THE JOURNAL OF THE LEARNING SCIENCES, 2(3), 277-305
Copyright © 1992, Lawrence Erlbaum Associates, Inc.

Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems

Douglas C. Merrill
Princeton University

Brian J. Reiser
Princeton University

Michael Ranney
University of California-Berkeley

J. Gregory Trafton
Princeton University

There has been much debate about instructional strategies for computerized learning environments. Many of the arguments designed to choose between the various philosophies have appealed, at least implicitly, to the behavior of effective human teachers. In this article, we compare the guidance and support offered by human tutors with that offered by intelligent tutoring systems. First, we review research on human tutoring strategies in various domains. Then we investigate the capabilities of a widely used technique for providing feedback, model tracing. Finally, we contrast the types of guidance and support provided by human tutors with those in intelligent tutoring systems, by examining the process of recovering from impasses encountered during problem solving. In general, the support offered by human tutors is more flexible and more subtle than that offered by model tracing tutors, but the two are more similar than sometimes argued.

Individualized instruction is often thought to be the most effective form of instruction, particularly for problem-solving domains (e.g., Bloom, 1984;

Requests for reprints should be sent to Brian J. Reiser, Institute for the Learning Sciences, 1890 Maple Avenue, Evanston, IL 60201.

Cohen, Kulik, & Kulik, 1982). The driving motivation for early work in intelligent tutoring systems was the desire to capture the effective behaviors of human tutors, thereby creating an optimal educational tool (Carbonell, 1970; Collins, Warnock, & Passafiume, 1975).

Since the inception of the field, many interactive learning environments have been developed, representing a number of different approaches for delivering on the educational promise of computerized learning tools. These different approaches include systems that are very guiding and provide directive feedback to ensure that students do not flounder during problem solving (e.g., Anderson, Boyle, & Reiser, 1985; Goldstein, 1982). Other researchers argue that the best use of computers in instruction is to provide a context in which students can explore a domain and learn by discovering its principles (Papert, 1980; Schank & Farrell, 1987; Schwartz, 1989). These systems provide tools for a student to experiment with the domain, but typically do not track whether the student has embarked upon a poor strategy or made errors, and hence do not intervene in such situations. Other systems lie somewhere between these approaches, adopting the attitude of a coach that intervenes as little as possible, offering students suggestions primarily upon request or when the system determines that an error could be grossly counterproductive (e.g., Burton & Brown, 1982; Lesgold, Lajoie, Bunzo, & Eggan, 1991). Such systems do not offer the full freedom of an exploratory environment, but also do not direct the students' actions as much as some tutors.

A recent focus of research on providing guidance in an intelligent tutoring system is the technique of *model tracing* (Anderson, Boyle, Corbett, & Lewis, 1990; Anderson, Boyle, Farrell, & Reiser, 1987; Anderson et al., 1985), in which the students' problem-solving steps are compared with the reasoning of an underlying domain expert. This matching is used to provide ongoing feedback to students while they progress through a problem. Evaluations of model tracing systems have demonstrated their effectiveness in facilitating students' learning and problem solving in several mathematics and computer programming domains (Anderson et al., 1985; Anderson et al., 1990).

The model tracing methodology produces a tutor that may intervene fairly frequently, and the interventions are typically very directive. This has led some researchers to raise concerns about this approach. Some researchers have argued that there may be drawbacks to the model tracing approach because it does not allow students to learn by finding their own errors and repairing them, nor to reflect on multiple solution strategies (Collins & Brown, 1988; Schoenfeld, 1988). Others have argued that diagnostic feedback, such as is offered by model tracing systems, may actually interfere with students learning the metacognitive skills that enable them to manage their own learning (Scardamalia, Bereiter, McLean, Swallow, & Woodruff, 1989). Finally, some researchers have questioned whether the directive nature of guiding intelligent tutoring systems can achieve the benefits of the more gentle and indirect guidance of human tutors (Fox, 1991; Lepper & Chabay, 1988), or that the problem-solving models could be sophisticated enough to teach more than simple procedural skills (Ridgway, 1988).

The model tracing methodology is becoming an increasingly widespread technique for implementing guidance in an intelligent tutoring system, yet the concerns that have been raised suggest that there may be drawbacks to this approach. Therefore, it is important to consider the extent to which model tracing intelligent tutoring systems have succeeded in accurately modeling human tutors. Are systems such as model tracing tutors that intervene and guide students' problem solving faithful to the behavior of human tutors? Do these systems achieve the effectiveness exhibited by good human tutors?

In this article, we argue that model tracing tutors indeed capture crucial aspects of the behavior of human tutors. Both human and computer tutors support students' reasoning and ensure that the problem solving remains productive. They do this by intervening to ensure that errors are detected and repaired and that students can work around any known (or yet undiscovered) impasses. However, we shall see also that the nature of this support differs somewhat between human and computer tutors. Human tutors allow their students to do more of the process of recovering from impasses than computer tutors, and thus human tutors may allow students to feel more in control of the interaction.

To evaluate these issues, we need to briefly survey what is known about human tutoring, and then examine how guidance can effectively be implemented in computer tutors. We focus our discussions on the technique of model tracing because this methodology has been perhaps the most extensively evaluated of the proposals for embedding tutorial guidance in a computer. Following the discussions of the capabilities of computer tutors, we turn to an analysis that evaluates the common intervention strategies of human and computer tutors and reveals some of the differences between them.

WHY ARE HUMAN TUTORS EFFECTIVE?

A number of studies have documented the effectiveness of human tutors (Bloom, 1984; Cohen et al., 1982; Lepper, Aspinwall, Mumme, & Chabay, 1990), supporting a common intuition that when a student has difficulties, the best course of action is to provide the student with one-on-one instruction. What do tutors do that is so effective? In this section, we examine several studies of tutoring and review the critical features proposed by investigators of human tutoring.

Studies of individualized tutoring in a variety of domains have demonstrated improvements in both learning time and subsequent performance. Several recent studies have begun to analyze the pedagogical strategies of tutors to ascertain what underlies their effectiveness (e.g., Fox, 1991; Leinhardt & Ohlsson, 1990; Lepper et al., 1990; Lepper & Chabay, 1988; McArthur, Stasz, & Zmuidzinas, 1990; Merrill, Reiser, & Landes, 1992; Putnam, 1987). These studies suggest a reason why tutoring is so effective: Experienced human tutors maintain a delicate balance, allowing students to do as much of the work as possible and to maintain a feeling of control, while providing students with enough guidance to keep them from becoming frustrated or confused.

By allowing the students to do most of the problem solving, tutors allow them to learn by doing. A central part of the learning process occurs when students attempt to apply the instructional material to solve problems for themselves (Anderson, 1983; Anzai & Simon, 1979). Important learning may occur when students encounter obstacles, work around them, and explain to themselves what worked and what did not (Chi, Bassok, Lewis, Reimann, & Glaser, 1989; Ohlsson & Rees, 1991; VanLehn, 1990). However, this type of learning has potential cognitive and motivational pitfalls. Students trying to solve problems can expend much time and effort pursuing blind alleys because of errors or poor strategies. Of course, in some cases students may learn something valuable while searching for a solution. In many cases, however, such episodes leave students confused and frustrated; it may be difficult to return to the point in the solution before the error occurred, and misattributing the sources of a difficulty may even result in acquisition of faulty knowledge (Lewis & Anderson, 1985; Sweller, 1988). The assistance of a tutor enables a type of *guided learning by doing,* in which the students reap the rewards of active problem solving while the tutors minimize the dangers. In this way, tutoring has both cognitive and motivational advantages.

Lepper and his colleagues have characterized the impact of tutoring on the motivation of students (Lepper et al., 1990; Lepper & Chabay, 1988). They argued that tutors are highly interactive, yet intervene very indirectly during the learning sessions. Thus tutors help students solve problems successfully while minimizing their own apparent role in the success. Tutors manage to promote a sense of challenge, provoke curiosity, and maintain students' feeling of control. For example, Lepper et al. (1990) found that experienced tutors tended to emphasize the difficulty of the upcoming task, thus allowing failure to be attributed to something other than the students' lack of ability. This strategy was particularly effective for students who had already experienced difficulties in school. In addition, Lepper et al. found that expert tutors tended to draw the students' attention to an error and then provide a second chance at the solution rather than giving explicit corrective feedback. Furthermore, the tutors in the study usually asked the student questions instead of providing explicit direction. The style of the feedback is important—Lepper et al. found that students chose to solve more difficult problems when tutors used this indirect style than when tutors used a more direct style. These results suggest that expert tutoring involves the use of subtle cues to guide and support students, thus maximizing their motivation to learn.

