Recasting the Feedback Debate: Benefits of Tutoring Error Detection and Correction Skills

Santosh Mathan
CMU-HCII-03-102
CMU-CS-03-219
April 4th, 2003

School of Computer Science
Human Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Kenneth R. Koedinger, Chair
John R. Anderson
Albert T. Corbett
Marsha C. Lovett

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

This research was sponsored in part by the National Science Foundation under grant number CDA-9720359. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies or endorsement, either express or implied, of the NSF, the U.S. government.

Keywords: immediate feedback, delayed feedback, feedback, cognitive tutor, intelligent tutoring system, student model, errors, examples, spreadsheet, formulas, tutoring.

Abstract

Traditionally, intelligent tutoring systems have provided feedback on the basis of a so-called *expert model*. Expert model tutors incorporate production rules associated with error free and efficient task performance. These systems intervene with corrective feedback as soon as a student deviates from a solution path.

This thesis explores the effects of providing feedback on the basis of a so-called *intelligent novice* cognitive model. An intelligent novice tutor allows students to make errors, and provides guidance through the exercise of error detection and correction skills. The underlying cognitive model in such a tutor includes both rules associated with solution generation, and rules relating to error detection and correction. There are two pedagogical motivations for feedback based on an intelligent novice model. First, novice performance is often error prone and students may need error detection and correction skills in order to succeed in real world tasks. Second, the opportunity to reason about the causes and consequences of errors may allow students to form a better model of the behavior of domain operators.

Learning outcomes associated with the two models were experimentally evaluated. Results show that learners who receive intelligent novice feedback demonstrate better learning overall, including better retention and transfer performance than students receiving expert model based feedback.

Another focus of the research described here has been to help students form a robust and accurate encoding of declarative knowledge prior to procedural practice with an intelligent tutoring system. Examples have been widely used as a component of declarative instruction. However, research suggests that the effectiveness of examples is limited by the fact that inferences concerning the specific conditions under which operators may be applicable are only implicit in most examples, and may not be apparent to students without self-explanation. This thesis explores the effectiveness of a technique referred to in this thesis as *example walkthroughs*. Example walkthroughs interactively guide students through the study of examples. They present question prompts that help students make the inferences necessary to select problem solving operators that will lead to a solution. Students make these inferences by responding to multiple choice prompts. Evaluations suggest that example walkthroughs may provide a cost effective way to boost learning outcomes in intelligent tutoring systems.

Acknowledgements

I would like to thank my advisor, Ken Koedinger, for innumerable ways in which he contributed to this research and my professional development. I have come away from practically every meeting with Ken with a renewed appreciation for his intellect, creativity, and character. I consider myself extremely fortunate to have had the opportunity to collaborate with him over the years.

I would like to thank my committee members for their guidance. Albert Corbett's research and scholarship inspired my interest in the subject of feedback. Albert has always been generous with his time, and provided valuable feedback on every aspect of this work. I am also grateful to Marsha Lovett for her help over the years, particularly during the formative stages of this research. Marsha's comments and feedback helped me flesh out many of the early ideas that formed the basis of this research. I am also grateful to John Anderson for his comments and feedback. Despite a very busy schedule, and a request to join my committee at a relatively late stage of my research, John kindly agreed to be a committee member.

Greg Aist, Ryan Baker, Suresh Bhavnani, Ken Kwok, Seok Kwok, Daniel Morris, and Kerry Ojakian enriched my personal and intellectual life in countless ways over the course of my years at CMU.

I am grateful to my office mates Alex Weissman, Keith Weber, and Junlei Li for contributing to an enjoyable and intellectually stimulating environment every day.

Gloria Gruber in Human Resources provided outstanding help with recruiting participants for our studies.

I would like to express my thanks to Reny for being my best friend, and a reviewer of all my papers and presentations. I owe thanks to little Sara for all the joy she has brought.

I am grateful to my parents, Abraham and Anna, for all the encouragement, opportunity and love over the years.

This thesis is dedicated to the memory of my uncle and friend, Tom Abraham.

Table of Contents

Chapter 1

Introduction

Procedural knowledge has been defined as the ability to execute action sequences to solve problems (Rittle-Johnson, Siegler, & Alibali, 2001). As Anderson (1993) has suggested, this knowledge is optimized for efficient use and is limited to specific use contexts. Conceptual knowledge[1] on the other hand has been defined as the implicit or explicit understanding of principles that govern a domain and interrelations among units of knowledge in that domain (Rittle Johnson et. al., 2001). This knowledge is more flexible and can be applied broadly.

This thesis examines the efficacy of two avenues for fostering the joint development of procedural and conceptual knowledge in intelligent tutoring systems:

- First, this research assesses the benefits of interactively guiding students through the study of examples during declarative instruction.

- Second, this thesis examines the effects of structuring feedback to give students the opportunity to exercise error detection and correction skills.

These interventions were evaluated in two studies associated with a spreadsheet tutor.

---

[1] May be viewed as a particular type of declarative knowledge.

1.1 Design of Feedback

A major focus of the research described in this document is on the impact of feedback on the course and effectiveness of learning. It focuses specifically on the issue of when it might be appropriate to provide feedback.

Butler and Winne (1995) have defined *feedback* as "information with which a learner can confirm, add to, overwrite, tune, or restructure information in memory — this information may include domain knowledge, metacognitive knowledge, cognitive strategies and tactics." The source of feedback may be intrinsic or extrinsic to a task environment. Many task environments are rich in *internal feedback*. A learners actions produce clearly discernable consequences that point to the appropriateness of actions. This information can serve to guide subsequent actions. However, internal feedback may be non-existent or difficult for novices to interpret in many problem solving domains. In these contexts it may be necessary to provide external feedback to guide students through the problem solving process. In this document, the term feedback refers to *external feedback* – feedback that complements or substitutes the intrinsic feedback inherent in a task environment.

Inherent in any design decision concerning feedback is the issue of when it might be appropriate to intervene following an error. Designers are faced with a choice of presenting feedback as soon as an error is detected – *immediate feedback*, or presenting *delayed feedback* – giving learners an opportunity to detect and correct errors on their own. Research on the subject does not offer an unambiguous answer. A review of the literature might lead a reader to conclude that each of these feedback modalities offers distinct pedagogical advantages. Instructional designers may consequently face a mutually exclusive choice between the purported benefits offered by these two alternatives.

This chapter reviews research assessing the pedagogical benefits of immediate and delayed feedback and offers an integrative perspective that may enable an instructional designer to jointly realize the benefits offered by both modalities.

1.1.1 Feedback in Cognitive Tutors

Much of the discussion regarding feedback will be situated in the context of Cognitive Tutors. The design of Cognitive Tutors is grounded in the theoretical account of skill acquisition embodied in the ACT-R theory of cognition (Anderson, 1993). ACT-R suggests that knowledge associated with a skill is first encoded in a declarative form. Interpretive processes generate problem solving behavior. As a consequence of active problem solving, inert declarative knowledge is transformed into goal oriented production rules. Both declarative and procedural knowledge are strengthened through practice.

Cognitive tutors support the development of skilled performance by providing context sensitive hints and feedback to students over the course of problem solving. In cognitive tutors, feedback plays a central role in guiding the learning process. As such, the question of when to provide feedback is of crucial importance in the design of these systems.

The construction of Cognitive Tutors is based on design principles listed in Figure-1 (Anderson, Corbett, Koedinger & Pelletier, 1995). Tutors built on the basis of these principles have been successful in a variety of different domains – producing a standard deviation of improvement in student performance over traditional classroom interventions (Anderson et al., 1995). While there is broad agreement within the educational research community on most of these principles, the principle of immediate feedback has been the source of considerable controversy. This controversy is examined in more detail below.

- Represent competence as a production set
- Communicate the goal structure underlying the problem
- Provide instruction in the problem solving context
- Promote an abstract understanding of the problem solving knowledge
- Minimize working memory load
- Adjust the grain size of instruction with learning
- Facilitate successive approximations of the target skills
- Provide immediate feedback

Figure 1: ACT-R tutor design principles

1.1.2 Research Supporting Immediate Feedback

The prescription of immediate feedback is based on studies comparing the pedagogical effects of immediate and delayed feedback. Lewis and Anderson (1985) explored the issue of feedback latency in the context of a maze based adventure game. Each room in the maze had a set of features (such as lamp, fireplace, doorkeeper etc). Players had a set of operators (e.g. Bow, knock, light fire) that would, in the presence of certain features move them towards the ultimate goal of finding treasure. Subjects were trained to play the game in either an immediate or delayed feedback condition. In the immediate feedback condition subjects were notified any time they applied operators that would lead them towards dead ends. In the delayed feedback condition, subjects were allowed to pursue dead ends up to a depth of one room before being informed of the inappropriateness of a previous choice. Subjects then had to use a backup operator to back out of the dead end path. In a posttest, subjects trained in the immediate feedback condition were more accurate at specifying correct operators when presented with descriptions of room features. The only case in which delayed feedback subjects were more accurate was in the case of rooms with features indicative of dead ends – these subjects were more familiar with the use of the backup operator.

While the Lewis and Anderson study shows a distinct benefit for immediate feedback, certain characteristics of the task limit the generalization of these findings to other problem solving domains. Anderson, Conrad, and Corbett (1989) have commented that the maze task "was a situation where the total correct solution was never laid out before subjects and they had to integrate in memory a sequence of moves". In many problem-solving domains, particularly in many academic tasks, the final solution, along with intermediate steps, is available for learners to study. ACT-R suggests that these solutions allow students to learn by analogy.

Corbett and Anderson (2001) compared the pedagogical benefits of immediate and delayed feedback in the context of their LISP tutor. Students in their study worked with 4 different versions of the tutor. In the Immediate Feedback condition, students were presented with a feedback message as soon as an error was made. In the Error Flag Feedback condition students were informed that an error was made without any explanatory text concerning the nature of the problem or subsequent task interruption. In the Demand Feedback condition, the tutor would check for errors after following an explicit request from the student. In the No Feedback condition, students received no guidance during problem solving but were told whether their solution was correct at the end of the problem.

Performance on a paper and pencil post-test showed that all feedback conditions were better than no feedback. However, there were no statistically significant differences among feedback conditions in the post-test. The primary difference among the feedback conditions was in terms of the learning rate. Students in the immediate feedback condition completed training significantly faster than those in the Demand and No Feedback conditions. Immediate feedback served to minimize floundering and keep the leaning process efficient. These results represent the basis for the prescription of immediate feedback in Figure 1.

1.1.3 Research Supporting Delayed Feedback

Despite these results, the principle of immediate feedback has been criticized on two grounds. First, it has been suggested that immediate feedback offered by cognitive tutors is qualitatively different from that offered by human tutors. For instance, research by Merrill, Reiser, Merrill, & Landes (1995) reveals that human tutors do not intervene immediately on errors that may provide learning opportunities. Other researchers have observed that human tutors let learners engage in error detection and correction (Fox, 1993). Furthermore, research on human tutoring strategies shows that tutors try to instill a sense of control in learners (Lepper & Chabay, 1988). These observations demand close attention as the best human tutors produce better learning outcomes than cognitive tutors (Bloom, 1984; Anderson, Corbett, Koedinger, & Pelletier, 1995) – these differences in tutoring strategies could be among the factors that contribute to this difference.

Second, immediate feedback has been criticized on the basis of empirical studies that highlight benefits of delayed feedback. Some research suggests that delayed feedback may contribute to better retention and transfer. For instance, Lee (1992) compared immediate and delayed feedback in the context of a genetics tutor. Students in the immediate feedback condition received feedback as soon as an error was detected. In contrast, students in the delayed feedback condition received feedback at the end of the problem. As in Corbett and Anderson (2001), students in the immediate feedback completed training problems significantly faster. In a posttest the following day, students in both conditions performed equally well on problems encountered during training. However, students in the delayed feedback condition performed significantly better on a far transfer task.

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

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

	Immediate Feedback	**Delayed Feedback**
Efficiency	✓	
Transfer and Retention		✓

Figure 2: Tradeoff between the benefits of immediate and delayed feedback

Considered together these studies might suggest a potential trade-off between the benefits offered by immediate and delayed feedback (Figure-2). On the basis of the pattern of findings just summarized, some researchers (e.g. Bjork, 1994; Nathan, 1998) have argued that immediate feedback promotes efficiency during training, while delayed feedback might to lead to better retention and transfer performance. An explanation for these results could be found in the so-called guidance hypothesis -- described next.

