Fostering the Intelligent Novice: Learning From Errors With Metacognitive Tutoring

Santosh A. Mathan

Human Centered Systems
Honeywell Labs
Minneapolis, MN

Kenneth R. Koedinger

Human Computer Interaction Institute
Carnegie Mellon University

This article explores 2 important aspects of metacognition: (a) how students monitor their ongoing performance to detect and correct errors and (b) how students reflect on those errors to learn from them. Although many instructional theories have advocated providing students with immediate feedback on errors, some researchers have argued that immediate feedback eliminates the opportunity for students to practice monitoring for and learning from errors. Thus, they advocate delayed feedback. This article provides evidence that this line of reason is flawed and suggests that rather than focusing on the timing of feedback, instructional designers should focus on the "model of desired performance" with respect to which feedback is provided. Instead of delaying feedback, we suggest broadening our model of correct behavior or desired performance to include some kinds of incorrect, but reasonable behaviors. This article explores the effects of providing feedback on the basis of a so-called *intelligent novice* cognitive model. A system based on an intelligent novice model allows students to make certain reasonable errors, and provides guidance through the exercise of error detection and correction skills. There are two pedagogical motivations for feedback based on an intelligent novice model. First, jobs today not only require ready-made expertise for dealing with known problems, but also intelligence to address novel situation where nominal experts are thrown back to the state of a novice. Second, the opportunity to reason about the causes and consequences of errors may allow students to form a better model of the behavior of domain operators. Results show that students receiving intelligent novice feedback acquire a deeper conceptual understanding of domain principles and demonstrate better transfer and retention of skills over time.

The question of when to intervene with feedback following errors is a controversial issue with important metacognitive implications. Some researchers have argued that immediate feedback following errors helps minimize unproductive floundering and associated frustration. Consequently, students remain motivated and efficient during problem solving. Others have argued that immediate feedback following errors may impede metacognitive activities. For instance, activities associated with self-regulation may be circumvented by feedback. Students may come to rely on the presence or absence of feedback messages to assess their performance. They may use error messages to guide error-correction steps. Proponents of delayed feedback argue that delay provides students with opportunities to engage in activities associated with self-regulation. However, inappropriate levels of delay can introduce inefficiencies in the instructional process and overwhelm students who may need guidance. Unfortunately, the polarized nature of the feedback debate leaves developers of instructional systems with a mutually exclusive choice between the distinct benefits of two feedback modalities.

This article presents a theoretical perspective that offers the opportunity for developers of instructional systems to jointly realize the benefits of immediate and delayed feedback. This article also summarizes evidence supporting this perspective. We situate our analysis in the context of

Correspondence should be addressed to Santosh A. Mathan, 3660 Technology Dr., Human Centered Systems, Honeywell Labs, Minneapolis, MN 55418. E-mail: santosh.mathan@honeywell.com

computer-based learning environments called Cognitive Tutors.

COGNITIVE TUTORS

We discuss the metacognitive implications of feedback timing in the context of Cognitive Tutors. Cognitive Tutors are computer-based instructional systems that use cognitive models—fine-grained representations of successful and unsuccessful strategies employed by learners—to track students through complex problem spaces (Anderson, 1993). Using a process called model tracing, these systems can assess student performance on an action-by-action basis. Such a diagnostic capability offers the opportunity to provide context-specific feedback and advice to learners. Whereas there is considerable flexibility in terms of how feedback is implemented in these systems, Cognitive Tutors have generally provided feedback immediately after student errors.

Cognitive Tutors have been rigorously assessed in classroom and laboratory contexts. They have been shown to reduce training time by half and increase learning outcomes by a standard deviation or more (Anderson, Corbett, Koedinger, & Pelletier, 1995). These systems have been used to teach concepts ranging from programming to genetics. Tutors for algebra and geometry are in use by 200,000 students in over 1,800 schools around the country. The U.S. Department of Education has designated Cognitive Tutors one of five exemplary curricula in K–12 mathematics education. However, despite these successes in the laboratory and real world, the strategy of providing immediate feedback on errors has been controversial. We elaborate on the controversy surrounding this issue in the following pages.

REVIEW OF FEEDBACK RELATED RESEARCH

Immediate Feedback

Inherent in any design decision concerning feedback is the issue of when it might be appropriate to provide feedback. Designers are faced with a choice of presenting feedback as soon as an error is detected—immediate feedback, or presenting delayed feedback—giving learners an opportunity to detect and correct errors on their own. Research on the subject does not offer a clear-cut answer to inform such decisions. A review of the literature, summarized here, shows distinct pedagogical advantages offered by each of these feedback modalities.

Several studies indicate that learning is more effective and efficient with immediate feedback. For instance, Lewis and Anderson (1985) explored the issue of feedback latency in the context of a maze-based adventure game. Each

room in the maze had a set of features (such as lamp, fireplace, doorkeeper, etc.). Players had a set of operators (e.g., Bow, knock, light fire) that would, in the presence of certain features, move them toward the ultimate goal of finding treasure. Participants were trained to play the game in either an immediate or delayed feedback condition. In the immediate feedback condition, participants were notified any time they applied operators that would lead them toward dead ends. In the delayed feedback condition, participants were allowed to pursue dead-end paths up to a depth of one room before being informed of the inappropriateness of a previous choice. Participants then had to use a backup operator to back out of the dead-end path. In a posttest, participants trained in the immediate feedback condition were more accurate at specifying correct operators when presented with descriptions of room features. The only case in which delayed feedback participants were more accurate was in the case of rooms with features indicative of dead ends—these participants were more familiar with the use of the backup operator.