Fox's (1991) analyses of discourse interactions between tutors and students are generally consistent with Lepper's view of tutorial interactions. She argued that tutors provide a "safety net" during problem solving, so that student errors are kept to a minimum. In Fox's study, tutors employed subtle techniques to notify students that a step in the solution required repair. These tutors provided frequent feedback, typically indicating very briefly their agreement with each step. A short hesitation (often less than 1 sec) in responding with an "okay" typically led students to assume that something was amiss with the current step. After this clue, students frequently corrected the mistake. When more explicit help was required, the tutor focused the student's attention on the part of the solution that required modification or on information potentially useful for repairing the error. In Fox's view, tutors usually avoid telling students they are wrong or precisely how a step is incorrect; instead they lead students to discover the error themselves. Fox's results present a picture of tutoring as a guided problem-solving session, in which the student takes steps and corrects wrong paths, while the tutor helps the student stay on track. In some cases, students overtly request guidance in the form of goals to set, missing information to fill in, or explicit confirmation regarding the correctness of a step, but in most cases such requests are not necessary because the tutor provides such information through hints, leading questions, verbal agreement, and other indirect methods. In fact, the amount of help offered at every step is striking, particularly because the tutors need to say very little to accomplish this.

Not all analyses suggest that tutoring effectiveness relies on such implicit communicative methods, however. For example, McArthur et al. (1990), in studies of remedial tutoring, argued that tutors are much more directive than suggested by Fox (1991) and Lepper et al. (1990). McArthur et al. found that their tutors carefully structured the task the student was to follow, similar to the behavior of classroom teachers studied by Leinhardt and Ohlsson (1990). The tutors in the McArthur et al. study made sure that the student was aware of the current solution goals at all times. These tutors remediated errors upon occurrence, including not only pointing out where the error lay but often also suggesting a technique for solving the problem.

Like the analyses of Fox and Lepper, the work of McArthur et al. suggests that human tutors follow students' solutions very carefully and redirect students when they encounter impasses. Fox and Lepper argued that this redirection is very subtle, often involving only a pause or a leading question, whereas McArthur et al. found evidence that this redirection may be more directive, at least in remedial tutoring.

These studies suggest that impasses in problem solving present important opportunities for tutorial intervention. Much of the tutorial interactions observed in these studies were centered around episodes in which a student became stuck, discovered an error, or committed an error. The central role of errors in structuring tutorial interventions has also been stressed by Littman and his colleagues (Littman, 1991; Littman, Pinto, & Soloway, 1990). They argued that the content and timing of human tutorial feedback depend upon the error's context. Littman et al. gave tutors completed computer programs that contained marked errors and asked the tutors to describe how they would teach these students. Their tutors attempted to determine what misconceptions were embodied in the errors and how important each one was, and then used this information to set up tutoring plans. These plans corrected the most important bugs first, the less important ones later. The tutors used knowledge about categories of bugs and their potential causes to determine the relative importance of the errors. If a bug showed that a student had a poor understanding of earlier material, the tutors considered it more important. Similarly, if the knowledge gained from correcting one bug facilitated fixing another bug, the tutors set up the interventions to capitalize on this fact, focusing the student's attention on the more central bug. Finally, these tutors suggested a repair for an error that would mask other errors during the program's execution before remediating anything else. Thus, the tutors of Littman et al. modulated their responses based on how critical the student errors were.

Merrill, Reiser, and Landes (1992) also found that a tutor's policy on intervention seemed to rely upon the context of the student's error. In some cases, principally syntactic errors, the tutor immediately told the student what to do to fix the error. However, when the error involved misunderstandings about the actual behavior of objects in the domain, the tutor often focused the student on the features of the solution that were incorrect. In contrast, when the student began working on an inappropriate plan or forgot an important goal, the tutor often helped reformulate the goal that the student should pursue. In still other cases, the tutors simply ignored certain errors, returning to them at a later, more useful point. It appeared that the tutors modulated their intervention depending on the potential learning consequences of the error. Tutors quickly corrected errors that would be distracting and might lead to floundering, quickly focused the students on more serious problematic components of a solution so that they could fix them, and, finally, withheld comments or offered much less directive feedback about errors that might lead to productive learning episodes later.

The type of support that tutors provide students mirrors that of students in *cognitive apprenticeship* (Collins, Brown, & Newman, 1989). In these situations, teachers model the desired skill, coach students as they practice the skill, and gradually withdraw their support as students gain proficiency. The tutorial situation is similar, in that tutors provide only as much support as is necessary to help students overcome impasses, and withdraw the support as soon as it is no longer needed.

This summary has revealed several perspectives on tutorial feedback. Fox (1991) and Lepper et al. (1990) argued that tutors use very subtle feedback upon errors or obstacles to maximize students' problem solving success. The McArthur et al. (1990) results also suggest that tutors follow students' solutions very carefully, but indicate that this feedback can be very directive. McArthur et al. argued that tutors give explicit feedback, sometimes even telling students how to solve a problem, and carefully structure students' tasks by reminding them of problem goals. Littman et al. (1990) and Merrill et al. (1992) argued that the context of the error is critical in determining feedback.

These analyses demonstrate that human tutoring is a highly interactive process in which the tutor employs constant feedback to support students' problem solving. The tutors provide enough direction to help students plan and create a solution, using error feedback and hints to prevent them from becoming lost in unprofitable solution paths. Human tutorial guidance appears to be structured, in large part, around the impasses that students encounter. The content and timing of feedback appear to depend critically on the consequences of the particular error or impasse encountered. Sometimes tutors allow students to discover their own errors, but might intervene immediately at other times. Regardless of the timing of the intervention, the feedback is carefully designed to allow students to do as much of the work as possible while still preventing floundering. The feedback can be quite subtle, however, because tutors appear to accomplish much of their interventions without being obvious that they are directing the student. This may account for the finding that tutored students feel very much in control of their own learning.

If we want to model human tutorial feedback on computers, we must consider how an intelligent tutoring system could offer similar highly interactive yet subtle guidance. How successful are current attempts to achieve these goals in a computerized tutor? What are the obstacles facing computer tutors? In the next section, we explore these questions by examining one highly successful technique used in intelligent tutoring systems, model tracing, and by evaluating its successes, the difficulties with the methodology, and its potential.

WHAT CAN BE ACHIEVED IN COMPUTER TUTORS?

As the last section showed, human tutors provide highly interactive feedback to support students' problem solving. This feedback is provided through the tutors' careful monitoring of students' problem solving. When students go off the track or get stuck, tutors generally intervene to help them recover from the impasse. The feedback from the tutors is ongoing, providing confirmation or questions at nearly every step, but the feedback they provide is very subtle.

Therefore, to model the abilities of human tutors, an intelligent tutoring system must be able to follow students' reasoning during problem solving. If a system follows a student's solution step by step, it can check that each action the student takes is a legal construction in the domain and track whether the student is still on a viable solution path. Feedback can be provided upon errors, and hints can be suggested if students are unsure how to proceed.