1.1.4 The Guidance Hypothesis

The guidance hypothesis offered by Schmidt, Young, Swinnen & Shapiro (1989) provides an account of the tradeoff shown in Figure-2. According to the guidance hypothesis, feedback serves to precisely direct learner actions following each presentation of feedback. Students can get through problems by implementing prescriptions embodied in feedback messages. This may have the effect of boosting performance during, and immediately following training. However, 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 could prevent important secondary skills from being exercised — these skills could include error detection, error correction, and metacognitive skills. Activities that feedback may limit the practice of are considered in more detail below:

Debugging

Debugging is a requisite skill in many academic domains. Klahr and Carver's (1988) task analysis points to some important components of this skill:

- Determining whether a program functions as anticipated
- Noting the nature of discrepancies if any
- Identifying the likely location of bugs
- Identifying bugs
- Repairing bugs.

Many of these activities are preempted by immediate feedback. Students may not get the chance to independently exercise skills needed for error detection and correction. As a consequence, task performance may be impeded in circumstances that demand these skills.

Metacognition

Immediate feedback may also impede the acquisition of important metacognitive skills. This may occur in two ways: by reinforcing unsound beliefs about the learning process, and by preventing the exercise of important metacognitive processes.

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

Besides holding the potential for influencing unsound metacognitive beliefs, immediate feedback may deny students the opportunity to practice important metacognitive skills. Appropriately tailored, delayed feedback may offer students the opportunity to examine the products of their actions and monitor their progress without external help. Students take on more of the charge of verifying their solutions. Self monitoring of one's problem solving activity is an important characteristic of expert performance — one that is sorely lacking among many students (Schoenfeld, 1987; Palinscar & Brown, 1984). Shoenfeld (1987) has shown that modeling and practicing self monitoring skill can positively impact problem solving behavior among students.

The guidance hypothesis suggests that immediate feedback may promote the development of *generative skills*. Generative skills are skill components that are involved in selecting appropriate problem solving operators and implementing these operators in specific task contexts. However, *evaluative skills* – skills called for in evaluating the effect of applying these operators, correcting errors, and monitoring one's own cognitive process – may go unpracticed (Figure-3). These evaluative functions are instead delegated to feedback. Consequently, task performance may be compromised in situations where students must jointly exercise evaluative and generative skills. Transfer tasks and retention tests are representative of situations where student performance is likely to be error prone and subject to floundering — where the joint exercise of generative and evaluative skills may be essential for success.

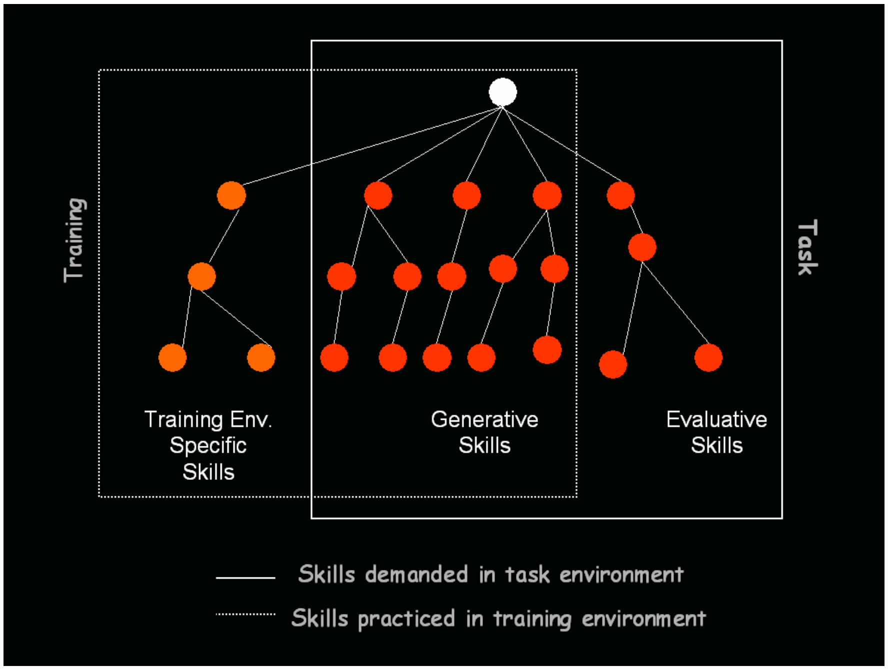


Figure 3: Discrepancy between skills supported by the training environment and skills required in task environment.

Additionally, the exercise of evaluative skills may provide an opportunity for a deeper conceptual understanding of domain principles. As Merrill, Reiser, Merrill, and Landes (1995) have theorized, errors provide an opportunity to develop a better model of the behavior of operators in a domain. They attribute this to the fact that error recovery requires that students construct explanations about the causes and consequences of errors and act on their analyses. This kind of active self-explanation and problem solving, they argue, contributes to a better understanding of domain operators and their applicability in problem contexts.

1.1.5 The Designers Dilemma

The research on feedback just summarized presents the designer with a dilemma. Immediate feedback keeps the learning process efficient. Additionally, some of the most effective and broadly used cognitive tutors provide immediate feedback on errors (Corbett, Koedinger, Hadley, 2001). However, a designer may also wish to realize benefits such as the development of debugging and metacognitive skills purportedly offered by delayed feedback. Unfortunately, the research reviewed here offers little guidance as to what an appropriate level of delay might be in a given context. This could have serious consequences. At best, an inappropriate level of delay can introduce inefficiencies into the learning process. At worst, delayed feedback can recede to a no-feedback condition, leading to unproductive floundering and student frustration.

1.1.6 An Integrative Perspective

Casting the debate on when to intervene following an error in terms of latency imposes an undesirable trade-off. Should the designer of an instructional system pursue the efficient and productive practice offered by immediate feedback? Or, should one attempt to realize benefits such as better retention and transfer that may be afforded by delayed feedback. However, a designer has to weigh these purported benefits of delayed feedback against possible costs. Inappropriate levels of delay may contribute to floundering and associated frustration.

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

Expert Model

Currently feedback in cognitive tutors is based on what is broadly referred to as an *expert model*. Such a model of desired performance characterizes the end-goal of instruction as error-free and efficient task execution. Feedback is structured so as to lead students towards expert-like performance. The tutor intervenes as soon as students deviate from a solution path. An expert model tutor focuses on the generative components of a skill. Figure-4 (left) 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* (c.f. Bruer, 1993). The assumption underlying such a model of desired performance is that an intelligent novice, while progressively getting skillful, is likely to make errors. Recognizing this possibility, the intelligent novice model incorporates error detection and error correction activities as part of the task. Feedback based on such a model would support the student in both the generative and evaluative aspects of a skill, while preventing unproductive floundering. While delayed feedback gives students the opportunity to exercise evaluative skills, an intelligent novice model based tutor explicitly models these skills and scaffolds students through the exercise of error detection and correction activities. Feedback with respect to a model of an intelligent novice may resemble delayed feedback, but it is really immediate feedback with respect to a model of desired performance that includes error detection and correction skills. Figure-4 (right) outlines student interaction with a tutor based on an intelligent novice model.

The analysis just presented recasts the feedback debate. The integrative perspective outlined here suggests that the model of desired performance, and not feedback timing, should be the crucial issue of focus in deciding when to intervene following an error. In the following pages this document will detail the design of two versions of a spreadsheet tutor – one based on an expert model the other on an intelligent novice model. Results from two studies evaluating learning outcomes associated with each will also be presented. However, before doing so, this document describes the theoretical motivations underlying the design of declarative instruction in the Excel Tutor.

Expert model	Intelligent Novice Model
Student reads problem statement and identifies goals to be accomplished Student plans actions to accomplish goals Student implements actions	Student reads problem statement and identifies goals to be accomplished Student plans actions to accomplish goals Student implements actions
Student attends to feedback If correct — student moves on If wrong — students has option to get instructions to fix problem	Student attends to outcomes and looks for discrepency between intended result and actual outcome If a discrepency is noted: Student identifies source of discrepency Student attempts to repair original solution If discrepency is missed, or repair attempt fails, student guided through error detection and correction process Student tests solution

Figure 4: Interaction with an Expert Model (left) and Intelligent Novice Tutor (right)

1.2 Design of Declarative Instruction

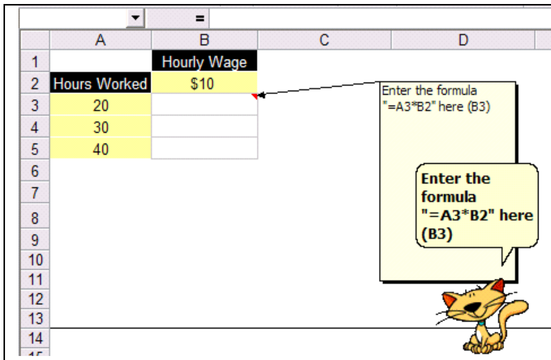
Declarative knowledge plays a crucial role in early skill acquisition. Under the ACT-R theory of skill acquisition (Anderson, 1993), declarative knowledge serves to structure initial problem solving attempts. Over the course of practice, knowledge compilation processes transform declarative encodings 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 (Singley & Anderson, 1989). For instance, in a study reported by Judd (1908), students were taught to hit underwater targets with darts. One group of students received both declarative instruction in principles of refraction, and procedural practice. The other group's instruction was entirely procedural. Both did equally well during training. However, performance differences became apparent in manipulations where the amount of water above the target was altered. Students whose instruction included declarative instruction in underlying principles were able to adapt their skill to the new circumstance much more quickly than students whose instruction was entirely procedural.

Examples have served as a tool for fostering the development of declarative knowledge (Chi, Bassok, Lewis, Reimann, & Glaser., 1989; Sweller & Cooper, 1985). 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. Researchers have observed that in order to use examples effectively students must make inferences about the specific conditions under which various operators are relevant and the transformations that result from applying these operators in various contexts (Chi et al., 1989). As Chi and colleagues (1989) have shown, good students do so through self explanations. However, they have also noted that many students fail to make the appropriate inferences.

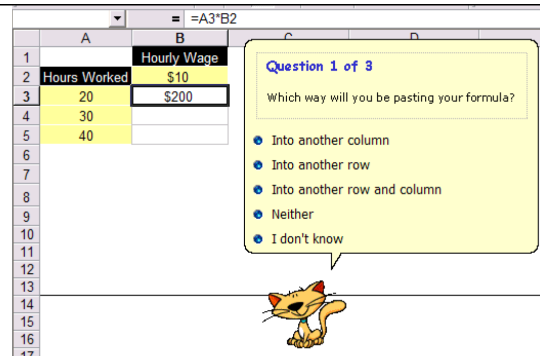
Recent research indicates that the effectiveness of examples can be enhanced by integrating elements of problem solving into the study of examples (Renkl, Atkinson, Maier, 2000). That is, students who study fully worked out examples, then complete intermediate steps in partially incomplete examples before problem solving, outperform students who transition directly to problem solving from the study of fully worked out examples. 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 – including Reciprocal Teaching (Palinscar & Brown, 1984), Cognitive Apprenticeship (Collins, Brown, & Newman, 1989), and PALs (Reif & Scott, 1999). On the basis of a CASCADE model of fading examples, Jones and Fleishman (2001) have suggested 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 *example walkthroughs* to facilitate a robust and accurate encoding of declarative knowledge. 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 examples demonstrated in the video with the help of example walkthroughs. Walkthroughs step students through the reasoning necessary to solve the example problems (Figure-5). Question prompts serve to guide students through the series of inferences necessary to select the sequence of operators that will lead to problem solving success. Incorrect inferences, which may result from an inaccurate or partial encoding of relevant declarative knowledge, are remedied with brief explanatory messages.

Example Walkthroughs differ from conventional approaches to declarative instruction in several ways: First, declarative information has traditionally been presented in a passive form (usually in the form of text, lecture, or video). In contrast, walkthroughs actively engage students in elaborating on information presented in video and text. Second, inferences concerning the applicability conditions of problem solving operators are typically implicit in examples. Example walkthroughs step students through the inferences necessary to pick operators that will lead to problem solving success. Third, any conceptual gaps are remedied immediately following the exposition of a concept instead of being deferred to problem solving contexts where working memory loads may be high.



1. Students is taken step-by-step through the actions necessary to solve a problem. However, the student does not implement system instructions by rote. Appropriate actions are selected on the basis of inferences made by the student (as illustrated in 2.)