Though the Lewis and Anderson study shows a distinct benefit for immediate feedback, certain characteristics of their task limit generalization of these findings to other problem solving domains. As Anderson, Conrad, and Corbett (1989) have suggested, the maze task was a situation where the total correct solution was never laid out before participants and they had to integrate in memory a sequence of moves. In contrast, in many problem-solving domains, particularly in many academic tasks, the final solution, along with intermediate steps, is available for learners to study. It is possible for students to assess their problem-solving steps after accomplishing relevant goals.

Corbett and Anderson (2001) compared the pedagogical benefits of immediate and delayed feedback in the context of their LISP tutor—a context more representative of academic tasks. Students in their study worked with four different versions of the tutor. In the Immediate Feedback condition, students were presented with a feedback message as soon as an error was made. In the Flag Feedback condition students were informed that an error was made without any explanatory text concerning the nature of the problem or any other interruption to the task. The Demand Feedback condition allowed learners to ask the tutor to check for errors of their own initiative. In the No Feedback condition students received no guidance in the course of problem solving but were told whether their solution was correct at the end of the problem. Performance on a paper and pencil posttest showed that all feedback conditions were better than no feedback on the posttest. There were no statistical differences among feedback conditions in the posttest scores. However, there was a large difference among the feedback conditions in terms of the learning rate. Students in the immediate feedback condition completed training significantly faster than those in the Demand and No Feed-

back conditions. Immediate feedback served to minimize floundering and keep the leaning process efficient.

Additional support for immediate feedback comes from research carried out in other instructional contexts. For instance, Bangert-Drowns, Kulik, Kulik, and Morgan (1991) conducted a meta analysis of 53 studies that compared immediate feedback to delayed feedback in test-like events. These studies examined applications ranging from classroom quizzes to computer-based learning environments. Bangert-Drowns and colleagues concluded that immediate feedback produces better learning outcomes than delayed feedback.

Delayed Feedback

Though the studies just described show an advantage for immediate feedback, many have been critical about the use of immediate feedback in the context of instructional systems. Researchers have argued that immediate feedback offered by Cognitive Tutors is qualitatively different from that offered by human tutors. For instance, research by Merrill and colleagues comparing human and computer tutors shows that human tutors do not intervene immediately on errors that may provide learning opportunities (Merrill, Reiser, Ranney, & Trafton, 1992). Other researchers have observed that human tutors let learners engage in activities associated with error detection and correction (Fox 1993). Furthermore, research on human tutoring strategies shows that tutors try to instill a sense of control in learners (Lepper & Chabay, 1988). Immediate feedback during problem solving, it could be argued, may detract from individual control over the problem-solving process. These comparisons to human tutors are important as skilled human tutors typically produce better learning outcomes than Cognitive Tutors—differences in tutoring strategies could be among some of the factors that underlie this difference.

Critics of immediate feedback have also argued that immediate feedback may foster unsound metacognitive beliefs among students. For instance, as Nathan (1998) has pointed out, immediate feedback may reinforce the belief prevalent among many students that problem solving is an immediate and single-step process rather than the deliberate and reflective process described by educational researchers. Furthermore, Bjork (1994) has highlighted the possibility that rapid progress through a task as afforded by immediate feedback may lead users to adopt an overly optimistic assessment of their level of comprehension and competence. As Nathan has noted, these possibilities closely correspond to epistemological beliefs identified by (Schommer, 1993) as being negatively correlated with academic achievement—that is, the degree to which students believe that learning requires minimal effort, that knowledge is acquired quickly, and in the certainty of knowledge that is learned.

Immediate feedback has also been criticized on the basis of empirical studies that highlight benefits of delayed feedback. These studies suggest that the benefits of delayed feedback may only become apparent over time, or in the context of trans-

fer tasks. For instance, in a study involving a genetics tutor, students received feedback as soon as an error was detected in the immediate feedback condition or at the end of a problem in the delayed feedback condition (Lee, 1992). Students in the immediate feedback completed training problems significantly faster. In a posttest the following day, students in both conditions performed equally well on problems encountered during training. However, students in the delayed feedback condition performed significantly better on a far transfer task.

Similar observations have been made in the area of motor learning. For instance, Schmidt and Bjork (1992) reported on a pattern of results showing that interventions that enhance performance during training may compromise retention and transfer. In one study, participants were asked to perform a complex arm movement within a specified interval. Feedback on movement accuracy was provided at the end of 1, 5, or 15 trials. Participants who were provided feedback after every trial made the fewest errors during training—followed by participants who received feedback in 5 trial blocks and 15 trial blocks, respectively. A retention test administered 10 min after training showed no difference in performance among the three groups. A retention test 2 days later showed a reversal in performance. Participants who received feedback in 15 trial blocks made the least errors—they were followed by participants who received feedback after five trials and every trial, respectively.