One class of successful techniques for following students' solutions and identifying errors entails matching the students' problem-solving steps with the reasoning of an underlying rule-based domain expert (Anderson et al., 1985; Clancey, 1987; Goldstein, 1982; Kimball, 1982). In the model tracing methodology, this matching is used as the basis for providing ongoing feedback to students while they progress through a problem (Anderson et al., 1985; Anderson et al., 1987; Anderson et al., 1990). The general strategy in model tracing systems is to present a problem for the student to solve, track the student's progress step by step, and intervene with explanatory feedback upon an error or a request for help. If the student's step is one that would be produced by executing one of the correct rules considered by the system, the tutor silently follows the student's path through the problem. In contrast, if the step is illegal or follows a strategy unlikely to succeed, the tutor intervenes with a suggestion. For example, an incorrect use of a geometry theorem might trigger a brief explanatory message associated with the buggy problem-solving rule that captures the misconception. In this situation, the feedback helps the student diagnose the error and suggests a way to approach its repair. In this example, the feedback might suggest that the theorem used by the student is a good step toward the goal, but that the student may have assumed one of the necessary premises rather than proving all these premises before applying the theorem. This type of feedback relies not only on analysis of the surface behavior of the solution, but also on inferences about the students' intended plan, as well as analyses of the potential plans for solving the problem (Anderson et al., 1985). A model tracing system may also respond to an error by finding a correct rule embodying an appropriate action in the current problem-solving context. In this case, the tutor guides the student toward a correct replacement step for the error.

The model tracing methodology has formed the basis for a number of intelligent tutors that teach computer programming (Anderson, Conrad, & Corbett, 1989; Anderson & Reiser, 1985; Reiser, Anderson, & Farrell, 1985; Reiser, Friedmann, Kimberg, & Ranney, 1988), proof construction in geometry (Anderson, Boyle, & Yost, 1986), solving algebraic equations (Milson, Lewis, & Anderson, 1990), and calculus (Singley, 1990).

There are two key steps to evaluating whether a model tracing tutor indeed facilitates students' learning. The most straightforward approach is to compare students learning with the tutor to other students learning in the standard learning situation, such as students reading a textbook and solving problems on their own. However, many intelligent tutoring systems also provide tools designed to support reasoning apart from the model tracing guidance, such as the structure-based editor in the Carnegie-Mellon University LISP Intelligent Tutoring System (Corbett, Anderson, & Patterson, 1990; Reiser et al., 1985) or the visual representations in the Geometry Tutor (Anderson et al., 1986) and in the GIL (Graphical Instruction in LISP) tutor (Merrill, Reiser, Beekelaar, & Hamid, 1992; Reiser, Kimberg, Lovett, & Ranney, 1992). Thus, it is also important to compare the model tracing tutor to a version that lets students use the same representational tools but does not provide model tracing feedback.

Several studies have demonstrated the first point, that model tracing tutoring systems indeed support students' problem solving and facilitate their learning of the target domain. The advantage of model tracing intelligent tutoring systems has been demonstrated in several domains. Anderson et al. (1990) found that students receiving help from the CMU LISP Intelligent Tutoring System mastered the material more quickly and performed better on posttests. Similarly, students using GIL learned the material in an introductory LISP curriculum more quickly and with less difficulty than students using the standard environment (Reiser, Ranney, Lovett, & Kimberg, 1989; Reiser, Beekelaar, Tyle, & Merrill, 1991). Anderson et al. (1990) also found that students using the Geometry Tutor performed better on posttests than students instructed in a traditional classroom.

It is necessary to seek the second type of evidence, separating the advantages of model tracing from the support of the tutor's interface. To this end, we have compared students learning to program using the model tracing version of GIL with students using an exploration-based version of

GIL that did not provide model tracing feedback (Reiser, Copen, Ranney, Hamid, & Kimberg, 1991). The model tracing tutor intervened upon errors and offered guidance to help students correct the errors and work around impasses. The exploration-based system provided the same graphical representation and a set of tools for editing and correcting programs, but did not intervene to offer suggestions or comment on the students' strategy. In the absence of such assistance, the exploratory students took almost twice as long to complete the curriculum. Not surprisingly, the exploratory students were better at finding bugs in a program on a debugging transfer test, presumably because they had more practice in finding and repairing their own errors than did the model tracing students. However, the two groups were equivalent on all the learning posttests that assessed their ability to construct programs and make predictions about their behavior, even though the model tracing students completed the learning sessions considerably more quickly.

Corbett and Anderson (1991) found also that the CMU LISP Intelligent Tutoring System helps students master programming more quickly and with better learning than students who use the same structured editor interface but without the tutor's model tracing guidance. The results of these two sets of studies suggest that model tracing tutors do indeed facilitate learning in some domains, and that their effectiveness derives at least in part from the model tracing guidance these systems provide.

The Reiser et al. (1991) results suggest also that there may be motivational benefits of model tracing, at least for lower ability students. The lower ability students, who received more active guidance from the model tracing system, exhibited more positive judgments about the domain and held higher opinions of their abilities than comparable students in the exploratory condition. These results are consistent with Snow and Lohman's (1984) review of learning outcomes, in which they suggest that structure and guidance are more important for low ability learners. Similarly, Schofield, Evans-Rhodes, and Huber (1990) observed much greater student involvement and motivation among students working with the Anderson et al. (1986) geometry tutor than is common for students in high school geometry classes. These results suggest that the guidance provided by intelligent tutors may have a positive impact on motivation, in that it prevents the quite frustrating floundering episodes suffered by students working alone. A successful tutoring system can bring a difficult domain within the student's competence and provide a challenging task with less danger of failure.

The next step in analyzing how model tracing tutors can assist learning is to investigate which aspects of their guidance are important for students' learning. For example, it may be that model tracing tutors are effective simply because they alert students to parts of solutions that require repair, and the explanations they present upon errors are unnecessary. Alternatively, explanatory feedback may help students understand their errors and learn more by fixing them. McKendree (1990) found that the content of feedback did indeed affect students' learning of geometry. For example, simply telling the student that an error had occurred was much less useful than reminding the student of the current goal or pointing out a feature of the error. McKendree argued that such feedback is particularly important for learning the goal structure of solutions.

Reiser, Connelly, Ranney, and Ritter (1992) compared the effectiveness of several feedback strategies in the GIL tutor. Subjects who received minimal feedback, in which erroneous steps were noted by the tutor without further information, performed more poorly during the learning sessions and on posttests than subjects who received guidance about the location of the error in the current step. Explanatory feedback characterizing which features of the solution were incorrect provided additional benefits for learning. With increasing quality of feedback, subjects made fewer errors, deleted fewer (correct) partial solutions, were faster to solve the assigned problems, and performed better on posttests.

The results reviewed in this section demonstrate that model tracing tutors facilitate students' learning, helping them learn more quickly, with less difficulty, and with better subsequent performance than students working in the standard environments. Second, model tracing guidance produces additional benefits beyond the tutor's helpful interface. Finally, explanatory content is responsible for at least part of the model tracing's effectiveness. This explanation helps students fix errors more easily than they could alone, even if notified that an error had occurred. These results suggest that there is promise in the model tracing methodology for implementing feedback in a computerized tutor.

In the previous section, we argued that human tutors follow students' reasoning carefully and provide feedback at every step. The results reviewed here suggest that model tracing can profitably provide some components of this type of feedback for students. In the next section, we look in more detail at the important similarities and differences between the feedback strategies of human tutors and computer tutors. We focus on the tutor's assistance in overcoming impasses because this appears to be a key component in the tutor's efforts to assist learning. Do model tracing tutors respond differently than human tutors when a student encounters an impasse?

EXAMINING PEDAGOGICAL TECHNIQUES OF COMPUTER TUTORS AND HUMAN TUTORS

An intelligent tutoring system that follows students' reasoning and provides feedback on errors using the model tracing methodology can achieve significant pedagogical improvements over students working without assistance. However, there are important differences in the styles and abilities of computer tutors and human tutors. In order to evaluate the potential for achieving tutorial guidance in computer tutors, we must examine the possible consequences of these differences. In this section, we compare various ways in which computer tutors and human tutors support and guide students' learning. We consider what lessons can be learned from human tutors for the design of interactive learning environments.

Our discussion focuses on how tutorial feedback can assist students' learning. Questions about the type of support and guidance students should receive are at the core of the design of interactive learning environments. What type and how much guidance should a learning environment provide to ensure that the problem solving is productive, without overly encroaching on the student's active role in the problem solving? The debates about educational software echo the vigorous earlier debates of educational theorists about how best to provide instruction, in which some theorists argued that students should be free to control their own learning and learn through exploration (Bruner, 1961; Davis, 1966) and others stressed the importance of structure and guidance for students (Ausubel, 1963; Skinner, 1968). What do the observations of human tutors suggest about this controversial issue?