2. Multiple choice prompts scaffold students through the inferences necessary to select operators that will lead to problem solving success. Incorrect responses, resulting from faulty declarative structures, can be remedied with brief explanatory messages.

Figure 5: Screenshots of walkthroughs from a spreadsheet tutor to be described in later pages

1.2.1 ACT-R analysis of Example Walkthroughs

ACT-R suggests that learning is probabilistic in nature – that is, there is some probability that students may fail to encode information in a way that can be recalled appropriately in a given context of use. There are numerous ways in which declarative knowledge may fail to be retrieved appropriately when needed (Anderson, 1993, pg. 70). First, students may fail to encode examples accurately. Furthermore, examples may be encoded weakly or encoded at study in a manner that is unsuitable for retrieval in a specific task context. Additionally, as a result of inadequate motivation, students may fail to attend to declarative instruction appropriately and fail to encode information entirely. Each of these possibilities is discussed in conjunction with details of how walkthroughs may play a role in circumventing these outcomes.

Appropriate knowledge encoding at the symbolic level

Numerous researchers have noted that student often fail to encode declarative knowledge relevant to successful problem solving in an appropriate way (e.g. Chi et al., 1989; Reif & Allen, 1992; Pirolli & Anderson, 1985). Consider the example[2] in Figure-6. A student may note the fact that the quantity Hourly-Wage lies in a single cell, while the quantity Hours-Worked is specified in different cells and form associated working memory elements. This encoding may form the basis for the creation of the rule reflected in Figure-6. That is, the student may assume that the '$' sign always precedes the cell reference for the quantity that is only represented in a single cell. While this shallow rule may be appropriate for problems isomorphic to the example, the declarative encodings and associated production rules are likely to be inadequate to solve the problem depicted in Figure-7.

Example walkthroughs have the potential for reducing inappropriate encoding of declarative chunks by focusing student attention on elements of a problem that are crucial to effective problem solving. Walkthroughs step students through the problem solving process and lead them to attend to the features of a problem relevant to effective problem solving.

Example walkthroughs also mitigate the possibility inappropriate encoding of declarative knowledge as a result of poor attention. As a consequence of the interactive nature of example walkthroughs, students are likely to be more actively engaged in the knowledge acquisition process than students acquiring new knowledge purely on the basis of passive alternatives such as video and textual expositions.

Strength of encoding

Retrieval of declarative knowledge relevant to a specific problem solving context in ACT-R is governed by the base level strength of declarative chucks coupled with activation received from associated chunks (Anderson, 1993, pg. 70). The theory predicts that memory increases as a power function of practice and decreases as a power function of delay.

---

[2] This example assumes an understanding of spreadsheet cell referencing concepts. An overview of cell referencing concepts is presented in Chapter-2.

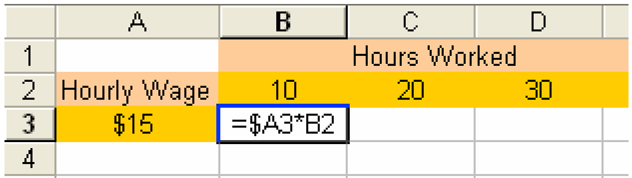
Walkthroughs provide students with the opportunity to use the knowledge presented through videos and text immediately following introduction. Relevant knowledge is rehearsed and strengthened immediately following presentation instead of being deferred to later problem solving context where working memory resources may be additionally taxed.

Encoding in context of use

ACT-R suggests that the activation of declarative knowledge chunks is also determined by activation received over associative links from other chunks activate in a given context. Walkthroughs provide students with the opportunity to practice applying newly introduced knowledge in a context that is very close to later problem solving contexts. This has the potential for providing opportunities to build redundant associative links to other declarative chunks that are likely to be active in problem contexts. The close correspondence between the conditions under which declarative knowledge is introduced and the actual context of use makes it more likely that chunks relevant for problem solving will be active when required.

The analysis just presented points to features that could contribute to the effectiveness of learning from examples. First, students may be more likely to encode information relevant to problem solving if their attention is explicitly directed to important elements of an example. Second, opportunities to elaborate on newly acquired information may foster a more robust encoding of declarative knowledge. Third, checks of student comprehension in conjunction with the study of examples could contribute to a more accurate encoding of declarative knowledge.

We now describe implementation of feedback and declarative instruction based on the analysis just presented. We do so in the context of a spreadsheet tutor.



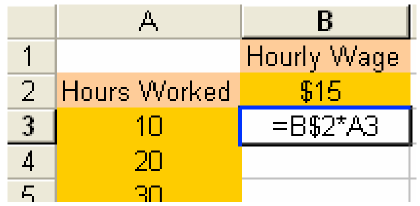
DECLARATIVE ENCODING

```
Variable-1>
isa variable
quantity-label hourly-wage
orientation lies-in-a-single-cell
reference-for-first-cell A3


Variable-2>
isa variable
quantity-label hours-worked
orientation lies-in-different-cells
reference-for-first-cell B2
```

BUGGY PRODUCTION

```
IF

Variable-1>
isa variable
orientation lies-in-a-single-cell
cell-address =cell-1-reference

Variable-2>
isa variable
orientation lies-in-different-cells
cell-address =cell-2-reference

THEN

Enter a formula that multiplies =cell-
1-reference and =cell-2-reference and
insert a '$' sign ahead of cell-1-
reference
```

Figure 6: Inappropriate encoding of an example and a buggy production stemming from it. The declarative encoding fails to consider the vertical and horizontal orientation of variables



DECLARATIVE ENCODING

```
Variable-1>
isa group-of-cells
quantity-label hourly-wage
horizontal-orientation lies-in-one-column
vertical-orientation  lies-in-one-row
column-reference-of-first-cell B
row-reference-of-first-cell 2

Variable-2>
isa group-of-cells
quantity-label hourly-wage
horizontal-orientation lies-in-one-column
vertical-orientation  lies-in-different-rows
column-reference-of-first-cell A
row-reference-of-first-cell 3

Paste-Area>
isa group-of-cells
quantity-label Earnings
horizontal-orientation lies-in-one-column
vertical-orientation  lies-in-different-rows
```

PRODUCTION RULE

```
IF

Variable-1>
isa group-of-cells
quantity-label hourly-wage
horizontal-orientation lies-in-one-column
vertical-orientation  lies-in-one-row
column-reference-of-first-cell =Col-1
row-reference-of-first-cell =Row-1

Variable-2>
isa group-of-cells
quantity-label hourly-wage
horizontal-orientation lies-in-one-column
vertical-orientation  lies-in-different-rows
column-reference-of-first-cell =Col-2
row-reference-of-first-cell =Row-2

Paste-Area>
isa group-of-cells
quantity-label Earnings
horizontal-orientation lies-in-one-column
  vertical-orientation  lies-in-different-rows
column-reference-of-first-cell =Col-3
row-reference-of-first-cell =Row-3


THEN

Create formula that multiplies =Col-1 =Row-1
and =Col-2 =Row-2 and place a "$" sign ahead
of =Row-1
```

Figure 7: Appropriate encoding of an example and a production stemming from it. The declarative encoding appropriately considers the vertical and horizontal orientation of variables. This allows for precise determination of which row or column references may require absolute references.

Chapter 2

The Learning Domain — Cell Referencing

Spreadsheets have been widely regarded as exemplary end-user programming environments (Nardi, 1993). 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 users. A commonly reported usability problem concerns the appropriate use of absolute and relative references — these are schemes that allow users to perform iterative computations.

2.1 Absolute and Relative Referencing

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 letter, whereas row indices (often called row references) are denoted by number. Cells may contain alphanumeric data and formulas. Formulas can refer to values in specific cells by referring to their addresses. So a user could enter a formula in cell C3 (in column C and row 3) that adds the content of cell A3 and B3 by typing "=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-8. 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.

	A	B	C	D	E
1		January	February	March	Total
2	Rent	700	700	700	=B2+C2+D2
3	Electricity	60	53	72	=B3+C3+D3
4	Phone	100	58	75	=B4+C4+D4
5	Total	=B2+B3+B4	=C2+C3+C4	=D2+D3+D4	

	A	B	C	D	E
1		January	February	March	Total
2	Rent	700	700	700	2100
3	Electricity	60	53	72	185
4	Phone	100	58	75	233
5	Total	860	811	847	

Figure 8: Relative Referencing allows formulas in B5 and E2 to be reused by copying and pasting

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-8, =B2+B3+B4 becomes =D2+D3+D4 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-8, =B2+C2+D2 becomes =B4+C4+D4 when moved across rows from E2 to E4).

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

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

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

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

Figure 9: Inappropriate use of relative references (left) remedied with the use of absolute references (right)

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-9. The value in cell B2 (Hourly Wage) has to be multiplied with the values in cells A3, A4, and A5. 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 Cell Referencing: A difficult skill

Research suggests that the appropriate use of absolute and relative references presents difficulties for both novice and experienced spreadsheet users.

It has been observed that cell referencing skills are difficult for novices to learn. For instance, Doyle (1990) observed 78 undergraduate management students learning to use spreadsheets over the course of 15, hour and a half long sessions. The author recorded problems that persisted among students over the course of these sessions. Difficulties associated with the appropriate use of absolute and relative references are reported as one of ten persistent difficulties associated with learning to use spreadsheets.

The use of absolute and relative references also presents problems for experienced spreadsheet users. For instance, Hendry and Green (1994) interviewed users who develop and maintain complex spreadsheets as part of their work. Their interviewees included several university staff members who use spreadsheets for data analysis and simulations, a manager who keeps track of costs at a software company, and a secretary who maintains a financial reporting spreadsheet. Users cited the appropriate use of relative and absolute referencing as being difficult to learn and a common source of error over the course of routine use.

In a study involving nine IBM employees, Brown and Gould (1987) asked participants to carry out spreadsheet tasks that included data entry, data formatting, and data manipulation using formulas. Participants had an average of 2.7 years of experience using spreadsheets and reported using spreadsheets for an average of 8 hours each week. Despite the fact that participants expressed high confidence in the accuracy of the spreadsheets, Brown and Gould found over 40 percent of spreadsheets to contain errors. The inappropriate use of absolute and relative references contributed to 3 of the 17 errors observed.

Baxter and Oatley (1991) examined 16 experienced spreadsheet users learning to use a brand of spreadsheet they had no familiarity with. These users either used spreadsheets “at least once per week”, or had “completed a course on spreadsheets”, or “had taught spreadsheet use”. Most of their participants had backgrounds in accounting or business. While the brand of spreadsheet had no significant effect on task performance, the authors noticed that most experienced users hesitated to reuse formulas by copying and pasting in order to avoid errors.

The following pages describe the design and evaluation of a cognitive tutor designed to help students acquire cell referencing skills by doing.

Chapter 3
Study-1

The purpose of Study-1 was to compare the relative efficacy of a cell referencing tutor based on an intelligent novice model with one based on an expert model. Additionally, the study was aimed at assessing the effectiveness of example walkthroughs.

3.1 Expert Model Tutor Description

The expert model version of the spreadsheet tutor used in Study-1 emphasized the exercise of generative skills. Figure-10 illustrates the goal structure underlying the expert model tutor. The model of desired performance underlying a tutor based on an Expert Model emphasizes error free and efficient task performance. Feedback serves to guide students through the process of generating a solution to the problem. Any deviation from the solution path is remedied with immediate corrective feedback. Details of the design of declarative instruction and feedback design based on such a model are presented below. A production rule representation of the underlying expert model is specified in Appendix-1.

