An Empirical Assessment of Comprehension Fostering Features in an Intelligent Tutoring System

Santosh A. Mathan and Kenneth R. Koedinger

Center for Innovation in Learning / Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
mathan@andrew.cmu.edu, koedinger@cmu.edu

Abstract. This paper describes the design and evaluation of two features in an Intelligent Tutoring System designed to facilitate a deeper conceptual understanding of domain principles in conjunction with the development of procedural skills. The first feature described here relates to the timing of feedback. Some researchers have argued that immediate corrective feedback, as embodied in many cognitive tutors, can block the exercise of activities that may enable students to gain a deeper conceptual understanding of a domain. These include self-monitoring, error detection, and error correction skills. We compare an immediate feedback tutor with a tutor that allows students to reflect on problem solving outcomes, and engage in error detection and correction activities. The other feature reported here is a component of declarative instruction. We assess the use of Ex-ample Walkthroughs as a comprehension-fostering tool. Prior to procedural practice, Example Walkthroughs step students through the study of example problems and guide them to reflect on the reasoning involved in going from a problem statement to a solution. An evaluation has shown that the best learning outcomes were associated with a combination of immediate feedback and Example Walkthroughs. There are indications that a combination of lower cognitive load during procedural practice and a robust and accurate encoding of declarative concepts contributed to the observed outcomes.

1 Introduction

The research reported in this paper seeks to facilitate the joint development of procedural and conceptual knowledge. We focus on features of declarative instruction and problem solving feedback in an Intelligent Tutoring System designed toward this end. We begin with a description of the theoretical motivations underlying our design decisions. Later in the paper we describe how these ideas were implemented in the context of a tutor for Microsoft Excel and present results from an empirical evaluation of their efficacy.

1.1 Feedback: When?

Some of the most widely used intelligent tutoring systems provide immediate feedback on errors [6]. Empirical findings suggest that skill acquisition is most efficient with immediate feedback. For instance, Corbett and Anderson [5] compared the pedagogical benefits of immediate and delayed feedback in the context of their LISP tutor.

S.A. Cerri, G. Gouardères, and F. Paraguaçu (Eds.): ITS 2002, LNCS 2363, pp. 330–343, 2002.

While their comparison did not reveal statistically significant differences in terms of posttest performance, they did see reliable differences in the learning rate. Students in the immediate feedback condition completed training significantly faster. Immediate feedback served to minimize floundering and keep the learning process efficient.

Despite the fact that tutors based on such an approach have been very successful in classroom contexts [6], the principle of immediate feedback has been criticized on at least two grounds. First, critics point out that immediate feedback offered by cognitive tutors is qualitatively different from that offered by human tutors. For instance, Merrill et al. [10] found that human tutors do not intervene immediately on errors that may provide learning opportunities. Instead, they often guide learners through error detection and correction activities. Second, immediate feedback has been criticized on the basis of empirical studies that highlight benefits of delayed feedback. For instance, in a study involving a genetics tutor [9], students either received feedback as soon as an error was detected or at the end of a problem. As in Corbett and Anderson [5], students who received immediate feedback completed training problems significantly faster during training and performed equally well on near transfer tasks. However, students who received delayed feedback performed significantly better on a far transfer task. Similar comparisons in other domains (LISP [17]; Motor Learning [16]) show that while performance differences may not be apparent during or immediately following training, students trained in delayed feedback conditions may show better retention of skills over time.

The Guidance Hypothesis

The guidance hypothesis proposed by Schmidt et al. [15] provides an account of the suggested trade-off between the benefits offered by immediate feedback and those offered by delayed feedback. According to the guidance hypothesis, feedback serves to precisely direct learner actions following each presentation. This has the effect of boosting performance during and immediately following training. However, feedback can negatively impact learning in two ways. First, feedback could obscure important task cues – that is, learners may come to depend on feedback instead of cues inherent in the natural task environment. Second, feedback may prevent important skill components of a task from being exercised. In many academic tasks, these skills could include error detection & correction, and metacognitive skills.

The guidance hypothesis suggests that immediate feedback may promote the development of *generative skills* – that is, skills involved in selection and implementation of operators in specific task contexts. However, *evaluative skills* – skills called for in evaluating the effect of applying these operators, implementing steps to remedy errors, and monitoring one's own cognitive process may go unpracticed. These evaluative functions are instead delegated to feedback. As a consequence, performance may be compromised in transfer and retention tasks where the likelihood of errors is high and both generative and evaluative skills must be jointly exercised. Additionally, the exercise of evaluative skills may provide an opportunity for a deeper conceptual understanding. As Merrill et al. [10] have suggested, errors provide an opportunity to develop a better model of the behavior of operators in a domain.