An expert tutor has to satisfy two pedagogical goals that are potentially in conflict. One goal is to leave students in control, free to reason through problems for themselves, making mistakes, detecting them, and learning by recovering from those errors and working around impasses. This class of potential benefits leads some researchers to argue that computerized learning environments should primarily support students' exploration, providing tools that empower students to learn by discovery (e.g., Burton & Brown, 1982; Schank & Farrell, 1987; Schwartz, 1989; Shute, Glaser, & Raghavan, 1989). If the system intervenes more than is necessary and is overly restrictive and guiding, it may interfere with the benefits of active learning by doing.

On the other hand, a tutor also has the goal of preventing students from becoming confused and frustrated and ensuring that they learn from their problem solving. There are dangers in leaving students too free to explore. Sweller (1988) argued that the problem solving of beginning students, which is often based upon weak methods and extensive search, may lead to less effective learning than studying worked out example problems. Lewis and Anderson (1985) argued that if a solution to a problem is obtained through excessive floundering, it may be difficult for students to remember what path they took to the solution; hence it may be difficult to learn from that experience. Anderson et al. (1985) stressed the importance of providing error feedback while the buggy knowledge that led to the error is still active and available. Burton and Brown (1982), although stressing the importance of students learning without being overly directed, argued that discovery should be guided by a coach who intervenes occasionally upon errors to ensure that errors become productive learning experiences. VanLehn (1988) argued that students learn when they encounter and overcome impasses; hence this view also suggests the importance of providing guidance when the student is currently facing the impasse, rather than when the student has completed an erroneous solution. Indeed, studies of feedback in a variety of instructional contexts find that immediate feedback is more effective than feedback received after a delay (Kulik & Kulik, 1988). These arguments suggest that although there are benefits when students learn by solving problems, the problem solving may be more effective if it is somewhat guided.

Each type of learning environment has its tradeoffs. A system that provides too much guidance may interfere with the active nature of learning by doing. If error feedback or hints are too easily available, students may not commit sufficient effort to reasoning through an answer on their own, resulting in poorer learning than without feedback (Bangert-Drowns, Kulik, Kulik, & Morgan, 1991; Kulhavy, 1977). Furthermore, if students come to rely on the system's help to find and fix errors, they may learn less from those errors (Schank & Farrell, 1987) and learn less about how to manage future errors (Collins & Brown, 1988). On the other hand, learning without feedback or with delayed feedback can lead to counterproductive floundering from which it may be difficult to learn the target knowledge (Lewis & Anderson, 1985; Sweller, 1988). Blocking long episodes of floundering caused by confusing errors can facilitate learning with no apparent cost in later performance (Carroll & Carrithers, 1984).

There is likely to be no simple answer concerning the ideal learning environment—there are clear advantages to both exploration and guidance. Furthermore, the most appropriate type of learning situation may depend upon factors such as the confidence and ability of the learners, whether the material is an early topic in the curriculum or advanced material, and how easily students can elicit information on their own in the domain to evaluate the success of their reasoning. Nevertheless, guidance will sometimes be required to prevent unproductive floundering. Therefore, when providing this guidance it seems sensible to attempt to adopt the techniques of the most successful model available for providing guidance—human tutors.

FEEDBACK AND THE PROCESS OF RECOVERING FROM IMPASSES

When solving problems, students take steps toward the problem solution, generate subgoals, and then achieve or abandon them as they progress.

Typically, students make errors or reach points where they are unable to proceed. These impasses are crucial events in the problem-solving episodes, at which learning can occur or the problem solving can go awry and become frustrating. Hence, it is at impasses that tutorial scaffolding of problem solving is most valuable. This guidance can help the student overcome the impasse, put the problem solving back on track, and learn from the event. In our comparison of human and tutorial feedback, therefore, we focus on tutorial actions that help students overcome impasses. In particular, we focus primarily on situations in which students repair erroneous solutions with tutorial assistance.

Recovering from an error consists of a sequence of several components. First, the student must notice that an error has occurred. An error might be detected by generating situations to evaluate the current solution. An error also might be detected when the student has reached an impasse and cannot find a way to proceed (suggesting that a mistake earlier in the solution may have led to this dead end). Alternatively, an error might be detected through explicit tutorial feedback. Following the decision that an error has occurred, the student must locate which portion of the solution must be modified, including determining which features of the solution are at fault. Then, one must replace the erroneous portion of the solution with one that achieves the current subgoal. In addition, it may be useful to determine what misunderstanding caused the error.

As we shall see in the examples that follow, tutoring turns error recovery into a collaborative process, in which the tutor and student work together to repair errors. Tutorial interventions during problems solving could include assistance on some or all of these facets. We can characterize the directiveness of a tutorial response by identifying the portion of the processes in this sequence performed by the tutor, with the remainder left to the student. In this section, we present an analysis of this collaborative error repair process. The analysis will suggest that human and computer tutors are strikingly parallel in the type of reasoning they assist students in performing. The principal difference between them concerns the sharing of work in the collaboration. Human tutors allow their students to do more of the error recovery process than current computer tutors do. To investigate this assertion, let us consider the ways in which a human or computer tutor can collaborate in the impasse recovery process.

At one extreme of the spectrum, a tutor might generate a situation where students are led to discover their own errors. Examples of this sort of feedback are shown in Table 1. Instead of directly informing the student that an error has occurred, the tutor constructs a situation in which the student realizes that the solution is partially incorrect, so the student can then take over the error correction. This process is an important strategy

TABLE 1
Generating Situations to Help Students Detect Errors

A. Human LISP programming tutor (unpublished data discussed in Merrill, Reiser, & Landes, 1992).

Context: The student has misordered two cases of a conditional expression in her solution.
Tutor: Say we put in 5? We put 5 in, and we go to the first case, and it would say *nil*—
Student: Oh, it would say yes then [in the second case], oh, that's right. So this [case] should come after the number one [case], that's right.

B. Human inquiry teacher (Collins & Stevens, 1982).

Context: Suppose a student suggests they do not grow rice in British Columbia because it is too mountainous.
Tutor: If British Columbia were flat could they grow rice there?

used by inquiry teachers to help students detect their own misconceptions (Collins & Stevens, 1982).

In addition, tutors could inform the student that an error has occurred in a new addition to the solution. This not only informs the student that there is a problem with the current solution, but also gives a clue about where to look for the error—namely, the latest step. In fact, Fox (1991) argued that tutors confirm every step that the student makes, and that a delay of confirmation of 1 to 2 sec informs the student that an error has occurred. Thus, the tutor conveys information by actually saying nothing. Students notice quickly that the tutor has failed to respond to a step and may be thereby indicating that they are no longer on the right track. Table 2 provides a typical example from Fox's analyses of this type of repair; in the absence of the usual confirmation, the student questions whether the step is correct and invites assistance.

Tutors may give the same information about the error in a more direct manner, as shown in Table 3. This table contains two examples in which tutors verbally intervene to alert the student to an error, after which the student takes over and repairs it with some additional assistance. In example B, the tutor tells the student that a mistake has occurred with the statement "There's a mistake there." This statement does not reveal exactly what feature of the solution is incorrect but only suggests that the student should reconsider the solution to attempt to repair it.

TABLE 2
Flagging an Error for the Student via the Absence of Confirmatory Feedback

Human mathematics tutor (Fox, 1991).

Student: because secant squared of theta is square root of [pause 0.8 sec]
Student: Can I do it that way?
Student: Can I say three minus one?
Tutor: Mmm. No, you want to say three squared. Because the secant is three.

TABLE 3
Flagging the Error for the Student via Direct Statements

A. Human physics tutor (Fox, 1991).
Student: Okay, so I guess I somehow have to . . . tangent of theta is going to be . . . sine of theta over cosine of theta. One over cosine of theta; so 3.
Tutor: Mkay. Now.
Student: Okay.
Tutor: Looking up here, just at what . . .
Student: Aha.
Tutor: . . . they've done. Cause I can tell, we're headed in the wrong direction.
Student: Yeah, they used to con—they use one of the pythagoreans.
Tutor: One plus tangent squared equals the secant squared.
Student: Secant squared.
B. Human arithmetic tutor (Lepper & Chabay, 1988).
Tutor: There's a mistake there. Can you find your mistake?
Student: What?
Tutor: Well, let's look and see. How much is nine and eight? [points to numbers]
Student: Nine and eight is seventeen.
Tutor: We put a seven [points] and carry the ten, right? Now add this column. [points]
Student: [erases the 5 in the column]
Tutor: Good for you.