In a study involving the LISP tutor, students (Schooler & Anderson, 1990) had to create LISP expressions containing one or two extractor and combiner functions. Students were trained in either an immediate feedback condition—where the feedback was provided as soon as an error was detected, or in a delayed feedback condition—where feedback was presented after expression was complete and students had hit Return to evaluate the same. In a posttest administered the following day, delayed feedback condition participants finished faster and made half as many errors as those trained in the immediate feedback condition

Considered together these studies suggest a potential trade-off between the benefits offered by immediate and delayed feedback. On the basis of the pattern of findings just summarized, some researchers (e.g., Bjork, 1994; Nathan, 1998) have argued that immediate feedback promotes efficiency during training, whereas delayed feedback might lead to better retention and transfer performance. The guidance hypothesis offered by Schmidt, Young, Swinnen, and Shapiro (1989) provides an explanation for this tradeoff. Feedback may serve to guide student performance during training. Students can get through problems by implementing prescriptions embodied in feedback messages. This may have the effect of boosting performance during, and immediately following, training. However, immediate feedback can negatively impact learning in two ways. First, it could obscure important task cues—that is, learners may come to depend on feedback to assess progress on tasks instead of cues inherent in the natural task environment. Second, feedback could prevent important secondary

skills from being exercised. Implicit in the guidance hypothesis is the idea that immediate feedback may promote the development of *generative skills*. Generative skills are skill components that are involved in selecting and implementing problem solving operators in specific task contexts. However, *evaluative skills*—skills called for in evaluating the effect of applying problem solving operators, correcting errors, and monitoring one's own cognitive process—may go unpracticed. These evaluative functions are instead delegated to feedback. Consequently, task performance may be compromised in situations where students must jointly exercise evaluative and generative skills. Transfer tasks and retention tests are representative of situations where student performance is likely to be error prone and subject to floundering—where the joint exercise of generative and evaluative skills may be essential for success. Additionally, researchers have argued that the exercise of evaluative skills may provide opportunities for deeper conceptual understanding of domain principles. As Merrill, Reiser, Merrill, and Landes (1995) have theorized, errors provide an opportunity to develop a better model of the behavior of operators in a domain. Error recovery requires that students construct explanations about the causes and consequences of errors and act on their analyses. This kind of active self-explanation and problem solving could contribute to a better understanding of domain operators and their applicability in problem contexts.

The Instructional Designer's Dilemma

The research on feedback just summarized presents designers of instructional systems with a dilemma. Immediate feedback keeps the learning process efficient. Additionally, some of the most effective and broadly used Cognitive Tutors provide immediate feedback on errors (Corbett, Koedinger, & Hadley, 2001). However, a designer may also wish to realize benefits such as the development of debugging and metacognitive skills purportedly offered by delayed feed-

back. Unfortunately, the feedback-related literature offers little guidance as to what an appropriate level of delay might be in a given context. This could have serious consequences. At best, an inappropriate level of delay can introduce inefficiencies into the learning process. At worst, delayed feedback can recede to a no-feedback condition, leading to unproductive floundering and student frustration.

AN INTEGRATIVE PERSPECTIVE

This article argues that the key to jointly realizing the benefits of immediate and delayed feedback may lie in an emphasis on the *model of desired performance* underlying Cognitive Tutors. The model of desired performance refers to the behaviors or performance we desire students to achieve. In Cognitive Tutors the model of desired performance is implemented as a set of production rules representing target skills in a specific domain. The model of desired performance plays a diagnostic role in intelligent tutoring systems. When student behavior is consistent with the model of desired performance, the system does not intervene. However, if student behavior is inconsistent with the model of desired performance, the system intervenes with feedback so as to guide students toward performance that is consistent with the model.

Expert Model

Currently, feedback in Cognitive Tutors is based on what is broadly referred to as an *expert model*. An expert model of desired performance characterizes the end-goal of instruction as error free and efficient task execution. Feedback is structured so as to lead students toward expert-like performance. The tutor intervenes as soon as students deviate from a solution path. An expert model tutor focuses on the generative components of a skill. Figure 1 (left) illustrates the student interaction with an expert model tutor.

Expert model	Intelligent Novice Model
Student reads problem statement and identifies goals to be accomplished	Student reads problem statement and identifies goals to be accomplished
Student plans actions to accomplish goals	Student plans actions to accomplish goals
Student implements actions	Student implements actions
Student attends to feedback	Student attends to outcomes and looks for discrepancy between intended result and actual outcome
If correct: Student moves on	If a discrepancy is noted:

If wrong:
    Student is notified of error.
    Student has option to get
    instructions to fix problem
	Student identifies source of discrepancy

	Student attempts to repair original solution
	If discrepancy is missed, or repair attempt fails:
	Student guided through error detection and correction process
	Student tests solution

FIGURE 1   Interaction with an expert model (left) and intelligent novice tutor (right).