The Designers Dilemma

The research on feedback just summarized presents the designer with a dilemma. Immediate feedback speeds up the learning process. Furthermore, some of the most effective and efficient cognitive tutors are based on this scheme [6]. However, a designer may also wish to realize benefits such as the development of debugging and metacognitive skills offered by delayed feedback. Unfortunately, the research reviewed here offers little guidance as to what an appropriate level of delay might be. At best, an inappropriate level of delay can reduce the efficiency of the learning process. At worst, delayed feedback can recede to a no-feedback condition. Unproductive floundering and frustration may characterize the learning process in such circumstances.

An Integrative Perspective

The mutually exclusive choice just described stems partly from the fact that much of the debate concerning when to provide feedback is cast in terms of feedback latency. We suggest that a more appropriate focus is on the underlying model of desired performance that serves as the basis for providing feedback to students.

Expert Model

Currently feedback in cognitive tutors is based on what is broadly referred to as an *Expert Model*. Such a model characterizes the end-goal of the instructional process as error-free task execution. Feedback is structured so as to lead students towards such performance. An Expert Model based tutor focuses on the generative components of a skill. Figure 1 (L) illustrates the student interaction with an Expert Model tutor.

Intelligent Novice Model

An alternative model that could serve as the basis for feedback in cognitive tutors is that of an *Intelligent Novice*. The assumption underlying such a model is that an intelligent novice, while progressively getting skillful, is likely to make errors. Recognizing this possibility, the Intelligent Novice model incorporates both self-monitoring, 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. Immediate feedback with respect to such a model would resemble delayed feedback with respect to an Expert Model. Figure 1 (R) outlines student interaction with a tutor based on an Intelligent Novice model.

Later in this paper we will detail the design of two versions of an Excel Tutor – one based on an Expert model the other on an Intelligent Novice model. We will also present results of a study evaluating learning outcomes associated with each. However, before we do so, we describe the theoretical motivations underlying the design of declarative instruction that precedes procedural practice with the two models.

1.2 Declarative Instruction

Declarative knowledge plays a crucial role in early skill acquisition. Under the ACT-R theory of skill acquisition [1], declarative knowledge serves to structure initial problem solving attempts. Over the course of practice, knowledge compilation processes transform declarative encodings into efficient, context specific production rules. Besides playing a guiding role in the initial stages of skills acquisition, declarative knowledge of principles underlying a domain can provide the basis for transfer of skills to novel task domains [18].

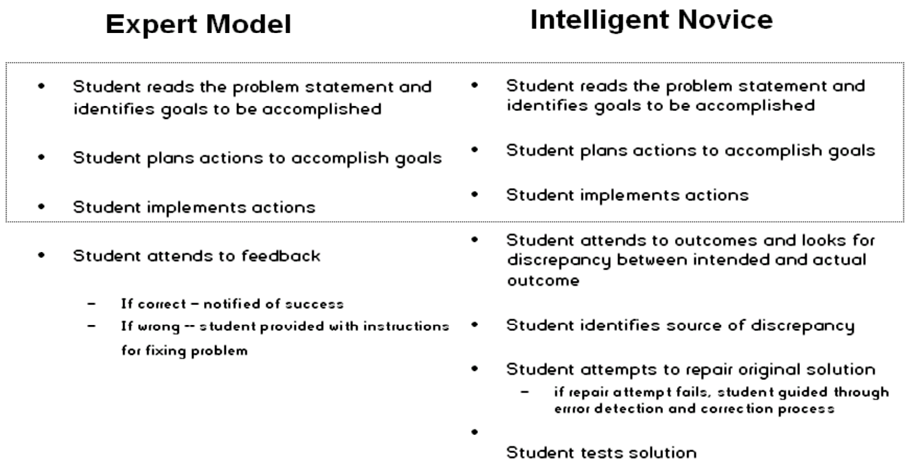

Fig. 1. Interaction with an Expert Model Tutor (L) and Intelligent Novice Model (R)