Tutorial behavior may be even more directive than simply verbally pointing out an error, as shown by the examples in Table 4. Here, instead of just telling the student that an error has occurred, the tutors point out the feature of the solution that is erroneous. Consider example A in Table 4, taken from a tutor helping a student with addition. Although stated as a question, this sort of feedback essentially tells the student the feature of the solution that must be modified.

Table 5 shows feedback that is more directive still. In this table, the feedback not only tells the student that an error has occurred, where it occurred, and what the erroneous feature of the solution is, but also offers a principle of the domain that explains why the feature is erroneous. So, in example B, the tutor tells the student that the desired quantity is inappropriate, thus locating and pointing out the erroneous feature of the solution, and also tells the student about a physical law (Kirchoff's Law) that clarifies why the quantity cannot be measured. In contrast, in example C of Table 4 the tutor does not offer a principle of LISP that indicates why *LAST* should return a list, but rather simply points out that this property of the function is violated in the solution.

Another degree of guidance that can be provided upon errors or other impasses is setting a goal for the student, as shown in Table 6. By setting a goal for a new step, a tutor may tell the student not only where an error occurred and why it is wrong, but also may suggest how to repair it. For

TABLE 4
Directing Students to the Erroneous Feature of the Solution

A. Human arithmetic tutor (Lepper & Chabay, 1988).

Student answer:

$$\begin{array}{r} 537 \\ +23 \\ \hline 5510 \end{array}$$

Tutor: Now look at that again. Can you put two numbers down in one column here?

B. Human physics tutor (Fox, 1991).

Student: F, that's what I had a problem with, was F, they said if the electric force between them is equal to the weight.

Tutor: Okay.

Student: So I tried to look at the weight.

Tutor: And all's they give you is the mass.

Student: And it yeah. Oh, that's what it was, it was the mass.

Tutor: Yeah.

Student: Oh, I see, I want weight.

Tutor: You wrote down mass. Yeah, what's the difference between weight and mass.

Student: I used to know this, let's see.

Tutor: I think, I think what it is is that . . . what is, when you do gravity problems . . .

Student: Right. It's that . . .

Tutor: What do you always do?

Student: You have to multiply it by the . . .

Tutor: By G.

Student: G. Right. [Several interchanges deleted]

Student: So I need to multiply this time G.

Tutor: Right.

C. GIL (Reiser, Kimberg, Lovett, & Ranney, 1992).

Tutor: Using *LAST* on (*a b c d*) is a good idea. However *d* is not what *LAST* will produce. The correct output should be a list.

TABLE 5
Using a Domain Principle to Explain Why a Feature of a Solution is Erroneous

A. Geometry Tutor (Anderson et al., 1990).

Tutor: ASA requires 3 premises. You have picked two premises.

The statements you chose: $\llcorner EJX \cong \llcorner EJY$, $\llcorner EXJ \cong \llcorner EXK$, are not used by any rule.

I'm going to start you off with a new set of premises.

B. SOPHIE (Brown, Burton, & de Kleer, 1982).

Student: What is the current through Node 4?

Tutor: The current through a node is not meaningful since by Kirchoff's law the sum of currents through any node is 0. Currents can be measured through parts (e.g., current through C6) or terminals (e.g., current through the collector of Q2).

TABLE 6
Setting Goals to Assist Students' Problem Solving

A. Human LISP tutor (Unpublished data discussed in Merrill, Reiser, & Landes, 1992).
Student step: The student types *((t,* which is an incorrect way to begin the last case of a conditional. (In LISP, one uses two left parentheses in a conditional only if the following atom is a predicate. Here, the following atom is *t,* so only one left parenthesis should be used.)
Tutor: Now here you have a choice. You could either use *t* or the predicate.

B. WEST (Burton & Brown, 1976).
Tutor: You don't seem to be bumping very much. Bumps are hard to get but they are usually a good idea. One good example would be the expression: (1 * 2) + 2, which would give you a SHORTCUT and a BUMP!! So you could have been at 54 while I would have ended up at 40.

example, the tutor might reformulate the student's solution plan. Comparing the reformulated plan to the current step reveals the erroneous feature of the solution, and recovering from the impasse can begin by implementing the tutor's suggestions. For example, in example A of Table 6, the tutor tells the student to take one or the other possible action, namely, to use either the *t* construct or a predicate. The tutor's goal setting leads the student to try to figure out the differences between the tutor's proposal and the current step, thus identifying the error and helping to begin the repair process. A tutor may also set goals that help the student solve further problems more effectively, as in example B of this table.

Finally, an understanding of the misconception underlying the error may be useful in learning how to repair it. Part of the initial promise of intelligent tutoring systems arose from the notion that systematic misunderstandings underlie many student errors, and automatic recognition of these misconceptions could be used as the basis of computerized teaching tools that would remediate incorrect knowledge and provide individualized practice (Brown & Burton, 1978; Burton, 1982; Clancey, 1986; Ohlsson & Langley, 1988; Sleeman, 1982).

For example, such feedback might point out that the student had confused two programming constructs, or had neglected to distribute multiplication over both addends, and so on. Table 7 presents examples of this style of feedback. These tutors perform most of the error correction, telling the student that an error has occurred, where it lies, and suggesting how to repair it, leaving the student the responsibility for implementing the tutor's suggestions. Most importantly, in these examples the computer tutor suggests how the student may have reasoned incorrectly to produce the error. This understanding of faulty reasoning underlying an erroneous solution is often called *diagnosis.* Many computer tutors have offered diagnoses in feedback, often consisting of statements about the student's intentions in making a solution step or about the chain of reasoning that led

TABLE 7
Diagnosis of Buggy Reasoning

A. Proust (Johnson, 1986)—a tutor for Pascal programming.
Context: Line 12 in student's program: NEW := NEW + 1.
Tutor: It appears that you were trying to use line 12 to read the next input value. Incrementing NEW will not cause the next value to be read in. You need to use a READ statement here.

B. The Algebra Tutor (Anderson et al., 1990).
Student step: The student rewrites the equation $15 - 3x = -x$ as $15 = -4x$.
Tutor: $15 = -4x$ is wrong because you added $-3x$ instead of its inverse to $-x$. Try again.

C. GIL (Reiser, Kimberg, Lovett, & Ranney, 1992).
Problem: Extract the next to last element of a list, e.g., get *d* from *(a b c d e)*.
Student's step: Take *REST* of *(a b c d e)* to get the output *(b c d e)*.
Tutor: OK, I think I see what you are trying to do. Are you trying to use *RESTs* on *(a b c d e)* to get a list beginning with *d?* It's not a bad idea but unfortunately it won't be a very general solution. Remember that your program should work for an input list of any length not just for *(a b c d e)*.
[The student clicks the "More Info" button.]
Tutor: Think about a way to get *(d)* by working from the back of the list rather than from the front.

to the error. The ability of computer tutors to analyze the student's reasoning behind the error rather than merely report on the surface features of a solution that are incorrect has been considered a major achievement of intelligent tutoring systems over prior computer-assisted instruction (e.g., Clancey, 1986). For example, in example C of Table 7, although the student has not yet completed the erroneous plan, the tutor infers that the student is following a mistaken strategy and explains why this strategy will not produce an appropriate program. In the examples in Table 7, the tutors mention their inferences about the students' intentions in the error feedback.