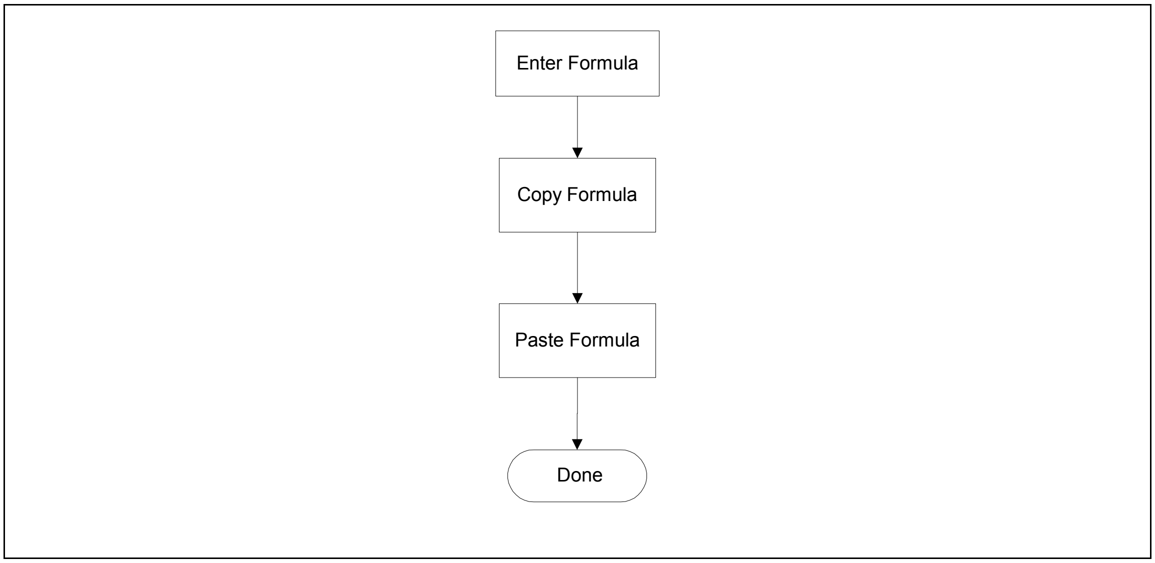


Figure 10: Goal structure associated with expert model based tutor (top)

3.1.1 Example Walkthrough

Declarative instruction in both the expert and intelligent novice versions of the spreadsheet tutor included 3 example walkthroughs. The first, illustrated the use of relative references; the second, focused on the automatic updates of formula references based on the direction of a paste operation (the mechanism underlying relative referencing); the third, focused on the appropriate use of absolute references. The first two walkthroughs were identical in the expert and intelligent novice versions of the spreadsheet tutor. However, the third walkthrough differed in the two versions of the tutor described here. The third walkthrough, focusing on the use of absolute reference, was designed to be consistent with the underlying model of desired performance embodied in each version of the tutor. Consequently, this document will focus on the design of the third walkthrough.

Example walkthroughs illustrating the use of absolute references in the expert model tutor used in Study-1 focused on generative skills. Students were 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.

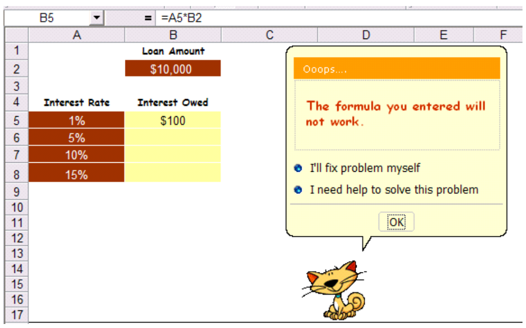
The expert model walkthrough guided students through these inferences by posing a series of questions:

- 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?

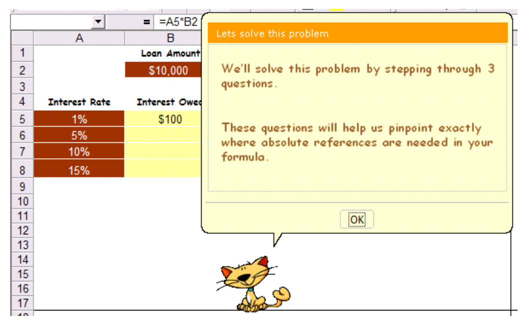
Students responded to these questions by picking from multiple-choice options. The system provided succinct explanations in response to errors. Screenshots of an example walkthrough associated with the expert version of the spreadsheet tutor are illustrated in Appendix-5.

3.1.2 Feedback

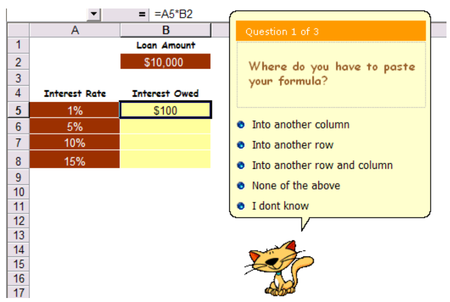
Students working with the expert model version of the spreadsheet tutor received feedback as soon as an incorrect formula was entered. The error notification message presented students with the choice of correcting the error on their own, or doing so with help from the system. If help was sought, students were guided through the process of identifying where, if any, absolute references were required in the formula (see Figure-11). Students were interactively guided by question prompts to solve the problem deductively. Screenshots of feedback associated with the expert version of the spreadsheet tutor used in Study-1 are presented in Appendix-3.



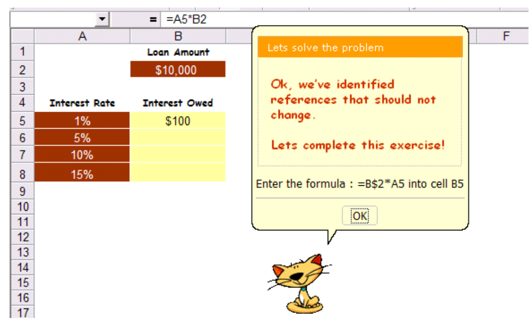
1. Student notified of error



2. Solution generation process is outlined



3. Student is stepped through a three step process to generate solution



4. Student prompted to enter correct formula

Figure 11: Feedback in the expert version of the tutor

3.2 Intelligent Novice Tutor Description

In addition to generative skills emphasized by expert model based tutors, an intelligent novice tutor provides practice in evaluative skills. Figure-12 illustrates the goal structure underlying the intelligent novice model. The intelligent novice model of performance anticipates the possibility of student errors and explicitly represents error detection and correction activities. As Figure-12 indicates, feedback guides students through the process of generating formulas, and copying and pasting them to verify the appropriateness of their solution. Additionally, the model guides students through the process of identifying and correcting bugs. Details of feedback and walkthrough based on the intelligent novice model are detailed below.

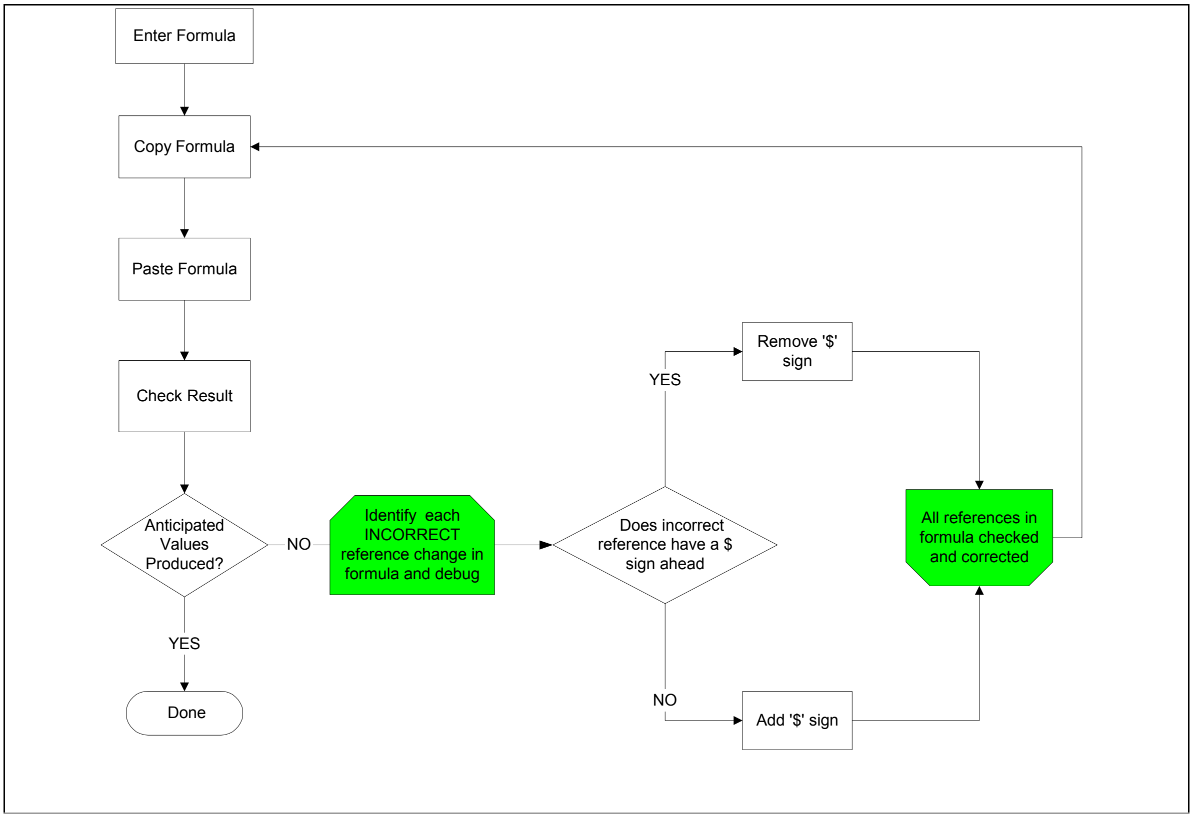


Figure 12: Goal structure associated with intelligent novice based tutor (top)

3.2.1 Example Walkthrough

In addition to helping students generate problem solutions, example walkthroughs in the intelligent novice version of the spreadsheet tutor guided students through the reasoning associated with the exercise of evaluative skills. First, students were prompted to predict the values and formulas that would result in each cell if the formula to be copied and pasted were to work correctly. Subsequently, students were prompted to copy and paste a formula without any absolute references into each cell of the example. Students were then guided to note the discrepancy between actual and intended values and formulas. Prompts served to help learners to use the identified discrepancies to determine where an absolute reference may be appropriate. Screenshots in Appendix-6 illustrate the steps embodied in the intelligent novice walkthrough.

3.2.2 Feedback

In contrast to the expert model tutor, the intelligent novice version allowed students to enter an incorrect formula and observe the consequences of pasting it over the relevant cells. Students receive an opportunity to detect the source of the error and correct the formula. Hints requested by students served to guide them through the error detection and correction process. An error in the formula correction step would result in immediate corrective feedback to minimize floundering and frustration. If a student failed to detect an error and tried to start a new problem, feedback directed the student to check for errors and request hints if needed. The error notification message at the formula correction step presented students with the choice of correcting the error on their own or doing so with help from the system.

If help was sought, the student was asked to predict formulas and values that would result if the original formula were to work accurately. These values and formulas were noted in a table on the spreadsheet (Figure-13). Subsequently, the student was asked to enter a formula without any absolute references and copy and paste it. Students were then prompted to note the values and formulas actually produced. Students were guided to use the discrepancy between actual and intended formulas to determine where absolute references, if any, were appropriate. Appendix-4 presents screen shots associated with the intelligent novice feedback in Study-1.

The next section describes results from an empirical evaluation contrasting learning outcomes associated with each of these models.

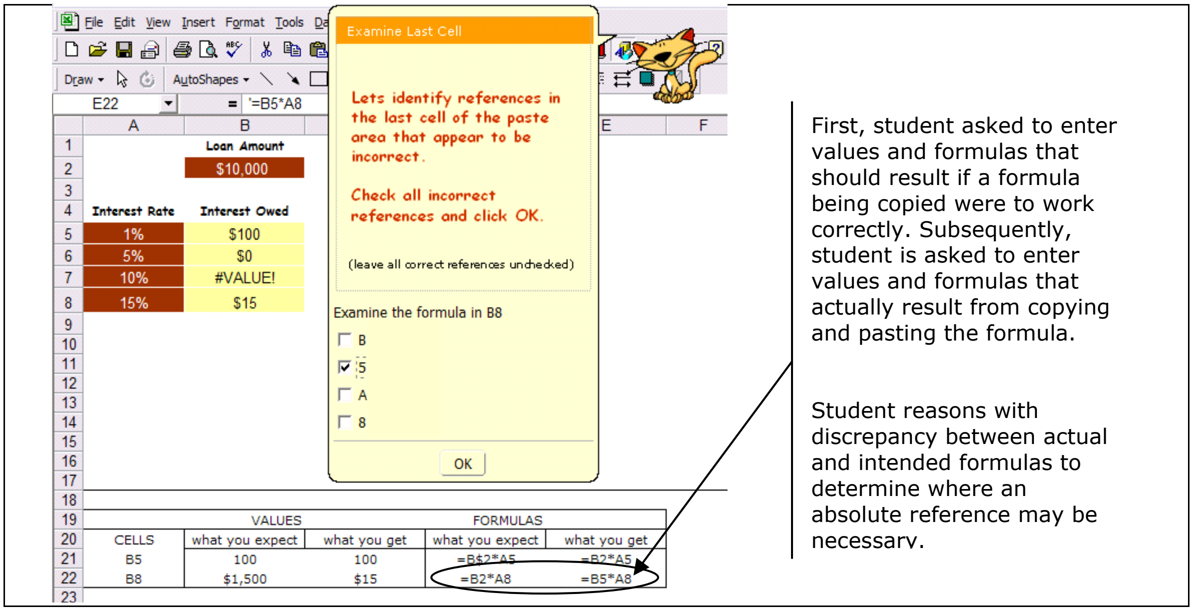


Figure 13: Feedback and hints in the Intelligent Novice condition guide students through error detection and correction activities. The so-called 'discrepancy table' at the bottom of is used to reason about errors.