The study of examples has been used as a tool for fostering the development of conceptual understanding [3]. Examples serve to introduce learners to the range of operators relevant to the solution of a class of problems, the specific conditions under which these operators apply, the transformations that result from the application of operators in specific problem contexts, and the overall sequence in which these operators are applicable. Recent research by Renkl, Atkinson, and Maier indicates that the effectiveness of examples can be enhanced by integrating elements of problem solving into the study of examples [14]. That is, students who study fully worked out examples, then complete intermediate steps in partially incomplete examples before problem solving, outperform students who transition directly to problem solving from the study of fully worked out examples. As Renkl et al. have suggested, elements of such an approach – that is, the progression from modeling of solutions with examples, to fading of scaffolds to independent problem solving – can be found in a variety of successful instructional techniques. These include: Reciprocal Teaching [12], and Cognitive Apprenticeship [4]. Jones and Fleishman [8] have theorized – on the basis of a CASCADE model of fading examples – that partially worked out examples focus attention on crucial parts of a problem, thus providing an opportunity for self-explanation. Furthermore, as a consequence of making problem-solving decisions at these points, students acquire search control knowledge (knowledge of the sub-goal structure for solving the task).

Declarative Instruction in the tutor described here incorporates what we call *Example Walkthroughs* to guide students in the study of examples. Students read textual expositions of concepts and watch video illustrations of the application of these concepts in the context of examples. Subsequently, instead of progressing directly into problem solving, students solve the examples demonstrated in the video with the help of Example Walkthroughs. These walkthroughs step students through the reasoning necessary to solve the example problems. At each step of the solution process, students are prompted with questions that serve to help them make the inferences necessary to perform the task correctly. Incorrect inferences, which may result from an inaccurate or partial encoding of relevant declarative knowledge, are remedied with brief messages that clarify the knowledge necessary to make the appropriate inference.

Example Walkthroughs differ from conventional approaches to declarative instruction in several ways. First, declarative information is typically presented in a passive form (usually in the form of text, lecture, or video expositions). In contrast, walkthroughs actively engage students in elaborating on information presented. Secondly, walkthroughs provide an opportunity to check and correct knowledge encoding in the context of representative tasks. Thirdly, walkthroughs allow conceptual gaps to be remedied immediately following the exposition of a concept, instead of being deferred to problem solving contexts where cognitive load may be high.

We now describe implementation of feedback and declarative instruction based on the analysis just presented. We do so in the context an Excel Tutor.

2 Excel Tutor

Spreadsheets have been widely regarded as exemplary end-user programming environments [11]. They allow non-programmers 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. [7]). A commonly reported usability problem concerns the appropriate use of absolute and relative references – these are schemes that allow users to perform iterative computations. These difficulties exist despite an abundance of manufacturer and third-party training materials. The tutor reported in this paper was designed to enable students to master cell referencing concepts. We elaborate on the tutorial domain below and go on to detail features of the tutor based on the theoretical analysis presented earlier.

2.1 Overview of Tutorial Domain

A spreadsheet is essentially a collection of cells on a two dimensional grid. Individual cells may be addressed by their column and row indices. Column indices (also called column references) are denoted by number, whereas row indices (often called row references) are denoted by letter. Cells may contain alphanumeric data and formulas. Formulas can refer to values in specific cells by referring to their addresses. So, a user could specify a formula in cell C3 (in column C and row 3) that adds the contents of cell A3 and B3 by entering: '=A3+B3'.

Formulas may be reused to perform iterative operations. This is accomplished through a scheme called relative referencing. Consider the spreadsheet depicted in Figure 2. 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, Excel modifies the formula 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 3 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.

In order to determine the appropriate relative references at new locations, Excel updates formulas based on where the formula is moved. When a formula is moved into a cell in a different column, Excel updates column references in the formula by the number of columns moved (see Figure 2, =B2+B3+B4 becomes =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 get updated by the number of rows moved (see Figure 2, =B2+C2+D2 becomes =B4+C4+D4 when moved across rows from E2 to E4).