COMPARISONS OF HUMAN AND COMPUTER TUTORS

Our characterization of the control of recovering from impasses is helpful in comparing the techniques of computer tutors with those of human tutors. We saw that a tutor might create a context in which the student is likely to detect that an error has occurred (Table 1), or might withhold confirmation, thereby suggesting that one has occurred (Table 2). If needed, the tutor might explicitly note that an error has occurred and give the student the opportunity to locate and repair it (Table 3). If more explicit direction is required, the tutor may direct the student to the erroneous surface feature of the solution, either explicitly or through leading questions (Table 4). The tutor could also describe a principle of the domain that explains why the surface feature of the solution is incorrect (Table 5). In addition, when the student encounters an impasse, the tutor could set a new goal for the student, helping the student overcome it (Table 6). Tables 4, 5, and 6 suggest that computer tutors can assist with some of the same error repair processes as human tutors. Table 7 reveals that computer tutor feedback also may include verbalizing the diagnosis of the misconceptions underlying the student's error.

We see a strong similarity between human and computer tutors. It is clear from these examples and from the discussion of human tutoring studies earlier that human tutors carefully monitor students' reasoning and quickly intervene to make sure that the students' problem solving remains on track. This intervention may be subtle, but it is clearly present. When students veer off the track, the tutors intervene in various ways to lead them back. Our first conclusion from these analyses is that human tutors do perform a type of model tracing, in which they monitor students' problem solving and help guide them back on track when necessary. Computer tutors using model tracing can effectively capture this aspect of how human tutors support students.

The next question then is what should be said when tutors intervene to support students as they conquer an impasse or repair an error. Lepper and Chabay (1988) and Putnam (1987) argued that human tutors do not provide diagnoses—verbalizations of the misconceptions that underlie student errors. Indeed, diagnosis has become a controversial point in the tutoring literature. Much effort has been placed in intelligent tutoring systems research on diagnosis of misconceptions (see Clancey, 1986, for a review). If human tutors can be shown to provide excellent feedback and guidance without doing diagnosis, this may call into question the focus of much intelligent tutoring systems research.

Indeed, it may require more than simply pointing out the student's misconception to produce effective feedback. Sleeman and his colleagues (Sleeman, Kelly, Martinak, Ward, & Moore, 1989; Sleeman, Ward, Kelly, Martinak, & Moore, 1991) investigated the relative effectiveness of simply reteaching an erroneous procedure versus also including an explanation of the reasoning that led to the error. They found that explicitly providing such *model-based remediation* did not improve students' performance more than simply leading the students through the correct procedure over again.

In addition, the examples offered in Tables 1 to 6 are consistent with the view that human tutors do not offer students diagnoses. Instead, they focus students on the erroneous part of the solution and help them to repair it, without communicating any diagnosis of what buggy reasoning the student may have done to lead to the error. In contrast, Table 7 contains examples of computer tutors offering verbalizations of the (presumed) student intentions that led to an error.

What is the role of diagnosis in tutoring? It is important to state clearly what is meant by this term. The question of whether tutors perform and convey diagnoses is a separate issue from whether they monitor students' problem solving and guide them back on track when needed. We have argued that the evidence on this point is clear—human tutors monitor students' solution processes to determine when students have left a good solution path, as do model tracing tutors. The question about diagnosis, then, concerns two additional aspects of guidance. First, do tutors perform diagnosis to determine where and how to focus attention on errors? Without diagnosis, a tutor could simply attempt to guide the student toward whatever correct step should replace the current erroneous one. Alternatively, a tutor might engage in detailed reasoning about the source of a particular error in a solution, and use this analysis to formulate feedback. Second, do tutors communicate diagnoses? Even if tutors do in fact perform diagnoses, it is not necessarily the case that tutors explicitly convey these inferences about the student's reasoning in the feedback.

In our view, human tutors perform diagnosis, but do not communicate their diagnoses to students. It is certainly apparent that a tutor must do more than simply state the student's misconception in order to help the student learn from the error, as the results of Sleeman and his colleagues point out (Sleeman et al., 1989; Sleeman et al., 1991). However, these studies do not address the question of whether tutors perform diagnosis, but rather show that verbalizing the misconception may not be sufficient for effective feedback. Thus, it may not be useful for the tutor to tell the student exactly what the diagnosis was, but the tutor may use this information when deciding what feedback to give to the student.

Indeed, in the examples we have cited earlier in this section, we can see that tutors rarely verbalize their inferences about the student's reasoning. However, in many cases, the focus of the tutors' questions suggests that tutors have indeed analyzed the students' misconceptions, but their strategy is to use the analysis to focus the student on the erroneous situation rather than to communicate the diagnosis itself. For example, in Table 4 the tutor queries the student about the difference between mass and weight, and in Table 6 the tutor reminds the student about LISP predicates and the special atom *t*. In each case, the tutors' questions suggest they have a hypothesis about where the subject may be confused, but the tutors do not verbalize these diagnoses as some computer tutors do.

To summarize, we argue that the first and most essential similarity between human and computer tutors is the support of problem solving through model tracing. The most important aspect of this support is helping students detect and repair errors and overcome impasses. If this process requires understanding the source of the error (such as a confusion between mass and weight), the tutor may use this diagnosis to focus the questions in assisting the recovery. One potentially important difference between computer tutors and human tutorial feedback is that many computer tutors explicitly verbalize this diagnosis and structure more of their feedback around it.

Next, we turn to a second difference between human and computer tutors, the portions of the error recovery process assisted by the tutor. Human tutor feedback tends to include fewer of the components of the recovery process than computer tutor feedback, which often contains explicit verbalizations of the student's misconception. In our examples, this difference was clear since there were no computer tutors in Tables 1 to 4, and virtually no human tutors in Tables 5 to 7. The goal of encouraging the student to tackle as much of the problem solving effort as possible suggests that human tutorial feedback may be superior in this regard to computer tutors.

A third difference between human tutorial feedback and that of computer tutors concerns the flexibility of the level of human tutors' interventions. Human tutors sometimes intervene immediately after an error has occurred, but at other times allow the solution progress to continue, returning to the error later. Thus, human tutors appear to strategically moderate their responses to errors (Littman et al., 1990; Merrill et al., 1992). The flexibility inherent in human tutors' intervention strategies is present in intelligent tutoring systems to only a limited extent. For example, a model-tracing tutor may respond to distracting low-level errors while the student is entering the step rather than waiting for the step to be complete. Thus, a system might query students about probable spelling mistakes or prevent students from entering syntactically illegal expressions (Anderson et al., 1985). Apart from this flexibility, however, the strategy for responding to errors is fixed in most intelligent tutoring systems. For example, the Anderson et al. (1986) Geometry Tutor always intervenes if students make two consecutive incorrect inferences, whereas Bonar and Cunningham's (1988) Bridge tutor waits to offer advice until students request a hint. The more sophisticated strategy of deciding to let some errors pass without comment until the solution is complete while responding to other types of errors during the solution is a goal for further research.

A fourth difference is that human tutors are able to provide feedback of a more subtle nature than computer tutors. Human tutorial sessions are highly interactive, with the student and tutor completing each other's sentences and making use of information channels like pauses, pointing, tone of voice, and so forth. This higher bandwidth of communication allows the human tutor to convey the same amount of information while saying less than would be explicitly verbalized by a computer tutor (Ranney

& Reiser, 1989). Although interesting techniques are being developed for the input and display of information (e.g., Miller, 1988; Shneiderman, 1983), this medium is clearly more limited than the variety of interaction strategies available to human tutors. Fox (1991) argued that the motivational benefits of human tutoring, such as those demonstrated by Lepper and his colleagues (Lepper et al., 1990; Lepper & Chabay, 1988), rely on the highly interactive nature of the communication between tutor and student. Because the student gets the opportunity to complete the tutor's sentences and fill in information for the tutor, the student feels less like a subordinate being guided by an expert and more like a peer. The role of the human tutor in the problem-solving success is generally less apparent than that of a computer tutor. Further research is needed to examine the motivational consequences of these differences in feedback style.

Finally, flexibility in curriculum is another area in which human tutors are more advanced than computer tutors. McArthur et al. (1990) argued that human tutors do not let students flounder on any given task too long; instead, they terminate any task that causes the students significant trouble and immediately set another goal. The next task might be designed to highlight the previous error or to analogize from a previous solution. Regardless of the precise form of the new task, human tutors clearly adapt the curriculum for pedagogical advantage. Some researchers have begun to examine adaptation of curricula in computer tutors. Lesgold (1988) argued that computer tutors must be able to adapt their curricula to the needs and successes of individual students, rather than progressing through a prescribed set of problems and activities. Lesgold proposed encoding curriculum knowledge in computer tutors by associating different instructional methods with goals and subgoals of the instruction. This arrangement allows different instructional goals to be achieved via different pedagogical methods, including didactic instruction, online coaching, and dynamic selection of future problems (Lesgold, Lajoie, Logan, & Eggan, 1990; Woolf & Murray, 1987).