Intelligent Novice Model

An alternative model that could serve as the basis for feedback in Cognitive Tutors is that of an intelligent novice (cf. Bruer, 1993). The assumption underlying such a model of desired performance is that an intelligent novice, someone with general skills facing a novel problem, is still likely to make errors. Recognizing this possibility, the intelligent novice model incorporates error detection and error correction activities as part of the task. Feedback based on such a model would support the student in both the generative and evaluative aspects of a skill, while preventing unproductive floundering. Delayed feedback merely gives students the *opportunity* to exercise evaluative skills. In contrast, an intelligent novice model-based tutor goes a step further by explicitly modeling these skills and providing instructional scaffolds through the exercise of error detection and correction activities. Feedback regarding a model of an intelligent novice may resemble delayed feedback, but it is really immediate feedback regarding a model of desired performance that includes error detection and correction skills. Figure 1 (right) outlines student interaction with a tutor based on an intelligent novice model.

The analysis just presented recasts the feedback debate. The integrative perspective outlined here suggests that the model of desired performance, and not feedback timing, should be the crucial issue of focus in deciding when to intervene following an error. This article details the design of two versions of a spreadsheet tutor—one based on an expert model the other on an intelligent novice model. Results from two studies evaluating learning outcomes associated with each will also be presented.

SPREADSHEET TUTOR

Spreadsheets have been widely regarded as exemplary end-user programming environments (Nardi, 1993). They allow nonprogrammers to perform sophisticated computations without having to master a programming language. However, despite decades of evolution in spreadsheet design, there are aspects of spreadsheet use that are sources of difficulty for novice and expert spreadsheet users (e.g., Hendry & Green, 1994). A commonly reported usability problem concerns the appropriate use of absolute and relative references—these are schemes that allow users to perform iterative computations. Difficulties in cell referencing have persisted despite an abundance of manufacturer and third-party training materials and decades of spreadsheet refinement. The tutor reported in this article was designed to enable students to master cell-referencing concepts. We elaborate on the tutorial domain later and go on to detail features of two versions of a tutor based on the theoretical analysis presented earlier.

Overview of Tutorial Domain

Spreadsheet formulas may be reused to perform iterative operations. This is accomplished through a scheme called relative referencing. Consider the spreadsheet depicted in Figure 2 (left). One could enter a formula in cell B5 that adds the contents of cells B2, B3, and B4. The corresponding operation can be performed in cells C5 and D5 simply by copying the formula entered in cell B5 and pasting it into these new locations. When pasted, the formula is automatically modified to refer to cells that lie at the same relative location as the original formula. For example, the formula in Cell B5 referred to the three cells above it. When the formula is copied and pasted into cells C5 and D5 the formulas are modified to refer to the three cells above these new locations.

To determine the appropriate relative references at new locations, formulas are updated based on where the formula is moved. When a formula is moved into a cell in a different column, column references in the formula are updated by the number of columns moved (Figure 2 [left], =B2+B3+B4 be-
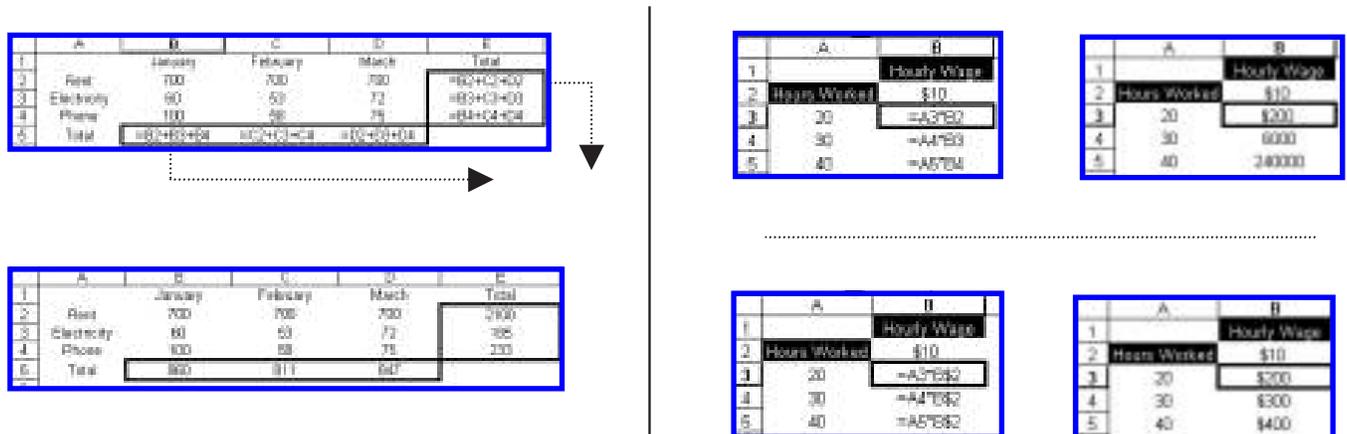

FIGURE 2   Relative references allow formulas in B5 and E2 to be reused (left). Incorrect use of relative refs (right top) remedied absolute refs (right bottom).

comes =D2+D3+D2 when moved across columns from B5 to D5). Similarly, when a formula is copied and pasted into a cell in a different row, all row references in the formula are updated by the number of rows moved (Figure 2 [left], =B2+C2+D2 becomes =B4+C4+D4 when moved across rows from E2 to E4).