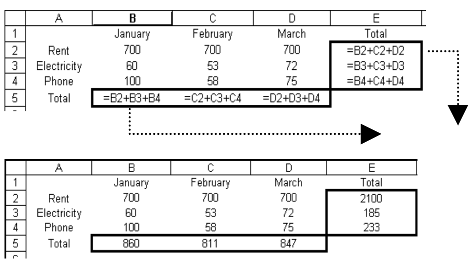

Fig. 2. Relative references allow formulas in B5 and E2 to be reused

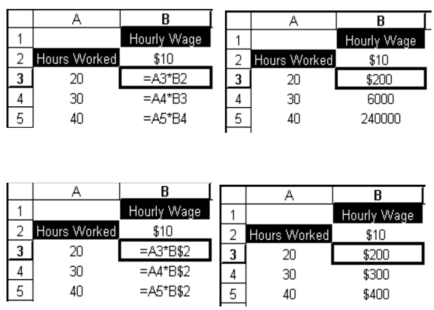

Fig. 3. Incorrect use of relative refs (top) remedied absolute refs (bottom)

While 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 3. 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 in order 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). In order 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, in order for the formula in B3 to work appropriately when copied and pasted, it would be modified to read =A3*B$2.

2.2 Expert Model Tutor Description

As mentioned earlier the Expert Model emphasizes generative skills. Details of the design of declarative instruction and feedback design based on such a model are presented below.

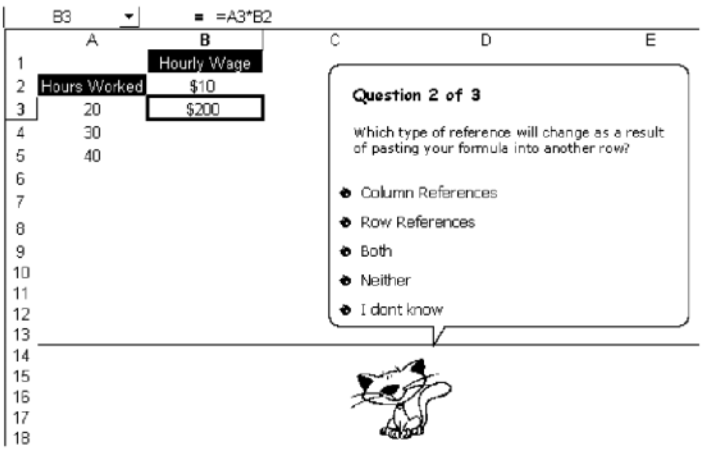

Fig. 4. Screenshot from Example Walkthrough for Expert Model Tutor Example Walkthrough

Example Walkthroughs corresponding to the Expert Model tutor focus on generative skills. Students are provided with a 3-step procedure, described below, in order to generate solutions to cell-referencing problems. As mentioned earlier, in order to determine where an absolute reference may be needed, users have to be able to identify the references in a formula that will change as a result of copying and pasting. Depending on where a formula will be pasted, row and/or column references will change. Each reference that will change must be inspected. Of these, references changes that are to be prevented must be preceded by a '$' symbol – an absolute reference. The Expert Model Walkthrough guides students through these inferences by posing a series of questions (Figure 4): [Which way will you be pasting your formula? (into another column/row/both?) , Which type of reference will change when moved? (column/row/both?) Of the references that will change, which ones should you prevent?]. Students respond to these questions by picking from multiple-choice options. The system provides succinct explanations in response to errors.

Problem Solving Feedback

During problem solving, students working with an Expert Model based tutor receive feedback as soon as an error is detected. 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 correct answer. If help is sought, the student is guided through the process of identifying where, if any, absolute references are required in a particular problem context. Students are interactively guided by question prompts to solve the problem deductively (see Figure 4).

2.3 Intelligent Novice Tutor Description

In addition to generative skills emphasized by Expert Model based tutors, the Intelligent Novice Model provides practice in evaluative skills.

Example Walkthrough

In addition to helping students solve example problems, the Intelligent Novice Example Walkthrough guides students through the reasoning associated with the exercise of evaluative skills. First, students are prompted to indicate the values and formulas they should see in each cell if the formula works correctly. Subsequently, students copy and paste a formula without any absolute references into each cell of the example. Students, then note the values and formulas produced as a result of copying the original formula. They are prompted to examine discrepancies between the actual and intended formulas. Prompts then guide learners to use the identified discrepancies to determine where an absolute reference may be necessary. Figure 5 illustrates a portion of the Intelligent Novice Example Walkthrough.