3.3 Experimental Method

An evaluation was conducted with a group of 36 participants recruited from a temporary employment agency. All subjects had general computer experience, including proficiency with word processing, email, and web applications – however, they were all spreadsheet novices.

Students were randomly assigned 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 walkthroughs (WT, noWT). There were 8 students in the Intelligent Novice Walkthrough condition (IN-WT), 9 students in the Expert Walkthrough condition (EX-WT), 10 students in the Intelligent Novice No Walkthrough condition (IN-noWT), and 9 students in the Expert No Walkthrough condition (EX-WT).

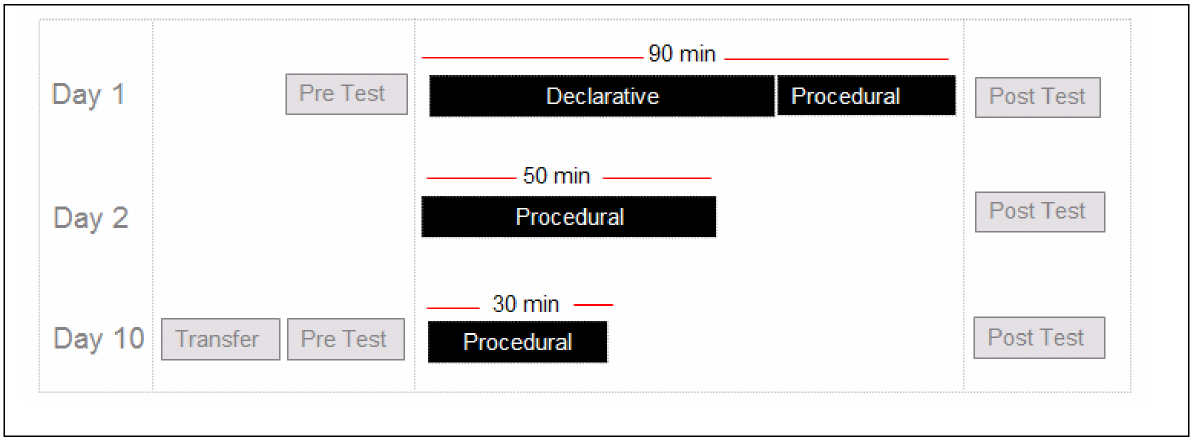


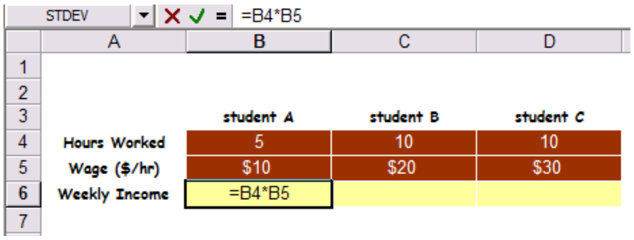
Figure 14: Procedure used in Study-1 and Study-2

The evaluation was conducted over the course of three days (Figure-14):

- 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. Students in the no walkthrough conditions went directly to procedural practice. 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. Procedural practice consisted of using the tutor to solve problems resembling the six types of problems illustrated in Figure-15. The session was preceded by a pre-test and was followed by a post-test.

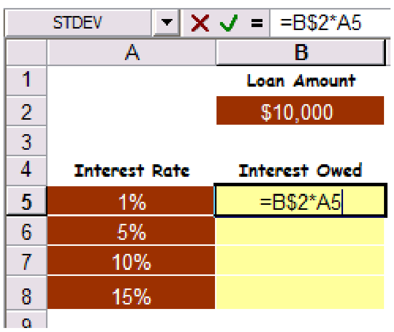
- Day-2: Students came in the next day for 50 minutes of procedural practice with the tutor. A post-test was administered following the instructional session.

- Day-10: Students came in for a third instructional session eight days after Day-2. Students attempted a pre-test and transfer task to measure retention prior to the instructional session. Then students had procedural practice for thirty minutes and finished with a final post test.

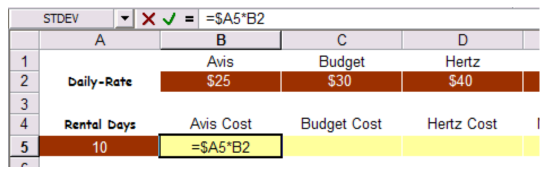


Type 1 Problem – formula with relative references only, pasted horizontally

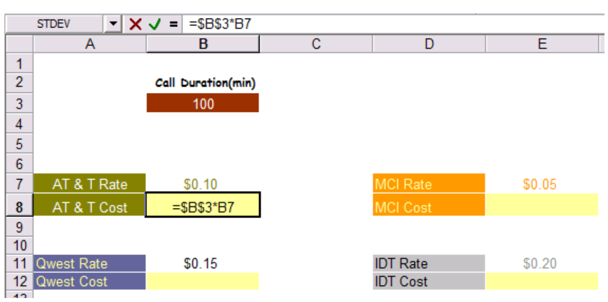
Type 2 Problem – formula with relative references only, pasted vertically



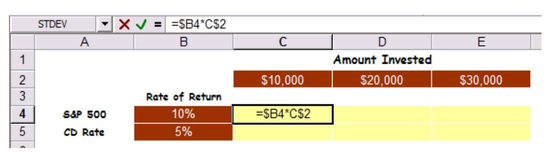
Type 3 Problem – formula with a single absolute row reference pasted vertically



Type 4 Problem – formula with a single absolute column reference pasted horizontally



Type 5 Problem – One of the cell references in the formula requires both an absolute row and absolute column reference in order to work when pasted both vertically and horizontally.



Type 6 Problem – One of the cell references in the formula requires an absolute column reference while the other requires an absolute row reference in order to work when pasted both vertically and horizontally.

Figure 15: Examples of the six types of problems represented in the tutor.

The pre and post-tests had two components: a test of problem solving (see Appendix 9) and a test of conceptual understanding (Appendix 10 & 11).

The problem-solving test consisted of problems isomorphic to training tasks. The conceptual test, which attempts to measure student understanding of cell referencing principles, consisted of two parts: the first part required students to exercise predictive skills (Appendix 11). 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 causal attribution skills (Appendix 10). 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 (Appendix 13). Students also were also asked to complete a computer experience questionnaire (Appendix 12). The questionnaire asked them to indicate the frequency with which they use various computer applications and rate their proficiency at each.

3.4 Results

No statistically significant differences were observed in student pre test scores (Table-1). The computer experience questionnaire provided the basis to assign a computer experience score to each participant. The computer experience score turned out to be a significant predictor of student performance (R^2=0.14, F(1,35)=5.74, p=0.02). As a consequence, the results reported here control for computer experience as a covariate.

cond	n		Computer Experience	Conceptual Pre Test	Coding Pre Test
IN-WT	8	Mean	91.8	17.4%	3.1%
		sd	25.4	13.4%	8.8%
		SE	8.9	4.7%	3.1%
		LS Mean	-	17.4%	3.1%
IN-noWT	10	Mean	84.3	22.1%	0.0%
		sd	37.9	14.1%	0.0%
		SE	11.9	4.4%	0.0%
		LS Mean	-	22.1%	0.1%
EX-WT	9	Mean	76.2	20.2%	0.0%
		sd	17.5	12.9%	0.0%
		SE	5.8	4.3%	0.0%
		LS Mean	-	20.2%	0.2%
EX-noWT	9	Mean	98.4	19.8%	0.0%
		sd	19.565	9.3%	0.0%
		SE	6.5	3.1%	0.0%
		LS Mean	-	19.9%	-0.2%
		FB Main Effect	F(1,32)=0.00, p=0.94	F(1,31)=0.00, p=.95	F(1,31)=1.23, p=.28
		WT Main Effect	F(1,32)=0.68, p=0.42	F(1,31)=0.26, p=.61	F(1,31)=1.43, p=.24
		FB*WT Interaction	F(1,32)=2.74, p=0.11	F(1,31)=0.31, p=.58	F(1,31)=0.82, p=.37

Table 1: Performance on pre-test measures

A repeated measure ANCOVA, over all the tests, shows a significant main effect for model and walkthrough. However, these main effects should be interpreted in light of the significant model-walkthrough interaction (F (1, 31) = 8.78, p = 0.006)(Figure 16) (Table 4). Overall, students in the expert-walkthrough condition outperformed students in all other conditions (Figure 16). A similar feedback by walkthrough interaction favoring the expert walkthrough condition was observed in the conceptual (F(1,31)=6.50, p =0.02) (Figure-17(C), Table-3) and problem-solving tests (F(1,31)= 5.59, p = 0.02) (Figure 17 (B), Table-2)

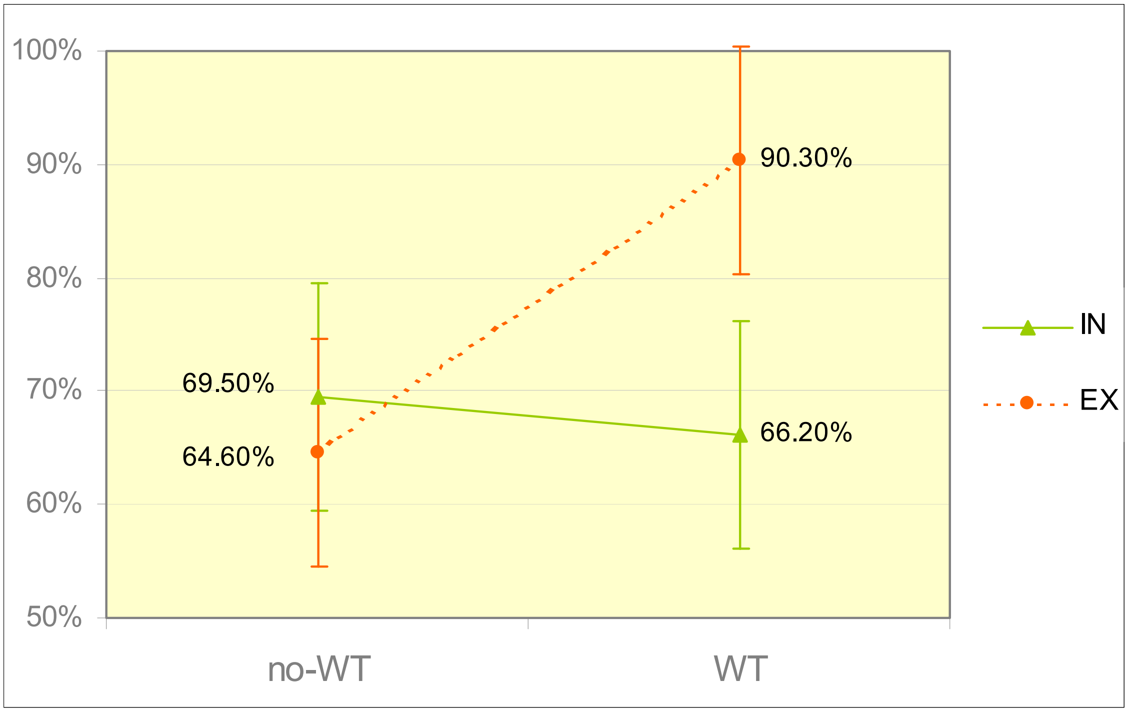
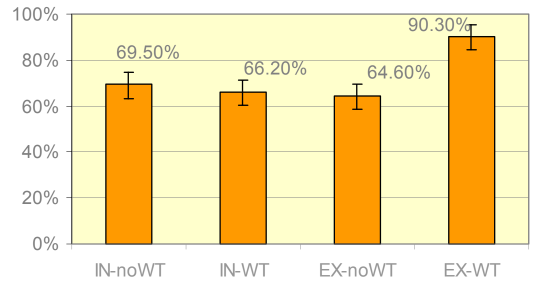
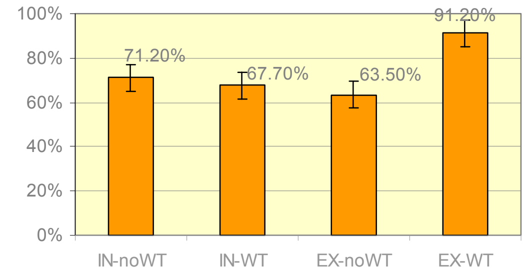


Figure 16: Overall Performance (averaging across all pre and post tests of problem solving and conceptual understanding – with the exception of the Day1 pre-test)

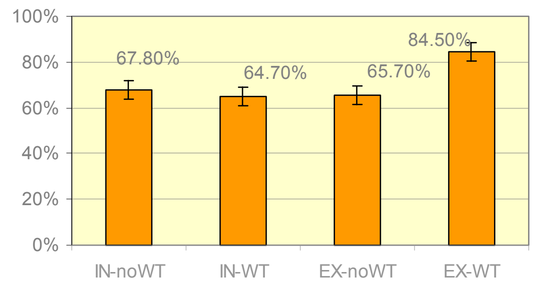
As shown in Figure-17 (E), students in the expert-walkthrough condition demonstrate the most robust performance on a retention test administered following an eight day retention interval (Day 10 – Pre Test). A similar pattern is observed when the problem solving and conceptual understanding components of the retention test are examined separately (Table-2 and Table-3). A marginally significant model-walkthrough interaction on the transfer task suggests that students in the Expert-Walkthrough condition were likely to apply their skills more broadly ($F=2.81$, $p=0.10$) (Figure-17 (D)).