Though relative referencing works in many task contexts, it is sometimes necessary to hold a row or column reference fixed regardless of where a formula is moved. Consider the example in Figure 2 (right). The value in cell B2 (Hourly Wage) has to be multiplied with the values in cells A3, A4, and A5 (Hours Worked). If the formula =A3*B2 is entered into B3 and pasted into cells B4 and B5, all row references will change to refer to cells that lie at the same relative location as those referred to by the formula in B3. This would produce =A4*B3 in B4 and =A5*B4 in B5 (instead of =A4*B2 and =A5*B2, respectively). For the formula to continue to refer to cell B2, the row reference 2 has to be held fixed as an absolute reference. This can be done by placing a "$" ahead of "2." Thus, for the formula in B3 to work appropriately when copied and pasted, it would be modified to read =A3*B$2.

Expert Model Spreadsheet Tutor

The theoretical analysis presented earlier informed the design of two versions of an intelligent tutoring system for teaching students cell referencing skills—one based on an expert model, the other on a model of an intelligent novice. During problem solving, students working with an expert model tutor receive feedback as soon as an incorrect formula is entered (Figure 3). The error notification message presents students with the choice of correcting the error on their own or getting help from the system in generating a solution. If help is sought, the student is interactively guided through the process of generating a solution. Students are first prompted to predict references that will change when the formula is copied and pasted. Subsequently, students are asked to identify references that must be prevented from changing.

Intelligent Novice Model Spreadsheet Tutor

In contrast, the intelligent novice tutor allows students to enter an incorrect formula, copy it, and paste it to observe the consequences of the error. The student is given an op-
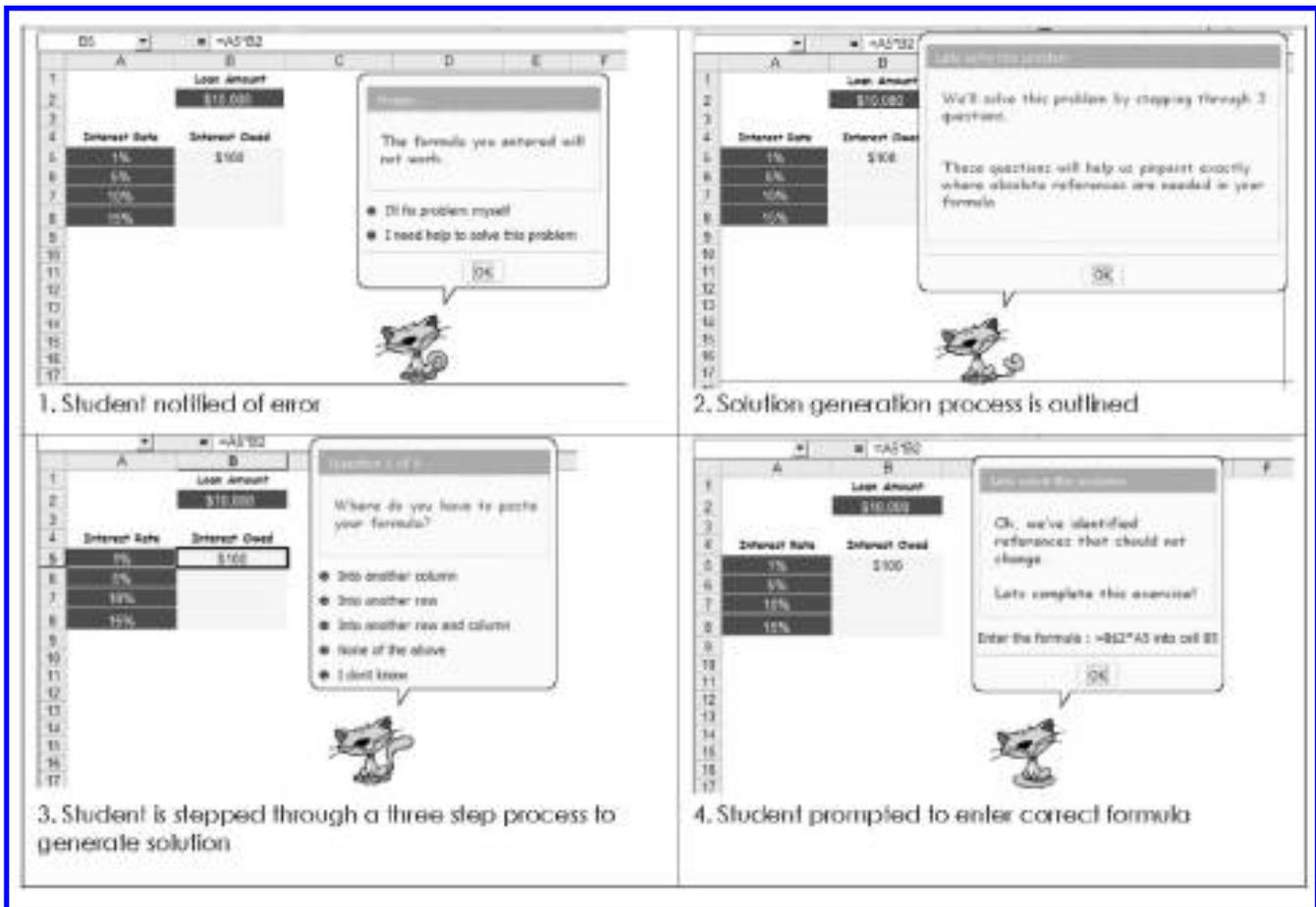

FIGURE 3   Feedback in the expert version of the tutor.

portunity to detect errors and make corrections if necessary. However, if a student fails to detect an error and tries to move on to a new problem, feedback directs the student to check for errors and request hints. An error at the formula correction step results in immediate corrective feedback to minimize unproductive floundering. Note, a delayed feedback tutor would not necessarily intervene at this point. The error notification message at the formula correction step presents students with the choice of correcting the error on their own or doing so with help from the system. If a student chooses to correct the error with help from the system, the student is taken through a two-step process to get back on track.