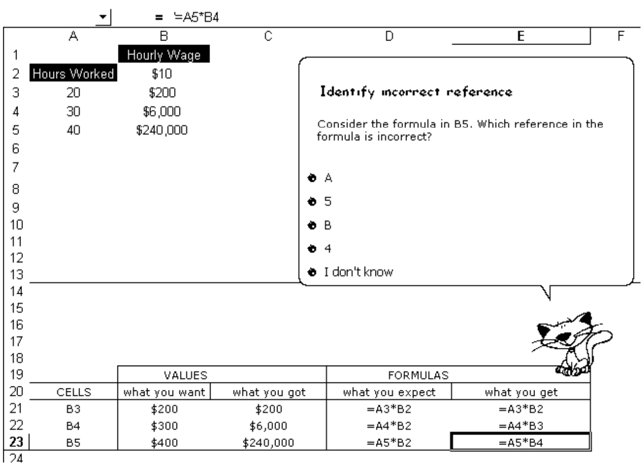

Fig. 5. Screenshots from Example Walkthrough for Intelligent Novice Model Tutor

Problem Solving Feedback

In contrast to the Expert Model based tutor, the Intelligent Novice Model allows students to enter an incorrect formula and copy and paste it to observe the consequences of the error. The student is given an opportunity to detect the source of the error and correct the formula. Hints requested by students can guide them through the error detection and correction process. An error in the formula correction step will result in immediate corrective feedback to minimize floundering and frustration. If a student fails to detect an error and tries to start a new problem, feedback directs the student to check for errors and request hints if needed. 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 help is sought, the student is asked to specify formulas and values that should result if the original formula were referenced appropriately. This is noted in a table on the spreadsheet. Subsequently, the student is asked to enter a formula without any absolute references and copy and paste it. The student then prompted to note the values and formulas produced. The student is prompted to use the discrepancy between actual and intended formulas to determine where absolute references, if any, may be appropriate (see Figure 5).

3 Method

An evaluation of the features described here was conducted with a group of 40 participants recruited from a local temporary employment agency. All subjects had general computer experience, including proficiency with word processing, email, and web applications – however, they were all spreadsheet novices. We randomly assigned students to one of four conditions associated with the manipulation of two factors: Model – Expert or Intelligent Novice (EX, IN), Declarative Instruction – With or Without Example Walkthrough (WT, noWT).

The evaluation was conducted over the course of three days. On *Day-1*, students came in for a 90-minute instructional session. Declarative instruction provided all students with an exposition of basic spreadsheet concepts: everything from data entry, copying and pasting to formula creation and cell referencing. Cell referencing lessons for all students included video examples of cell referencing problems being solved. Students in the Walkthrough conditions stepped through Example Walkthroughs immediately following the videos, prior to problem solving. Students in the No Walkthrough conditions went directly to problem solving. Declarative instruction took approximately 60 minutes for students whose instruction included walkthroughs, and 50 minutes for those whose instruction did not. 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 pre-test and was followed by a post-test. On *Day-2*, students came in the next day for 50 minutes of procedural practice with the tutor. A post-test was administered following the instructional session. On *Day-3*, eight days after Day-2, students came in for a third instructional session. Students attempted a pre-test and transfer task to measure retention prior to the instructional session. The third session consisted of 30 minutes of procedural practice and was followed by a post test

The pre and post-tests had two components: a test of problem solving and a test of conceptual understanding. The problem-solving test consisted of problems isomorphic to training tasks. 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. Students also were also asked to complete a computer experience questionnaire. The questionnaire asked them to indicate the frequency with which they use various computer applications and rate their proficiency at each.

4 Results

An analysis of pre-test scores showed student performance in all conditions to be close to zero. However, we suspected that computer proficiency might influence learning outcomes. The computer experience questionnaire provided the basis to assign a computer experience score to each participant. Analysis has shown the computer experience score to be a significant predictor of student performance ($F=8.57$, $p < 0.007$). The results reported in this paper control for computer experience as a covariate.

As shown in Figure 6A, a repeated measure ANCOVA, over all the tests, did not show a significant main effect for Model or Walkthrough. However, the analysis did reveal a significant Model-Walkthrough interaction ($F= 5.10$, $p < 0.03$). Overall, students in the Expert-Walkthrough condition outperformed students in all other conditions (Figure 6A). Though not shown in Figure 6, a similar pattern of scores was observed in the conceptual ($F=7.12$, $p < 0.02$) and problem-solving tests ($F= 7.4$, $p < 0.01$).