In summary, we suggest that model-tracing tutors can capture a principal feature of the support and guidance of human tutors. Model-tracing feedback focuses students' attention on the parts of solutions that need further work, prevents unnecessary floundering, and provides explanations that facilitate learning from their errors. This support of the students' recovery from impasses appears to underlie the strong pedagogical benefits that have been demonstrated for model-tracing tutors over standard instructional environments. We have also characterized several ways in which current computer tutors appear more rigid and directive than human tutors. As yet, however, there are no empirical investigations of either cognitive or motivational outcomes of these differences in feedback. Such investigations are an important goal for future research.

CONCLUSIONS

In this article, we described how human tutors scaffold students as they recover from impasses, and contrasted this with the behavior of computer tutors. Tutoring can be viewed as a collaborative problem-solving effort, with each party contributing to the solutions. This collaboration is particularly essential at impasses. The task of noticing, locating, and repairing an error is typically a mixture of the tutor's and the student's reasoning efforts.

Human tutors are highly interactive, giving feedback after almost every step, and also giving hints and suggestions upon errors. However, they try to leave as much of the error repair to the students as possible, while still providing as much assistance as necessary. We have argued that a major source of the model-tracing tutor's effectiveness is capturing this type of guidance provided by human tutors. Like human tutors, model-tracing tutors carefully monitor students' problem solving and intervene to keep students from going too far off track and to help students recover from impasses. Both computer and human tutors assist and guide students in several components of the error recovery process, including detecting errors, locating errors, and repairing them.

There are also several differences in the style of feedback that human and computer tutors use to assist in the error recovery process. In general, human tutors manage to assist students while having them do more of the error recovery process. Human tutors tend to offer assistance in flagging or locating an error, in contrast to computer tutors, which typically take on more of the error repair process, and may also convey diagnoses of the student's reasoning that led to the impasse. Human tutors are also more flexible in their level of assistance than computer tutors, and capitalize on a high bandwidth of communication to guide students in a more subtle manner than computer tutors. Together, these differences suggest that the computer tutor's assistance is more noticeable and its control more apparent than human tutorial intervention.

These differences suggest important avenues for future research. First, it is critical to determine whether there are indeed cognitive and motivational consequences of these feedback differences. We would expect students to learn more from fixing their errors when they can play a larger role in their repair. In addition, we would expect the more obvious control asserted by computer tutors to moderate the potential motivational benefits of model tracing guidance. Empirical investigations of these issues would help extend current theories of how tutoring can support learning. Second, such differences should suggest ways in which guidance can be more effectively embedded in computer tutors. It seems that future research should explore ways for computer tutors to enable students to collaborate more in the error recovery process. Although it may not be possible to match the flexibility and subtlety of human tutors with current computer tutoring techniques, our analyses suggest that that model tracing techniques can be profitably employed to achieve substantial cognitive and motivational benefits.

ACKNOWLEDGMENTS

We are indebted to Allan Collins and Stellan Ohlsson for their very helpful comments on this paper, and to Marsha Lovett, Patti Schank, Margaret Recker, Rachel Or-Bach, and other members of the Reasoning Group and the Functions Group of the EMST Program at Berkeley for their helpful comments on an earlier version of this manuscript. The research reported here was supported in part by contracts MDA903-87-K-0652 and MDA903-90-C-0123 from the Army Research Institute, a grant from the Spencer Foundation, and a research grant from the James S. McDonnell Foundation to Princeton University, and by a Faculty Research Grant from the University of California, Berkeley. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, expressed or implied, of these institutions.

REFERENCES

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence, 42,* 7–49.

Anderson, J. R., Boyle, C. F., Farrell, R. G., & Reiser, B. J. (1987). Cognitive principles in the design of computer tutors. In P. Morris (Ed.), *Modelling cognition* (pp. 93–133). New York: John Wiley.

Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science, 228,* 456–462.

Anderson, J. R., Boyle, C. F., & Yost, G. (1986). The Geometry Tutor. *Journal of Mathematical Behavior, 5,* 5–19.

Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science, 13,* 467–505.

Anderson, J. R., & Reiser, B. J. (1985). The LISP tutor. *Byte, 10,* 159–175.

Anzai, Y., & Simon, H. A. (1979). The theory of learning by doing. *Psychological Review, 86,* 124–140.

Ausubel, D. P. (1963). *The psychology of meaningful verbal learning: An introduction to school learning.* New York: Grune & Stratton.

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

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher, 13,* 4–16.

Bonar, J. G., & Cunningham, R. (1988). Bridge: Tutoring the programming process. In J. Psotka, L. D. Massey, & S. A. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned* (pp. 409-434). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Brown, J. S., & Burton, R. R. (1978). Diagnosing bugs in a simple procedural skill. *Cognitive Science, 2,* 155-192.

Brown, J. S., Burton, R. R., & de Kleer, J. (1982). Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 227-282). London: Academic.

Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review, 31,* 21-32.

Burton, R. R. (1982). Diagnosing bugs in a simple procedural skill. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 157-183). London: Academic.

Burton, R. R., & Brown, J. S. (1976). A tutoring and student modeling paradigm for gaming environments. *ACM SIGCSE Bulletin, 8*(1), 236-246.

Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 79-98). London: Academic.

Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-aided instruction. *IEEE Transactions on Man-Machine Systems, 11,* 190-202.

Carroll, J. M., & Carrithers, C. (1984). Blocking learner error states in a training wheels system. *Human Factors, 26,* 377-389.

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.

Clancey, W. J. (1986). Qualitative student models. In J. F. Traub (Ed.), *Annual Review of Computer Science* (Vol. 1, pp. 381-450). Palo Alto, CA: Annual Reviews, Inc.

Clancey, W. J. (1987). *Knowledge-based tutoring: The Guidon program.* Cambridge, MA: MIT Press.

Cohen, P. A., Kulik, J. A., & Kulik, C.-L. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal, 19,* 237-248.

Collins, A., & Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 1-18). New York: Springer-Verlag.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Collins, A., & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.), *Advances in instructional psychology* (Vol. 2, pp. 65-119). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Collins, A., Warnock, E. H., & Passafiume, J. J. (1975). Analysis and synthesis of tutorial dialogues. In G. H. Bower (Ed.), *The psychology of learning and motivation* (pp. 49-87). New York: Academic.

Corbett, A., & Anderson, J. R. (1991, April). *Feedback control and learning to program with the CMU Lisp Tutor.* Paper presented at the annual meeting of the American Educational Research Association, Chicago.

Corbett, A. T., Anderson, J. R., & Patterson, E. G. (1990). Student modeling and tutoring flexibility in the Lisp Intelligent Tutoring System. In C. Frasson & G. Gauthier (Eds.), *Intelligent tutoring systems: At the crossroads of artificial intelligence and education* (pp. 83-106). Norwood, NJ: Ablex.

Davis, R. B. (1966). Discovery in the teaching of mathematics. In L. S. Shulman & E. R. Keislar (Eds.), *Learning by discovery: A critical appraisal* (pp. 114-128). Chicago: Rand McNally.

Fox, B. A. (1991). Cognitive and interactional aspects of correction in tutoring. In P. Goodyear (Ed.), *Teaching knowledge and intelligent tutoring* (pp. 149–172). Norwood, NJ: Ablex.

Goldstein, I. P. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 51–77). London: Academic.

Johnson, W. L. (1986). *Intention-based diagnosis of errors in novice programs.* Palo Alto, CA: Kaufmann.

Kimball, R. (1982). A self-improving tutor for symbolic integration. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 283–307). London: Academic.

Kulhavy, R. W. (1977). Feedback in written instruction. *Review of Educational Research, 47,* 211–232.