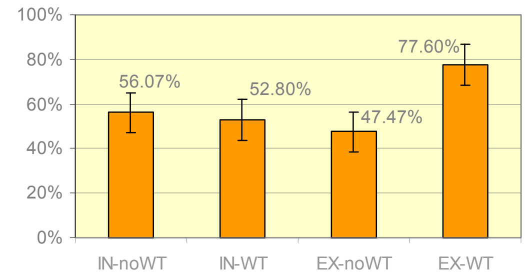
A Overall Performance (feedback*walkthrough interaction F(1,31)=8.78, p=0.006)



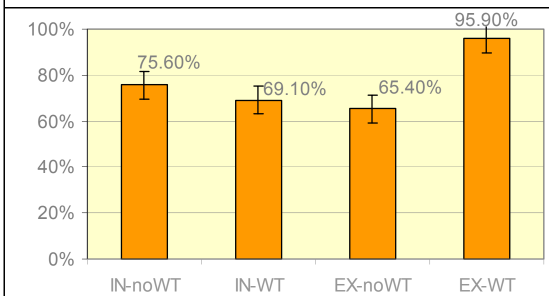
B Performance on coding test (feedback*walkthrough interaction F(1,31)= 5.59 p=0.02)



C Performance on test of conceptual understanding (feedback*walkthrough interaction F(1,31)= 6.50 p=0.02)



D Performance on transfer test (feedback*walkthrough interaction F(1,31)=2.81, p=0.10)



E Retention. Performance on a problem solving and conceptual understanding test following an eight day retention interval (feedback*walkthrough interaction F(1,31)= 9.14, p=0.01)

Figure 17: Summary of Results (Note: Error bars depict 95% confidence intervals)

cond	n		Day1-Post	Day2-Post	Day 10-Pre (Retention)	Day10-Post (Relearning)	Overall
IN-WT	8	Mean	**50.3%**	**86.3%**	**68.4%**	**73.4%**	69.6%
		sd	42.2%	17.1%	35.9%	33.3%	34.4%
		LS Mean	47.7%	84.9%	67.4%	70.9%	67.7%
IN-noWT	10	Mean	**64.5%**	**80.0%**	**51.5%**	**83.0%**	69.8%
		sd	34.4%	19.9%	41.2%	20.7%	31.9%
		LS Mean	66.5%	81.0%	52.3%	84.9%	71.2%
EX-WT	9	Mean	**77.8%**	**95.0%**	**96.1%**	**96.1%**	91.2%
		sd	25.5%	6.6%	5.5%	3.3%	15.2%
		LS Mean	84.7%	98.4%	98.8%	102.9%	96.2%
EX-noWT	9	Mean	**60.6%**	**73.9%**	**71.1%**	**67.8%**	68.3%
		sd	35.6%	32.1%	35.5%	38.6%	34.3%
		LS Mean	53.8%	70.5%	68.5%	61.1%	63.5%

Overall FB Main Effect $F(1,31)= 2.01$ $p=0.17$
Overall WT Main Effect $F(1,31)= 3.90$ $p=0.06$
Overall FB*WT Interaction $F(1,31)= 5.59$ $p=0.02$

Table 2: Performance on problem solving tests (averaging across all 8 problem solving pre/post tests – with the exception of Day1 pre-test)

cond	n		Day1-Post	Day2-Post	Day 10-Pre (Retention)	Day10-Post (Relearning)	Overall
IN-WT	8	Mean	**56.7%**	**71.0%**	**67.0%**	**68.7%**	**65.8%**
		sd	8.2%	16.6%	7.6%	15.8%	13.3%
		LS Mean	55.7%	69.9%	65.6%	67.4%	64.7%
IN-noWT	10	Mean	**65.0%**	**69.3%**	**68.2%**	**65.4%**	**67.0%**
		sd	20.3%	19.9%	20.0%	24.8%	20.6%
		LS Mean	65.7%	70.1%	69.3%	66.3%	67.8%
EX-WT	9	Mean	**69.8%**	**86.1%**	**84.1%**	**85.3%**	**81.3%**
		sd	10.9%	12.6%	16.1%	16.4%	15.2%
		LS Mean	72.4%	88.9%	87.8%	88.8%	84.5%
EX-noWT	9	Mean	**56.0%**	**78.6%**	**67.5%**	**73.0%**	**68.8%**
		sd	12.0%	13.1%	19.9%	22.4%	18.7%
		LS Mean	53.5%	75.9%	63.9%	69.6%	65.7%

Overall FB Main Effect $F(1,31)= 4.57$ $p=0.04$
Overall WT Main Effect $F(1,31)= 3.48$ $p= 0.07$
Overall FB*WT Interaction $F(1,31)= 6.50$ $p=0.02$

Table 3:Performance on tests of conceptual understanding (averaging across all 8 conceptual pre/post tests – with the exception of the Day-1 pre test)

cond	n		Overall Performance on all 16 Pre and Post Tests
IN-WT	8		66.2% (Mean: 67.7%, sd: 25.9%)
IN-noWT	10		69.5% (Mean: 68.4%, sd: 26.7%,)
EX-WT	9		90.3% (Mean: 86.3%, sd: 15.9%)
EX-noWT	9		64.6% (Mean: 68.5%, sd: 27.4%)
		FB Main Effect	F(1,31)=4.18, p=0.05
		WT Main Effect	F(1,31)=5.57, p=0.02
		FB*WT Interaction	F(1,31)=8.78, p=0.006

Table 4: Overall Performance (averaging across all 16 conceptual and problem solving tests)

cond	n		Transfer Test Scores
IN-WT	8		52.8% (Mean: 55.2%, sd: 30.9%)
IN-noWT	10		56.1% (Mean: 54.3%, sd: 32.2%)
EX-WT	9		77.6% (Mean: 71.3%, sd: 22.8%)
EX-noWT	9		47.5% (Mean: 53.7%, sd: 39.5%)
		FB Main Effect	F(1,31)=0.72, p=0.40
		WT Main Effect	F(1,31)=1.92, p=0.18
		FB*WT Interaction	F(1,31)=2.81, p=0.10

Table 5: Transfer Performance

cond	n		Problems Completed Over 3 Training Sessions
IN-WT	8		49.6 (Mean: 53.6, sd: 48.0)
IN-noWT	10		81.7 (Mean: 78.7, sd: 48.6)
EX-WT	9		93.0 (Mean: 82.6, sd: 28.5)
EX-noWT	9		71.3 (Mean: 81.6, sd: 40.3)
		FB Main Effect	F(1,31)= 2.04, p=0.16
		WT Main Effect	F(1,31)= 0.20, p=0.66
		FB*WT Interaction	F(1,31)= 4.97, p=0.03

Table 6: Problems Completed (over all 3 training sessions)

cond	n		Retention Performance
IN-WT	8		69.1% (Mean: 71.1%, sd: 25.3%)
IN-noWT	10		75.6% (Mean: 74.2%, sd: 24.0%)
EX-WT	9		95.9% (Mean: 90.7%, sd: 12.8%)
EX-noWT	9		65.4% (Mean: 70.4%, sd: 30.7%)
		FB Main Effect	F(1,31)= 1.97, p=0.17
		WT Main Effect	F(1,31)= 4.09, p=0.05
		FB*WT Interaction	F(1,31)= 9.14, p=0.01

Table 6: Retention Tests (performance on tests of problem solving and conceptual understanding following an eight day retention interval [Day 10 – Pre Test])

Qualitative differences were observed in the way students in each condition dealt with errors during training. 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 eventually 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.

3.5 Discussion

Study-1 contrasted learning outcomes associated with a tutor that models an Intelligent Novice with an Expert Model based tutor that focuses exclusively on generative skills. Study-1 was also aimed at assessing learning outcomes associated with example walkthroughs.

Contrary to 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. 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.

Better performance in the expert walkthrough condition may be explained by a combination of relatively low cognitive load during practice and the provision of an explicit procedure for interpretive use of declarative knowledge. Furthermore, walkthroughs are likely to have contributed to a robust encoding of declarative knowledge. The basis for these claims is detailed below.

1. Working Memory Load

Procedural practice with the intelligent novice model was more taxing on working memory than with the expert model tutor for at least two reasons. 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-13). These features were also inherent in the intelligent novice example walkthroughs, potentially compromising their efficacy.

2. Explicit Procedure to Guide Problem Solving

Students in the expert conditions 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 (Reif and Allen, 1992). 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.

3. 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, $F=3.09$, $p < 0.09$).

Study-1 highlighted a few areas that require further investigation. First, while Example Walkthroughs show the potential for better learning outcomes than procedural practice with a cognitive tutor alone, the impact of walkthroughs on learning outcomes must be replicated in future studies. The need to replicate the results associated with walkthroughs is partly warranted by the fact that walkthroughs only produced a reliable difference in the Expert-Walkthrough condition and not the Intelligent Novice-Walkthrough condition.

Second, Study-1 illustrates the fact that there is a risk associated with the concurrent exercise of generative and evaluative skills. The interface elements that support the exercise of error detection and correction skills have consequences on working memory load. Study-2, described next, involved an attempt to reduce the working memory load associated with the exercise of evaluative skills in the intelligent novice condition.

Chapter 4

Study-2

As the discussion associated with Study-1 has suggested, it is plausible that relatively high working memory loads associated with the intelligent novice condition may have impeded learning. Study-2 was aimed at establishing whether learning outcomes associated with the intelligent novice condition could be improved by reducing the working memory load imposed on students.

Several researchers have suggested that artifacts of a learning environment can tax limited cognitive resources. For instance, Anderson, Corbett, Koedinger & Pelletier (1995) include a prescription for reducing working memory load in a list of tutor design principles. They base this prescription on the fact that learning a new production rule in ACT requires that all the relevant information (relevant to the condition and action of the to-be-learned production) be simultaneously active in memory. Keeping other information active could potentially interfere with learning target information. This principle, they suggest, calls for minimizing presentation and processing of information not relevant to the target productions.

John Sweller and colleagues have argued that greater attention to the role and limitations of working memory by instructional designers can have a positive impact on learning outcomes. Their research suggests that high working memory loads can interfere with learning. Sweller has pointed to specific instructional design elements that are likely to contribute to working memory load. These include, extended problem solving activity (Sweller and Cooper, 1985; Sweller, 1988) and split attention – where students have to integrate information from various sources (Chandler and Sweller, 1991, 1992; Ward and Sweller 1990).