The first step (Figure 4—left) focuses on error detection. The system picks a cell into which a student may have pasted an incorrect formula. The system then highlights the cells inappropriately referenced by the underlying formula. Additionally, a tag indicating the incorrect formula underlying the selected cell is also shown. On the basis of these cues the student can determine the discrepancy between actual and intended outcomes and identify incorrect references. The second step (Figure 4, right) focuses on error correction. Having detected the source of error in the first step, the second step requires students to identify references that must be held fixed in order for the formula to work as intended. It is important to note that both versions of the spreadsheet tutor are implemented using model tracing and immediate feedback. The difference in feedback timing is a consequence of different models of desired performance embodied in each system.

EVIDENCE FOR FEEDBACK BASED ON AN INTELLIGENT NOVICE MODEL OF DESIRED PERFORMANCE

The two versions of the spreadsheet tutor were experimentally evaluated (Mathan & Koedinger, 2003). Participants were recruited from a local temporary employment agency. All participants had general computer experience, including proficiency with word processing, e-mail, and Web applications. However, they were all spreadsheet novices. We randomly assigned students to either the intelligent novice or expert condition.

Experimental sessions spanned the course of 3 days. On Day 1, students came in for a 90-min instructional session. Declarative instruction provided all students with an exposition of basic spreadsheet concepts: everything from data entry and copying and pasting to formula creation and cell referencing. The remainder of the session was spent on procedural practice. Students solved a variety of problems that called for the exercise of cell-referencing skills. The session was preceded by a pretest and was followed by a posttest. On Day 2, students came in the next day for 50 min of procedural practice with the tutor. A posttest was administered following the instructional session. On Day 3, 8 days after Day 2, students came in for a third instructional session. Students attempted a pretest and transfer task to measure retention prior to the instructional session. The third session consisted of 30 min of procedural practice and was followed by a posttest.

The pre- and posttests had two components: a test of problem solving and a test of conceptual understanding. The problem-solving test consisted of problems isomorphic to
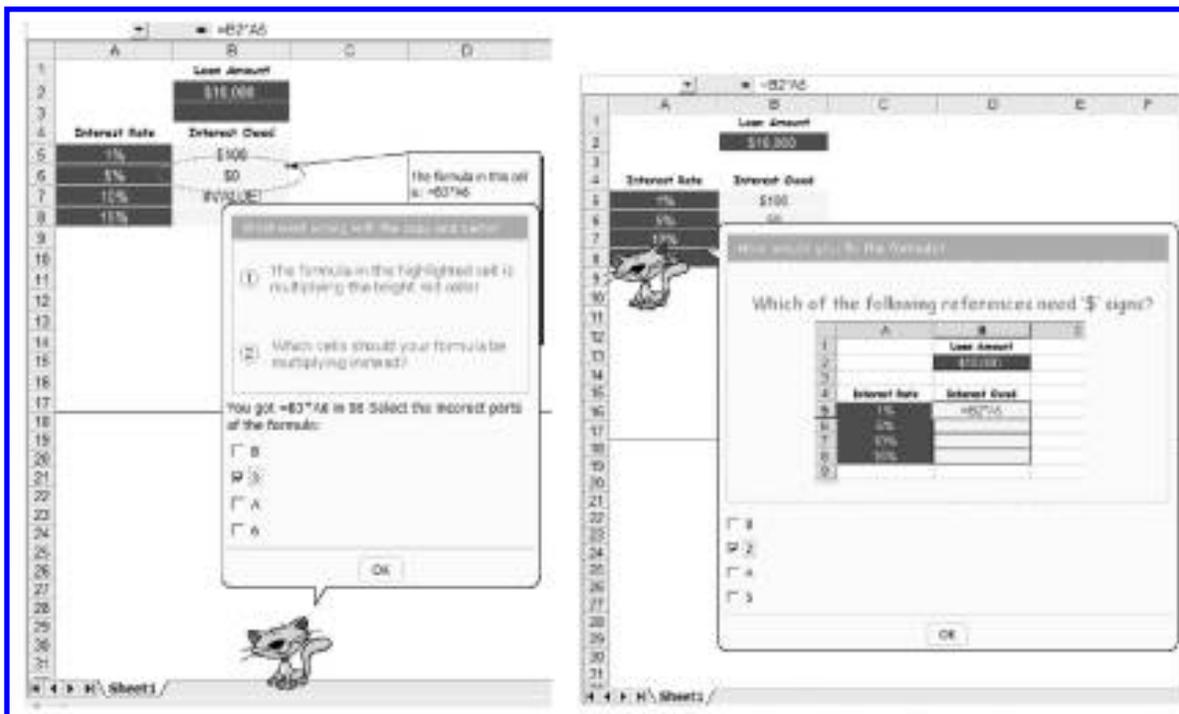

FIGURE 4   Error Detection (left) and Error Correction (right) steps in the Intelligent Novice Tutor.

training tasks; in other words, students were asked to write and use Excel formulas. The conceptual test consisted of two parts: The first part required students to exercise predictive skills. Students had to identify an outcome (from a selection of screenshots) that could have resulted from copying and pasting a given formula. The second called for students to exercise error-attribution skills. Students had to examine a given spreadsheet table and identify which of several formula alternatives could have produced the observed outcome. The transfer task called for the exercise of cell-referencing skills in the context of a structurally complex spreadsheet. All students were assessed for incoming mathematical ability using a test of basic algebraic symbolization.