Kulik, J. A., & Kulik, C.-L. C. (1988). Timing of feedback and verbal learning. *Review of Educational Research, 58,* 79–97.

Leinhardt, G., & Ohlsson, S. (1990). Tutorials on the structure of tutoring from teachers. *Journal of Artificial Intelligence in Education, 2,* 21–46.

Lepper, M. R., Aspinwall, L., Mumme, D., & Chabay, R. W. (1990). Self-perception and social perception processes in tutoring: Subtle social control strategies of expert tutors. In J. M. Olson & M. P. Zanna (Eds.), *Self inference processes: The sixth Ontario symposium in social psychology* (pp. 217–237). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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-Verlag.

Lesgold, A. (1988). Toward a theory of curriculum for use in designing intelligent instructional systems. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 114–137). New York: Springer-Verlag.

Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1991). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In J. H. Larkin & R. W. Chabay (Eds.), *Computer assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches* (pp. 201–238). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Lesgold, A., Lajoie, S., Logan, D., & Eggan, G. (1990). Applying cognitive task analysis and research methods to assessment. In N. Frederiksen, R. Glaser, A. Lesgold, & M. G. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition* (pp. 325–350). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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

Littman, D. (1991). Tutorial planning schemas. In P. Goodyear (Ed.), *Teaching knowledge and intelligent tutoring* (pp. 107–122). Norwood, NJ: Ablex.

Littman, D., Pinto, J., & Soloway, E. (1990). The knowledge required for tutorial planning: An empirical analysis. *Interactive Learning Environments, 1,* 124–151.

McArthur, D., Stasz, C., & Zmuidzinas, M. (1990). Tutoring techniques in algebra. *Cognition and Instruction, 7,* 197–244.

McKendree, J. (1990). Effective feedback content for tutoring complex skills. *Human-Computer Interaction, 5,* 381–413.

Merrill, D. C., Reiser, B. J., Beekelaar, R., & Hamid, A. (1992). Making processes visible: Scaffolding learning with reasoning-congruent representations. In *Proceedings of ITS-92: The Second International Conference on Intelligent Tutoring Systems.* Montreal: Springer-Verlag.

Merrill, D. C., Reiser, B. J., & Landes, S. (1992, April). *Human tutoring: Pedagogical strategies and learning outcomes.* Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

Miller, J. R. (1988). The role of human-computer interaction in intelligent tutoring systems. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 143–190). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Milson, R., Lewis, M. W., & Anderson, J. R. (1990). The teacher's apprentice: Building an algebra tutor. In R. Freedle (Ed.), *Artificial intelligence and the future of testing* (pp. 53–71). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Ohlsson, S., & Langley, P. (1988). Psychological evaluation of path hypotheses in cognitive diagnosis. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 42–62). New York: Springer-Verlag.

Ohlsson, S., & Rees, E. (1991). The function of conceptual understanding in the learning of arithmetic procedures. *Cognition and Instruction, 8,* 103–179.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic.

Putnam, R. T. (1987). Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal, 24,* 13–48.

Ranney, M., & Reiser, B. J. (1989). Reasoning and explanation in an intelligent tutoring system for programming. In G. Salvendy & M. J. Smith (Eds.), *Designing and using human-computer interfaces and knowledge based systems: Proceedings of the Third International Conference on Human-Computer Interaction* (pp. 88–95). New York: Elsevier.

Reiser, B. J., Anderson, J. R., & Farrell, R. G. (1985). Dynamic student modeling in an intelligent tutor for LISP programming. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 8–14). Los Angeles: Kaufmann.

Reiser, B. J., Beekelaar, R., Tyle, A., & Merrill, D. C. (1991). GIL: Scaffolding learning to program with reasoning-congruent representations. In *The International Conference of the Learning Sciences: Proceedings of the 1991 conference* (pp. 382–388). Evanston, IL: Association for the Advancement of Computing in Education.

Reiser, B. J., Connelly, J. W., Ranney, M., & Ritter, C. (1992). *The role of explanatory feedback in skill acquisition.* Unpublished Manuscript, Princeton University, Princeton, NJ.

Reiser, B. J., Copen, W. A., Ranney, M., Hamid, A., & Kimberg, D. Y. (1991). *Cognitive and motivational consequences of tutoring and discovery learning.* Unpublished Manuscript, Princeton University, Princeton, NJ.

Reiser, B. J., Friedmann, P., Kimberg, D. Y., & Ranney, M. (1988). Constructing explanations from problem solving rules to guide the planning of programs. In *Proceedings of ITS-88: The International Conference on Intelligent Tutoring Systems* (pp. 222–229). Montreal.

Reiser, B. J., Kimberg, D. Y., Lovett, M. C., & Ranney, M. (1992). Knowledge representation and explanation in GIL, an intelligent tutor for programming. In J. H. Larkin & R. W. Chabay (Eds.), *Computer-assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches* (pp. 111–149). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Reiser, B. J., Ranney, M., Lovett, M. C., & Kimberg, D. Y. (1989). Facilitating students' reasoning with causal explanations and visual representations. In D. Bierman, J. Breuker, & J. Sandberg (Eds.), *Proceedings of the Fourth International Conference on Artificial Intelligence and Education* (pp. 228–235). Springfield, VA: IOS.

Ridgway, J. (1988). Of course ICAI is impossible . . . worse though, it might be seditious. In J. Self (Ed.), *Artificial intelligence and human learning: Intelligent computer-aided instruction* (pp. 28–48). London: Chapman and Hall.

Scardamalia, M., Bereiter, C., McLean, R. S., Swallow, J., & Woodruff, E. (1989). Computer-supported intentional learning environments. *Journal of Educational Computing Research, 5,* 51–68.

Schank, R. C., & Farrell, R. (1987). Creativity in education: A standard for computer-based teaching. *Machine-Mediated Learning, 2,* 175–194.

Schoenfeld, A. H. (1988). Mathematics, technology, and higher order thinking. In R. S. Nickerson & P. P. Zodhiates (Eds.), *Technology in education: Looking toward 2020* (pp. 67–96). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Schofield, J. W., Evans-Rhodes, D., & Huber, B. R. (1990). Artificial intelligence in the classroom: The impact of a computer-based tutor on teachers and students. *Social Science Computer Review, 8,* 24–41.

Schwartz, J. L. (1989). Intellectual mirrors: A step in the direction of making schools knowledge places. *Harvard Educational Review, 59,* 51–61.

Shneiderman, B. (1983). Direction manipulation: A step beyond programming languages. *IEEE Computer, 16,* 57–69.

Shute, V. J., Glaser, R., & Raghavan, K. (1989). Inference and discovery in an exploratory laboratory. In P. L. Ackerman, R. J. Sternberg, & R. Glaser (Eds.), *Learning and individual differences* (pp. 279–326). New York: Freeman.

Singley, M. K. (1990). The reification of goal structures in a calculus tutor: Effects on problem solving performance. *Interactive Learning Environments, 1,* 102–123.

Skinner, B. F. (1968). *The technology of teaching.* New York: Appleton-Century-Crofts.

Sleeman, D. (1982). Assessing aspects of competence in basic algebra. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 185–199). London: Academic.

Sleeman, D., Kelley, A. E., Martinak, R., Ward, R. D., & Moore, J. L. (1989). Studies of diagnosis and remediation with high school algebra students. *Cognitive Science, 13,* 551–568.

Sleeman, D., Ward, R. D., Kelly, E., Martinak, R., & Moore, J. (1991). Overview of recent studies with PIXIE. In P. Goodyear (Ed.), *Teaching knowledge and intelligent tutoring* (pp. 173–185). Norwood, NJ: Ablex.

Snow, R. E., & Lohman, D. F. (1984). Toward a theory of cognitive aptitude for learning from instruction. *Journal of Educational Psychology, 76,* 347–376.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12,* 257–285.

VanLehn, K. (1988). Toward a theory of impasse-driven learning. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 19–41). New York: Springer-Verlag.

VanLehn, K. (1990). *Mind bugs: The origins of procedural misconceptions.* Cambridge, MA: MIT Press.

Woolf, B., & Murray, T. (1987). A framework for representing tutorial discourse. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp. 189–192). Milan, Italy: Kaufmann.