4.1 Working Memory Load in the Intelligent Novice Condition

Several design elements associated with walkthroughs and feedback in the intelligent novice tutor are likely to have taxed working memory resources in the intelligent novice condition. We examine each of these in turn below:

4.1.1 Extended Problem Solving Activity

Sweller has suggested that expert problem solving behavior is characterized by a reliance on domain specific knowledge, in the form of schemas. He has defined schemas as cognitive structures that allow "problem solvers to recognize a problem state as belonging to a particular category of problem states that normally require particular moves." (Sweller, 1988, p. 259)

Sweller (1998) has argued that problem solving, characterized by extensive means ends analysis, can impede schema acquisition. He has proposed two ways this could happen. First, means ends analysis demands selective attention to the differences between the current problem solving state and the goal state. However, effective schema acquisition, in contrast, demands attention to previously used problem solving operators and the relations between problem states and operators. Sweller (1988) has argued that these elements that must be attended to for schema acquisition may be ignored by learners engaged in means ends analysis. Second, working memory load imposed by means ends analysis may substantially limit the cognitive processes that can be devoted to schema acquisition. Means ends analysis requires simultaneous consideration of the current problem state, the goal state, relationship between problem solving operators, and often requires that users maintain a goal stack to manage sub goals in the problem solving process.

Error detection and correction activities in the intelligent novice tutor used in Study-1 are likely to have prolonged the problem solving process and contributed to a fairly high working memory burden. The process of filling out the so-called discrepancy table – an area in the spreadsheet to keep track of expected and actual values and formulas – required students to maintain subgoals unrelated to the original problem solving context (see Appendix-4). For instance, the process of specifying expected values required computing the product of various quantities – not an easy task for several students. Additionally, the process of entering formulas actually obtained by copying and pasting was rather cumbersome. In order to specify the formulas obtained after copying and pasting a formula, students had to click on the appropriate cells in the problem, examine the formula in the formula bar, store the formula in memory, return to the appropriate cell in the discrepancy table, recall the formula, and type it in. Errors in computing the anticipated values and formulas, and typos in entering actual values and formulas contributed to nested problem solving episodes that had little to do with gaining a better understanding of cell referencing concepts.

An important focus of the design effort that preceded Study-2 was to reduce the number of steps involved in the error detection and correction process and to reduce the opportunity for errors while exercising these skills.

4.1.2 Split Attention

Sweller and colleagues (Ward and Sweller, 1990; Chandler and Sweller, 1991, 1992) have shown that integrating information from multiple sources can have detrimental effects on learning. They theorize that cognitive processes required to integrate information from various sources can impede schema acquisition. The design of the intelligent novice tutor in Study-1 often required students to split attention across multiple areas of the spreadsheet.

Students had to attend to the part of the spreadsheet containing the problem, the discrepancy table used to keep track of expected and actual values, and prompts from the office assistant.

Additionally, much of the reasoning about the causes and consequences of errors required reasoning with formulas produced following a copy and paste operation. However, formulas are not usually visible directly on the spreadsheet – only the results produced by underlying formulas are displayed on the spreadsheet. Users have to take multiple steps to view the formulas underlying cells requiring consideration. Students were observed using one of two procedures to identify formulas underlying cells. Some students would select a cell and view the underlying formula by examining the formula bar. Students would have to commit the formula to memory, return to the discrepancy table, recall the formula, and type it into the appropriate part of the discrepancy table. Other students would double click on the relevant cell in order to view the underlying formula. This would place the cell in edit mode, revealing the underlying formula. Students would commit this formula to memory, hit enter to revert to a non edit mode, return to the discrepancy table, recall the formula and type it into the appropriate part of the discrepancy table.

4.2 Redesigned Intelligent Novice Tutor Description

4.2.1 Feedback

The analysis of problems with the original intelligent novice interface pointed to possible improvements. The intelligent novice version of the tutor was redesigned prior to Study-2. It embodied the following features:

The system was redesigned to reduce the number of error detection and correction steps. Steps associated with the exercise of error detection and correction activities were reduced from close to twenty to just two. Additionally, the possibility of error during the error detection and correction process was reduced by eliminating the need for typing. Instead, students were guided through error detection and correction activities using multiple choice prompts(see Figure-18 (left)).

Efforts were also made to reduce the need for students to integrate information distributed across various parts of the spreadsheet. The discrepancy table was eliminated both in the walkthroughs and feedback associated with the intelligent novice version of the tutor. The system was redesigned so that all the reasoning about errors and their consequences occurred in the context of the problem. Formulas relevant for students to consider were made visible using tags. Additionally, the system was redesigned to provide visual cues to highlight the discrepancy between actual and intended outcomes (see Figure-18). In the original intelligent novice tutor, students reasoned about the consequences of errors by comparing formula strings in the discrepancy table. The redesigned intelligent novice tutor highlights cells incorrectly referenced by a formula. As a consequence, students reason about errors in the original problem solving context with visual cues instead of reasoning with abstract symbol strings in a setting separated from the original problem context.

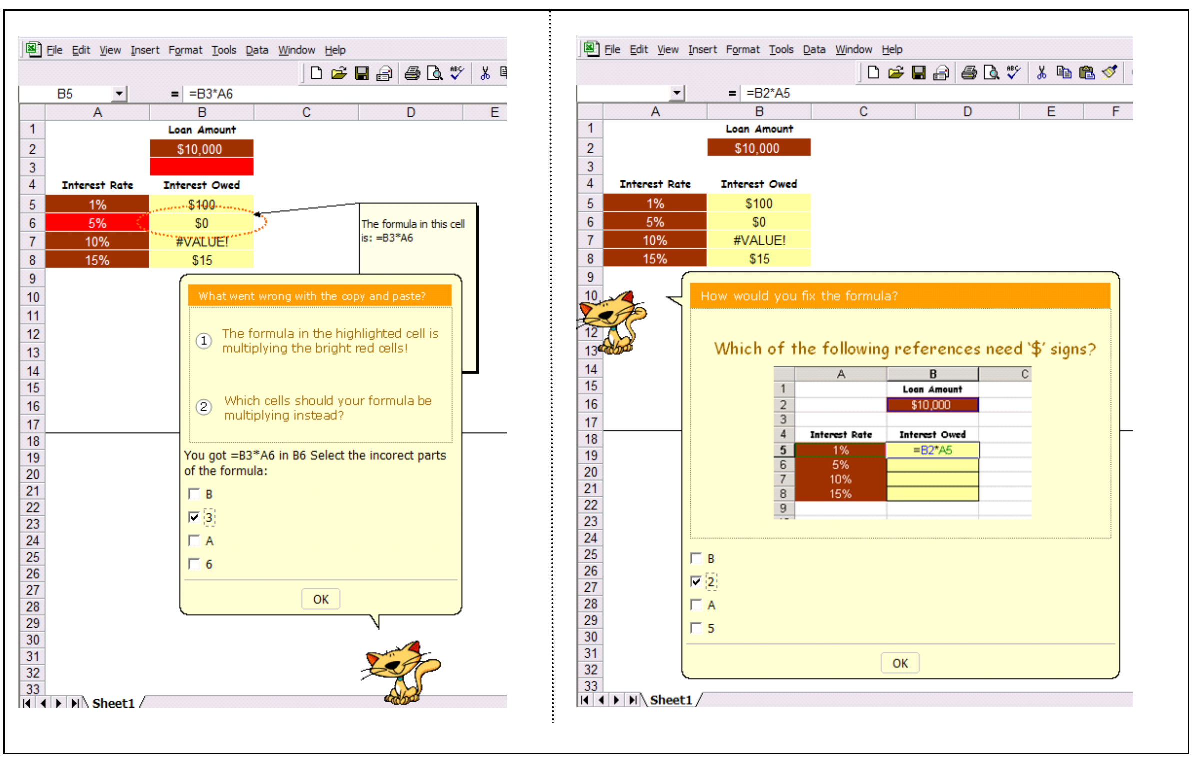


Figure 18: Feedback in the redesigned intelligent novice tutor. Step-1 (left) highlights the discrepancy between actual and intended outcomes and prompts students to identify the error. Step-2 (right) prompts students to generate a solution to the error detected in Step-1

Like the intelligent novice tutor in Study-1, the redesigned intelligent novice tutor allowed students to enter an incorrect formula, copy it, and paste it to observe the consequences of the error. The student was given an opportunity to detect errors and make corrections if necessary. However, if a student failed to detect an error and tried to move on to a new problem, feedback directed the student to check for errors and request hints. An error at the formula correction step resulted in immediate corrective feedback to minimize unproductive floundering. The error notification message at the formula correction step presented students with the choice of correcting the error on their own or doing so with help from the system. If a student chose to correct the error with help from the system, the student was taken through a two step process to get back on track. The first step (Figure 18 – left) focused on error detection. The system picks a cell that a student may have pasted an incorrect formula into, and highlights the cells inappropriately referenced by the underlying formula. Additionally, a tag indicating the incorrect formula underlying the selected cell is also shown. On the basis of these cues a student can determine the discrepancy between actual and intended outcomes and identify incorrect references. The second step (Figure-18 (right)) focuses on error correction. Having detected the source of error in the first step, the second step required students to identify references that must be held fixed in order for the formula to work as intended.

4.2.2 Example Walkthrough

Example walkthroughs associated the intelligent novice version of the tutor used in Study-2 were modified to eliminate the use of the discrepancy table. After reading the relevant declarative instruction and observing video examples, students solved the problems illustrated in the videos with the help of example walkthroughs. The walkthroughs embodied the following steps:

- Students were asked to copy and paste a formula without any absolute references over the relevant cells in a problem (the solution to the problem required at least one absolute reference).

- Students were asked to indicate whether the formula worked as intended.

- Students were prompted to identify the part of the formula the formula that did not change appropriately.

- Following error identification, students are guided through a three step process to generate a solution to the problem. The three steps asked students to consider the direction in which the formula was to be pasted, determine the type of references that were likely to be affected (row references, column references, or both), and among the references that are likely to change, identify the references that must be held fixed.

The walkthrough used to illustrate the use of absolute references in the intelligent novice condition of Study-2 was identical to the corresponding walkthrough in the expert condition of Study-1 (see Appendix-5). The walkthrough associated with the use of absolute references in the expert condition were modified prior to Study-2 (see Appendix 7). The motivations for these changes are described in the section 4.3.2.

4.3 Expert Model Tutor Description

Study-2 provided the opportunity to compare learning outcomes associated with the redesigned intelligent novice tutor, with an expert model tutor.

4.3.1 Feedback

Feedback in the expert model tutor used in Study-2 was identical in every respect to the version used in Study-1. Students working with the expert model version of the spreadsheet tutor received feedback as soon as an incorrect formula was entered. The error notification message presented students with the choice of correcting the error on their own or doing so with help from the system. If help was sought, students were guided through the process of generating a formula that would work appropriately.

4.3.2 Example Walkthrough

While expert model feedback remained the same in both studies, a modification was made to the walkthrough used to illustrate the use of absolute references. The goal of this modification was to make the distinction between the instructional consequences of the expert and intelligent novice models of desired performance more clear.

In Study-1, the expert model example walkthrough illustrating the use of absolute references gave students practice in both generative and evaluative skills. In other words, in the walkthroughs (but not in the tutor's feedback), the instruction was based on an intelligent novice model of desired performance. Before being guided through the process of generating a solution to walkthrough problems, students were given the opportunity to observe the consequences of copying and pasting a formula without absolute references (in a problem that required at least one absolute reference in order to work correctly). Subsequently, over a sequence of three steps, students were given the opportunity to identify the source of the error. This was done in order to give students a chance to see why absolute references were sometimes necessary. But, in principle, seeing why may not be a prerequisite for efficient expert performance. Thus, because the emphasis of the expert model tutor is on generative skills, the walkthrough designed to illustrate the use of absolute references was redesigned to focus on generating a solution. Evaluative components in the walkthrough were removed in order to correspond more closely to the model of desired performance associated with the expert condition – and, as a consequence, make the instruction associated with the intelligent novice and expert versions of the tutor more distinct. Appendix-7 illustrates the steps associated with the absolute referencing walkthroughs in Study-2 (Appendix-5 illustrates steps in the corresponding walkthroughs in Study-1.)

In all other ways, the walkthroughs associated with the expert condition were identical to those used in Study-1.

4.4 Experimental Method

An experimental comparision of the two tutors just described was conducted with a group of 49 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 independent factors: feedback (intelligent novice or expert model feedback), and availability of walkthroughs during declarative instruction (walkthrough, no walkthrough). There were 12 participants in the expert no-walkthrough condition (EX-noWT), 12 in the expert walkthrough condition (EX-WT), 13 in the intelligent novice no-walkthrough condition (IN-noWT), and 12 in the intelligent novice walktrough condition (IN-WT).

