). How self-regulated learners cope with academic difficulty: the role of adaptive help seeking. *Theory into Practice*, 41(2), 132-8.
- Nicaud, J. F., Bouhineau, D., Mezerette, S., Andre, N. (2007). *Aplusix II* [Computer software].
- Paas, F. G. W. C., & Van Merriënboer, J. J. G. (1994). Variability of worked examples and transfer of geometrical problem solving skills: A cognitive-load approach. *Journal of Educational Psychology*, 86, 122-133.
- Palincsar, A.S., & Brown, A.L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, 1, 117-175.
- Pintrich, P. R. (1999). The role of motivation in promoting and sustaining self-regulated learning. *International Journal of Educational Research*, (31), 459-70.
- Pintrich, P. R. (2004). A conceptual framework for assessing motivation and self-regulated learning in college students. *Educational Psychology Review* 16(4), 385-407.
- Pintrich, P. R., & De Groot, E. V. (1990). Motivational and Self-Regulated Learning Components of Classroom Academic Performance. *Journal of Educational Psychology*, 82(1), 33-40.
- Ploetzner, R., VanLehn, K. (1997) The Acquisition of Qualitative Physics Knowledge During Textbook-Based Physics Training. *Cognition and Instruction*, 15 (2), 169-205.
- 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.
- Rodrigo, M.M.T., Baker, R.S.J.d., d'Mello, S., Gonzalez, M.C.T., Lagud, M.C.V., Lim, S.A.L., Macapanpan, A.F., Pascua, S.A.M.S., Santillano, J.Q., Sugay, J.O., Tep, S., Viehland, N.J.B. (2008) Comparing Learners' Affect While Using an Intelligent Tutoring Systems and a

Simulation Problem Solving Game. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, 40-49.

Rodrigo, M.M.T., Baker, R.S.J.d., Lagud, M.C.V., Lim, S.A.L., Macapanpan, A.F., Pascua, S.A.M.S., Santillano, J.Q., Sevilla, L.R.S., Sugay, J.O., Tep, S., Viehland, N.J.B. (2007) Affect and Usage Choices in Simulation Problem Solving Environments. *Proceedings of Artificial Intelligence in Education 2007*, 145-152.

Roll, I., Alevan, V., McLaren, B. M., & Koedinger, K. R. (2007a). Can help seeking be tutored? Searching for the secret sauce of metacognitive tutoring. In R. Luckin, K. Koedinger, & J. Greer (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education, AIED 2007* (pp. 203–210). Amsterdam, the Netherlands: IOS Press.

Roll, I., Alevan, V., McLaren, B. M., & Koedinger, K. R. (2007b). Designing for metacognition - applying cognitive tutor principles to the tutoring of help seeking. *Metacognition and Learning*, 2(2), 125-40.

Roll, I., Baker, R.S., Alevan, V., McLaren, B., & Koedinger, K. (2005). Modeling students' metacognitive errors in two intelligent tutoring systems. In L. Ardissono, P. Brna and A. Mitrovic (Eds.), *Proceedings of the 10th International Conference on User Modeling, UM 2005* (pp. 379-388). Berlin: Springer-Verlag.

Rosenshine, Barak, Meister, Carla. (1994). Reciprocal teaching: A review of the research. *Review of Educational Research*, 64(4), 479. Retrieved July 21, 2008, from Research Library Core database. (Document ID: 4466263).

Scardamalia, M. & Bereiter, C. (1985). Development of dialectical processes in composition. In D.R. Olson, N. Torrance, & A. Hildyard (Eds.) *Literacy, language, and learning: The nature and consequences of reading and writing*. Cambridge: Cambridge University Press.

- Schmidt, R.A. & Bjork, R.A. (1992). New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training. *Psychological Science*, 3 (4), 207-217
- Schoenfeld, A. (1983). Beyond the purely cognitive: Belief systems, social cognitions, and metacognitions as driving forces in intellectual performance. *Cognitive Science*, 7, 329-363.
- Schoenfeld, A. H. (1987). What's all the fuss about metacognition? In *Cognitive science and mathematics education* (pp. 189-215). Hillsdale, N.J: Erlbaum.
- Schworm, S., & Renkl, A. (2002). Learning by solved example problems: Instructional explanations reduce self-explanation activity. *The 24th Annual Conference of the Cognitive Science Society*, , 816-21.
- Shih, B., Koedinger, K.R., and Scheines, R. (2008). A response time model for bottom-out hints as worked examples. In Baker, R.S.J.d., Barnes, T., & Beck, J.E. (Eds.) *Proceedings of the First International Conference on Educational Data Mining*, 117-126.
- Siegler, R. S. (2002). Microgenetic studies of self-explanation. In N. Garnott & J. Parziale (Eds.), *Microdevelopment: A process-oriented perspective for studying development and learning* (pp. 31–58). Cambridge, MA: Cambridge University Press.
- Sierra Online. (2001). *The Incredible Machine: Even More Contraptions* [Computer Software].
- Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*, 12, 257-85.
- Walonoski, J.A., Heffernan, N.T. (2006) Prevention of Off-Task Gaming Behavior in Intelligent Tutoring Systems. In M. Ikeda, K. D. Ashley, & T. W. Chan (Eds.), *Proceedings of 8th International Conference on Intelligent Tutoring System, ITS 2006*,(pp. 722-724) Berlin: Springer Verlag.
- Weerasinghe, A., & Mitrovic, A. (2004). Supporting self-explanation in an open-ended domain. In *Knowledge-Based Intelligent Information and Engineering Systems* (pp. 306-313). Berlin: Springer Verlag.

White, B. Y. & Frederiksen, J. R. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction* 16 (1): 3-118.

Wood, H. A., & Wood, D. J. (1999). Help seeking, learning, and contingent tutoring. *Computers and Education*, 33(2), 153-69.