As shown in Figure 6B, students in the Expert-Walkthrough condition demonstrated the greatest immediate learning as measured by the Day-1 and Day-2 post-tests, and the most robust retention of material. A similar pattern was observed when the problem solving and conceptual understanding scores were examined separately. A Model-Walkthrough interaction on the transfer task suggests that students in the Expert-Walkthrough condition also demonstrated the highest performance on the transfer test ($F=4.49$, $p<0.05$) (Figure 6C). There is also some indication of an aptitude treatment interaction (Figure 6D). While High computer experience students performed at about the same level on the transfer task regardless of condition, low computer experience students got a performance boost in the Expert-Walkthrough condition ($F = 3.56$, $p < 0.07$).

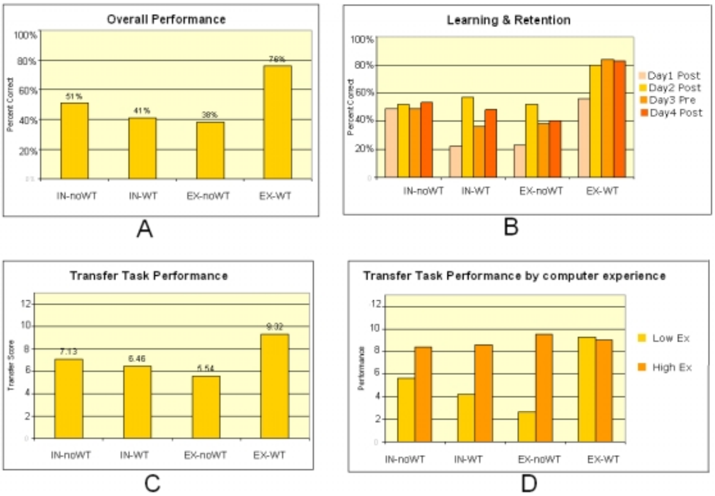

Fig. 6. Evaluation Results

We observed qualitative differences in the way students in each condition dealt with errors. Students in the Expert-Walkthrough condition were able to understand the error messages, repair their solutions, and get back on track efficiently. In contrast, several students in the Expert-noWalkthrough condition were unable to fully comprehend terms and concepts used in the error correction dialogs – several students had forgotten or expressed confusion about concepts described during declarative training. They tended to get to the solution by trial and error attempts at placing absolute references. Students in the Intelligent Novice conditions experienced the greatest frustration. The error analysis and fixing process appeared to become a fairly lengthy and involved problem-solving episode in itself – this frustration was particularly pronounced among low computer experience students.

5 Discussion

Contrary to our expectations, our evaluation did not reveal a main effect for Model or Example Walkthrough. Instead, a conjunction of features associated with Expert Model based feedback and Example Walkthroughs had the greatest impact on learning, retention and transfer outcomes. We examine features of declarative and procedural instruction associated with the Expert-Walkthrough condition below:

Explicit Procedure to Guide Problem Solving

Students in the Expert-Walkthrough condition had the benefit of a three-step procedure (expressed in the form of the three questions) to guide their problem solving efforts. Students were introduced to this procedure during Example Walkthroughs. Furthermore Expert Model based feedback during procedural practice kept students focused on applying these rules to solve problems. Prior research suggests that a procedure for interpreting declarative concepts in problem solving contexts contributes to better learning outcomes [13]. The Intelligent Novice Walkthrough on the other hand focused on imparting an understanding of the mechanism underlying cell referencing. Students had to generate a procedure based on their understanding of underlying concepts.

Cognitive Load under Intelligent Novice Model

Procedural practice with the Intelligent Novice model was more taxing on working memory than with the Expert Model tutor for at least two reasons [cf. 2]. First, error diagnosis and recovery steps under the Intelligent Novice condition often became extended problem-solving episodes in their own right. These episodes are likely to have interfered with the acquisition of solution generation schemas. Second, artifacts of the interface may have imposed additional cognitive load on learners. The error recovery steps required students to split attention between 3 areas: the problem, the table used to track expected and actual values and formulas, and messages from the office assistant (see Figure 5). These two features were also inherent in the Intelligent Novice Example Walkthroughs, potentially compromising their efficacy.