Declarative instruction in the two versions of the tutor described here incorporated what we call Example Walkthroughs to guide students through the study of examples prior to problem solving. Students began declarative instruction by reading textual expositions of concepts followed by video clips illustrating the application of these concepts. Subsequently, instead of progressing into problem solving, students were interactively stepped through a set of example walkthroughs. Walkthroughs served to lead students through the reasoning necessary to solve problems. At each step, multiple-choice prompts served to help students make inferences necessary to select actions that would take them to the next step in the problem-solving process. Incorrect inferences, which may have resulted from inaccurate or partial encoding of important concepts, were remedied through messages that clarified the knowledge necessary to make appropriate inferences. In our first experiment, students in the expert condition received videos and walkthroughs emphasizing generative skills. Declarative instruction for those in the intelligent novice condition emphasized both generative skills and evaluative skills (including error correction). To rule out the possibility that the results observed in the first study were a result of differences in the example walkthroughs, a second study was conducted. In the second study, students in both groups received example walkthroughs emphasizing generative and evaluative skills as part of their declarative instruction.

In our first experiment, students trained with intelligent novice feedback performed significantly better on tests of problem solving (effect size: 0.50), conceptual understanding (effect size: 0.59), transfer (effect size: 0.43), and retention (effect size: 0.33). We also examined online training data to determine if student performance during training mirrored these posttest measures. We also hoped to identify the point in the learning process where differences among the groups began to emerge. To do this, we examined the number of attempts required to solve training problems as a function of the opportunity to practice skill components associated with the six types of problems represented in the tutor. The *y*-axis in Figure 5 plots a performance measure, namely, the number of attempts required to write a correct formula, possibly with the help of the tutor. The *x*-axis plots the number of practice opportunities for problems of each type. Each point on the
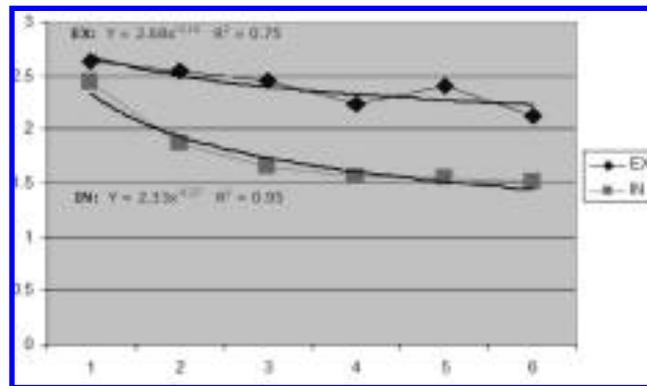

FIGURE 5   Comparison of learning curves associated with expert (EX) and intelligent novice (IN) versions of the spreadsheet tutor. Intelligent Novice students learn at a faster rate.

graph averages across the six types of problems represented in the tutor. Learning trends were estimated using best fitting power curves. A comparison of learning curves associated with the two tutorial conditions reveals that students in both groups start off performing at approximately the same level. However, over the course of successive attempts students in the intelligent novice condition learn at a faster rate. A repeated measures analysis of covariance showed a significant Feedback × Opportunity interaction. Our learning curve analysis suggests that effects of intelligent novice feedback begin relatively early in the learning process, and that the differences in the two groups persist over the course of training.

Results associated with our second experiment, in which students in both groups received identical declarative instruction, were similar to those obtained in Study 1. Students in the intelligent novice condition performed significantly better on tests of problem solving (effect size: .62), conceptual understanding (effect size: 1.05), transfer (effect size: 0.78), and retention (effect size: 0.70). Though the pattern of results is the same across studies, effect sizes in our second study appear larger. This difference may simply be random variation. It should be noted that, within each study, entering characteristics of students in each experimental group—as measured by a math test, computer skills experience assessment, and spreadsheet pretest—were the same.

DISCUSSION

Immediate feedback following errors has been criticized on grounds that it may prevent learners from exercising skills that are important for performing tasks outside the training environment. These include error-detection and error-correction skills. However, as Corbett and Anderson (2001) have pointed out, merely delaying feedback may be necessary, but not sufficient to promote the metacognitive processes of detecting, correcting, and learning from errors. Inappropriately designed delayed feedback can contribute to unproductive floundering and frus-

tration. Corbett and Anderson suggested that it may be necessary to provide direct feedback and support for these skills.