Accuracy and Robustness of Declarative Encodings

Expert-Walkthrough condition students are likely to have benefited from comprehension checks and the opportunity to elaborate on video examples during declarative instruction. There are indications that Expert Model students whose declarative instruction included walkthroughs had a more robust and accurate encoding to guide them during procedural practice. Students in the Expert-Walkthrough condition made half as many errors as those in the Expert-noWalkthrough condition on the first six problems – these problems represented the first presentation of the six types of problems included in the tutor (1.01 errors per problem vs. 2.79, $F=3.09$, $p < 0.09$).

6 Conclusion

Empirical results suggest that the best learning outcomes were associated with the combined use of Expert Model based feedback and Example Walkthroughs. Overall, students in the Expert-Walkthrough condition exhibited the strongest performance in transfer tests, tests of conceptual understanding, and on problem solving tasks isomorphic to those encountered during training. Furthermore, students in the Expert-Walkthrough condition exhibited robust retention of learning over the course of an eight-day retention interval. We have suggested that a combination of relatively low cognitive load during practice, the provision of an explicit procedure for applying declarative knowledge, and a robust and accurate declarative encoding contributed to observed outcomes.

The research reported here has implications for the design of cognitive tutors. First, Example Walkthroughs show the potential for boosting learning outcomes associated with cognitive tutors. Walkthroughs provide an opportunity for students to elaborate on video and textual expositions of examples. Furthermore they provide a way to check and remedy student comprehension of concepts prior to procedural practice. Expert-Walkthrough students are likely to have benefited from an explicit procedure (in the form of 3 questions) for applying declarative knowledge in problem contexts. Additionally, students were guided through the process of applying the procedure in the context of examples. Second, the exercise of evaluative skill, as afforded by the Intelligent Novice tutor, comes at a cost – if these activities become extended problem solving episodes, they have the potential for disrupting the acquisition of schemas [cf. 2] associated with generative skills.

References

1. Anderson, J. R. (1993). Rules of the mind. Hillsdale, N.J: LEA
2. Chandler, P. & Sweller, J. (1991). Cognitive load theory and the format of instruction. Cognition and Instruction, 8, 293-332.
3. 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
4. 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 (453-494). Hillsdale, NJ: LEA
5. 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 2001, ACM, 2001 245-252
6. Corbett, A. T., Koedinger, K. R., & Hadley, W. (2001). Cognitive Tutors: From the research classroom to all classrooms. In Goodman, P. (Ed) Technology Enhanced Learning (235-263). Mahwah, NJ: LEA
7. Hendry, D. G. and Green, T. R. G. (1994) Creating, comprehending, and explaining spreadsheets: a cognitive interpretation of what discretionary users think of the spreadsheet model. Int. J. Human-Computer Studies, 40(6), 1033-1065
8. Jones, R. M., Fleischman, E. S. (2001). Cascade Explains and Informs the Utility of Fading Examples to Problems. Proceedings of the 23rd Conf. of the Cognitive Sc. Society (pp. 459-464). Mahwah, NJ: LEA
9. Lee, A.Y. (1992) Using tutoring systems to study learning. Behavior Research Methods, Instruments, & Computers, 24(2, 205-212)
10. Merrill, D.C., Reiser, B.J., Merrill, S.K., and Landes, S. "Tutoring: Guided Learning by Doing". Cognition and Instruction, 1995, 13(3). 315-372.
11. Nardi, B. (1993). A Small Matter of Programming. Cambridge: MIT Press
12. Palincsar, A.S., & Brown, A.L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. Cognition and Instruction, 1, 117-175
13. Reif, F. & Allen, S. (1992). Cognition for interpreting scientific concepts: A study of acceleration. Cognition and Instruction, 9, 1-44
14. Renkl, A., Atkinson, R. K., & Maier, U. H. (2000). From studying examples to solving problems: Fading worked-out solution steps helps learning. In L. R. Gleitman & A. K. Joshi (Eds.), Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society, 393-398. Mahwah, NJ: LEA.

15. Schmidt, R.A., Young, D.E., Swinnen, S., & Shapiro, D.C. (1989). Summary knowledge of results for skill acquisition: Support for the guidance hypothesis. *J. of Experimental Psychology: Learning, Memory, and Cognition, 15*, 352-359
16. 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
17. Schooler, L. J. & Anderson, J. R. (1990). The disruptive potential of immediate feedback. Proceedings of the Twelfth Annual Conference of the Cognitive Science Society, 702-708, Cambridge, MA
18. Singley, M. K. & Anderson, J. R. (1989). Transfer of Cognitive Skill. Cambridge, MA: Harvard