The theoretical perspective presented in this article recasts the feedback debate. It shifts the argument concerning when to intervene following errors from a focus on latency to a focus on the model of desired performance. Feedback based on an intelligent novice model does not simply provide students with an opportunity to engage in evaluative skills as delayed feedback would. Instead, an intelligent novice system explicitly models error detection and correction activities and supports students in the exercise of these skills. The intelligent novice tutor described here was designed to provide immediate feedback. But it does this on the basis of a cognitive model that includes error detection and correction activities as an integral part of the learning process. Feedback based on an intelligent novice model provides a practical way for facilitating the joint exercise of generative and evaluative skills in Cognitive Tutors, while minimizing the potential for unproductive floundering.

A comparison of student performance under the expert and intelligent novice conditions attests to the effectiveness of intelligent novice model feedback on a variety of different measures. During training, students in the intelligent novice condition learned at a faster rate. Students in the intelligent novice condition outperformed students in the expert condition on a variety of tasks—including performance on isomorphs of training tasks, tests of conceptual understanding, transfer tasks, and retention tests following an 8-day retention interval. The analysis and results presented here suggest that the joint exercise of generative and evaluative skills can contribute to better learning outcomes than a focus on generative skills alone.

The perspectives presented in this article are relevant to activities described by a variety of metacognitive frameworks. For instance, Butler and Winne (1995) identified several skills that characterize self-regulated learning. These skills include interpreting and defining task, selecting tactics and strategies, monitoring progress, and using feedback to reflect on the appropriateness of actions. Though the emphasis of this article has been on monitoring and reflection, the approach described here can be employed to model skills such as task interpretation and strategy selection. Our analysis suggests that explicitly modeling metacognitive skills, and using these models to scaffold student performance, is likely to lead to more effective and efficient learning than approaches that merely give students an opportunity to apply these skills.

REFERENCES

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP Tutor. *Cognitive Science, 13,* 467–506.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences, 4*(2), 167–207.

Bangert-Drowns, R. L., Kulik, C. -L. C., Kulik, J. A., & Morgan, M. (1991). The instructional effect of feedback in test-like events. *Review of Educational Research, 61,* 213–238.

Bjork, R. A. (1994). Memory and metamemory considerations in the training of human beings. In J. Metcalfe & A. Shimamura (Eds.), *Metacognition: Knowing about knowing* (pp. 185–205). Cambridge, MA: MIT.

Bruer, J. (1993). The Mind's Journey from Novice to Expert. *American Educator, 17*(2), 6–46.

Butler, D. L., & Winne, P. H. (1995). Feedback and self regulated learning: A theoretical synthesis. *Review of Educational Research, 65,* 245–281.

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,* 245–252.

Corbett, A. T., Koedinger, K. R., & Hadley, W. H. (2001). Cognitive Tutors: From the research classroom to all classrooms. In P. S. Goodman (Ed.), *Technology enhanced learning: Opportunities for change* (pp. 235–263). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Fox, B. (1993). Correction in tutoring. *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society.* University of Colorado, Boulder.

Hendry, D. G., & Green, T. R. G. (1994). Creating, comprehending, and explaining spreadsheets: A cognitive interpretation of what discretionary users think of the spreadsheet model. *International Journal of Human–Computer Interaction, 40,* 1033–1065.

Lee, A. Y. (1992). Using tutoring systems to study learning: An application of HyperCard. *Behavior Research Methods, Instruments, & Computers, 24*(2), 205–212.

Lepper, M. R., & Chabay, R. W. (1988). Socializing the intelligent tutor: Bringing empathy to computer tutors. In H. Mandl & A. Lesgold (Eds.), Learning issues for intelligent tutoring systems (pp. 242–257). New York: Springer.

Lewis, M. W., & Anderson, J. R. (1985). Discrimination of operator schemata in problem solving: Learning from examples. *Cognitive Psychology, 17,* 26–65.

Mathan, S., & Koedinger, K. R. (2003). Recasting the feedback debate: Benefits of tutoring error detection and correction skills. In H. U. Hoppe, M. F. Verdejo, & J. Kay (Eds.), Artificial intelligence in education: Shaping the future of learning through intelligent techolgies. *Proceedings of AI-ED 2003* (pp. 39–46). Amsterdam, The Netherlands: IOS.

Merrill, D. C., Reiser, B. J., Merrill, S. K., & Landes, S. (1995). Tutoring: Guided learning by doing. *Cognition and Instruction, 13,* 315–372.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences, 2,* 277–306.

Nardi, B. (1993). *A small matter of programming: Perspectives on end user computing.* Cambridge, MA: MIT.

Nathan, M. J. (1998). Knowledge and situational feedback in a learning environment for algebra story problem solving. *Interactive Learning Environment, 5,* 161–180.

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.

Schmidt, R. A., Young, D. E., Swinnen, S., & Shapiro, D. C. (1989). Summary knowledge of results for skill acquisition: Support for the guidance hypothesis. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15,* 352–359.

Schommer, M. (1993). Epistemological development and academic performance among secondary students. *Journal of Educational Psychology, 85,* 406–411.

Schooler, L. J., & Anderson, J. R. (1990). The Disruptive Potential of Immediate Feedback. In M. Piattelli-Palmarini (Ed.), *Proceedings of the Annual Conference of the Cognitive Science Society* (pp. 702–708). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.