With the exception of the inclusion of a math test in Study-2, the procedures and tests used in Study-2 were identical to those used in Study-1. All students were assessed for mathematical ability using a test of basic algebraic symbolization. The math assessment was included because it appeared that basic algebraic reasoning ability may be related to the ability to master the use of formulas and cell referencing concepts.

4.5 Results

4.5.1 Pretest Measures

Table 7 summarizes student performance on pre-test assessments of computer experience, mathematical ability, cell referencing coding performance, and conceptual understanding of cell referencing concepts. An analysis of these pre-test measures revealed no reliable differences among conditions.

cond	n		Computer Experience	Conceptual Pre Test	Coding Pre Test	MATH (out of 5)
IN-WT	12	Mean	**88.4**	**17.3%**	**0.0%**	**3.6**
		sd	19.2	12.6%	0.0%	1.4
		LS Mean		16.7%	0.0%	
IN-noWT	13	Mean	**85.9**	**21.2%**	**4.6%**	**3.6**
		sd	39.5	16.5%	12.0%	1.5
		LS Mean		20.5%	4.6%	
EX-WT	12	Mean	**92.8**	**16.1%**	**2.1%**	**3.2**
		sd	33.5	10.3%	4.0%	1.6
		LS Mean		16.4%	2.1%	
EX-noWT	12	Mean	**93.0**	**21.7%**	**3.8%**	**2.9**
		sd	23.4	20.5%	7.7%	1.5
		LS Mean	-	22.6%	3.8%	-
		FB Main Effect	F(1,45)=0.435, p=0.51	F(1,44)=0.04, p=0.84	F(1,44)=0.09, p=0.76	F(1,45)=1.70, p=0.20
		WT Main Effect	F(1,45)=0.017, p=0.90	F(1,44)=1.31, p=0.26	F(1,44)=2.10, p=0.15	F(1,45)=0.07, p=0.80
		FB*WT Interaction	F(1,45)=0.025, p=0.88	F(1,44)=0.07, p=0.79	F(1,44)=0.45, p=0.51	F(1,45)=0.11, p=0.74

Table 7: Pre-test measures

4.5.2 Overall Results

As anticipated, the test of mathematical ability was a strong predictor of student performance (R^2 = 0.55, F (1, 48) =57.13, p < 0.001). Indeed, the math test was a better predictor of performance than the computer experience score (R^2 = 0.08, F(1, 48)=3.96, p=0.05). Consequently, the results presented here control for mathematical ability as a covariate. An analysis of pre-test scores revealed no statistically significant differences among experimental conditions (Table-7).

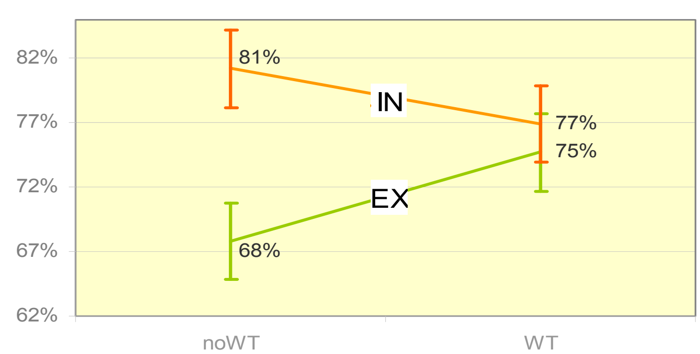


Figure 19: A main effect of Feedback (F(1,44)= 6.10, p <0.02) and a marginally significant Feedback by Walkthrough interaction (95% confidence intervals are displayed around math ability adjusted scores). All three treatment conditions do better than the expert no-walkthrough control condition.

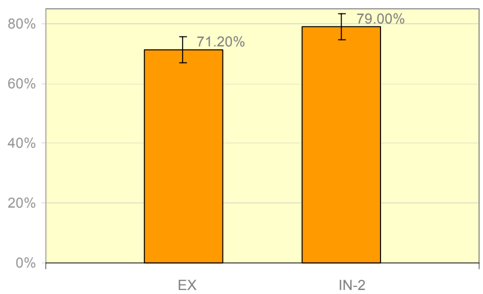
Table 8 summarizes overall student performance. An ANCOVA analysis of overall performance reveals a significant main effect for feedback (IN = 79%, EX = 71.20%, $F(1,44)= 6.10$, $p <0.02$) and marginally significant feedback by walkthrough interaction (IN-WT = 76.9%, IN-noWT = 81.2%, EX-WT=74.7% and EX-noWT = 67.8%; $F(1,44) = 3.25$, $p <0.079$) (see Figure-19 and Table-8). Students in all three treatment conditions did better than the EX-noWT control condition. However, the difference between the two walkthrough conditions was not statistically significant ($F(1,44)=0.18$, $p< 0.67$). Since the overall feedback by walkthrough interaction is only marginally significant, the remainder of the presentation of results will focus on the reliable main effect for feedback. The walkthrough manipulation is discussed further in Chapter 5.

4.5.3 Conceptual Understanding and Problem Solving Performance

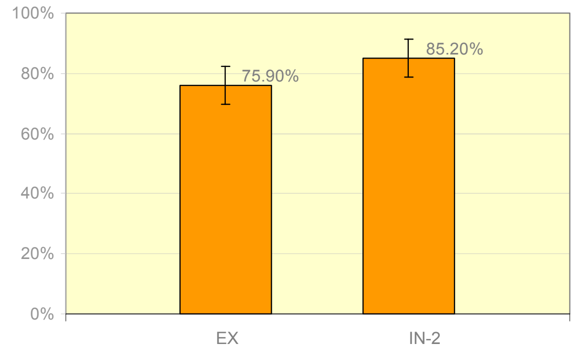
Separate repeated measures ANCOVA analyses on the problem solving and conceptual understanding components mirror the overall performance results in Table-8 and Figure-19. Student performance on all the problem solving pre and post tests (with the exception of the day 1 pretest) show that students in the intelligent novice condition performed significantly better than students in the expert condition (IN=85.20%, EX =75.90%, $F(1,44)= 4.24$, $p < 0.05$). Student performance on these tests is summarized in Table 9 and Table 10. Similarly, student performance in the conceptual understanding component of the pre and post test favors the intelligent novice condition (IN =72.90%, EX = 66.60%, $F(1,44)=4.06$, $p <0.05$). Performance on tests of conceptual understanding across three days is summarized in Table 11 and Table 12.

4.5.4 Transfer and Retention Performance

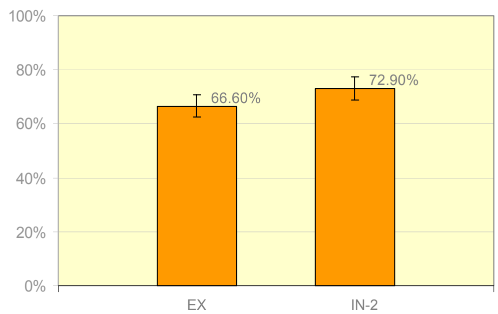
Students in the intelligent novice condition did significantly better on the transfer tasks than students in the expert condition (IN = 74.31%, EX = 59.79%, $F(1,44)=5.66$, $p<0.03$) (Table 13 and Table 14). Additionally, as Table-15 and Table-16 indicate, students in the intelligent novice condition performed significantly better on the Day-10 pre-test administered following an eight-day retention interval (IN = 81.20%, EX=72.50%, $F=4.07$, $p<0.05$).



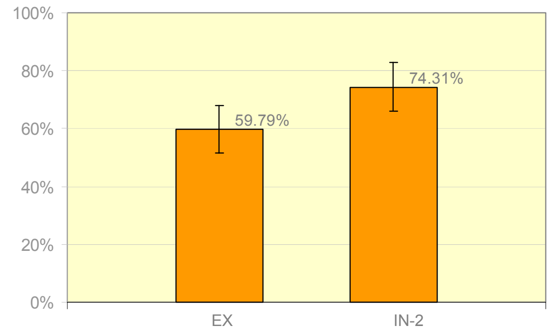
A Overall Performance (F(1,44)= 6.10, p <0.02)



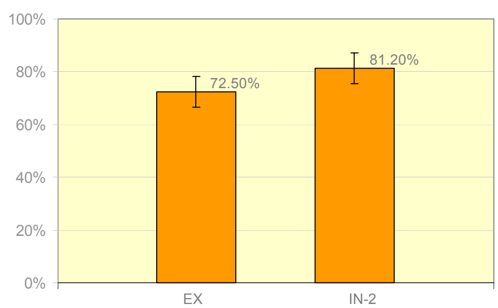
B Performance on Problem Solving Tests (F(1,44) = 4.23, p < .05)



C Performance on Pre and Post Tests of Conceptual Understanding. (F(1,44) =4.06, p < .05)



D Transfer Performance (F(1,44) = 5.662, p < .03)



E Retention Performance (F(1,44)=4.08, p < 0.05)

Figure 20: Summary of Results (Note: Error bars depict 95% confidence intervals)

Overall Performance

cond	n	Overall Performance on all Pre and Post Tests
IN-WT	12	**76.9%** (Mean: 79.0%, sd: 21.5%)
IN-noWT	13	**81.2%** (Mean: 83.5%, sd: 19.9%)
EX-WT	12	**74.7%** (Mean: 73.4%, sd: 22.7%)
EX-noWT	12	**67.8%** (Mean: 64.5%, sd: 24.8%, SE: 2.5%)
		Overall FB Main Effect F(1,44)= 6.10 p=0.02 Overall WT Main Effect F(1,44)= 0.19 p=0.67 Overall FB*WT Interaction F(1,44)= 3.25 p=0.08

Table 8: Overall Performance (averaging across all pre and post tests)

Problem Solving Performance

cond	n		Day1-Post	Day2-Post	Day 10-Pre (Retention)	Day10-Post (Relearning)	Overall
IN	25	Mean	**81.1%**	**90.4%**	**88.4%**	**91.8%**	87.9%
		sd	29.1%	15.4%%	22.7%	16.9%	21.8%
		LS Mean	77.6%	88.0%	86.0%	89.4%	85.2%
EX	24	Mean	**62.8%**	**76.5%**	**75.2%**	**77.9%**	73.1%
		sd	31.4%	25.0%	31.3%	28.6%	29.4%
		LS Mean	66.3%	79.0%	77.7%	80.4%	75.9%
FB Main Effect F(1, 44)=4.24, p<0.05							

Table 9: Performance across feedback conditions on problem solving tests (averaging across all pre/post tests)

cond	n		Day1-Post	Day2-Post	Day 10-Pre (Retention)	Day10-Post (Relearning)	Overall
IN-WT	12	Mean	**77.3%**	**91.2%**	**88.3%**	**90.8%**	86.9%
		sd	29.9%	14.6%	20.7%	19.8%	22.0%
		LS Mean	74.2%	89.0%	86.0%	88.6%	84.4%
IN-noWT	13	Mean	**84.6%**	**89.6%**	**88.5%**	**92.7%**	88.8%
		sd	29.2%	16.6%	25.2%	14.5%	21.8%
		LS Mean	81.1%	87.0%	85.9%	90.2%	86.1%
EX-WT	12	Mean	**70.6%**	**81.2%**	**81.2%**	**82.5%**	78.9%
		sd	29.9%	25.1%	30.8%	25.7%	27.5%
		LS Mean	72.6%	82.7%	82.7%	83.9%	80.5%
EX-noWT	12	Mean	**55.0%**	**71.7%**	**69.2%**	**73.3%**	67.3%
		sd	32.3%	25.0%	32.0%	31.7%	30.3%
		LS Mean	60.0%	75.3%	72.8%	76.9%	71.3%
Overall FB Main Effect F(1, 44)= 4.24 p<0.05 Overall WT Main Effect F(1, 44)= 0.72 p= 0.40 Overall FB*WT Interaction F(1, 44)= 1.45 p=0.23							

Table 10: Performance on problem solving tests (averaging across all 8 problem solving pre/post tests – with the exception of Day1 pre-test)

Performance on Test of Conceptual Understanding

cond	n		Day1-Post	Day2-Post	Day 10-Pre (Retention)	Day10-Post (Relearning)	Overall
IN	25	Mean	**67.7%**	**72.1%**	**78.6%**	**80.3%**	74.7%
		sd	18.5%	18.4%	17.3%	12.8%	17.4%
		LS Mean	65.9%	70.4%	76.4%	78.7%	72.9%
EX	24	Mean	**63.7%**	**64.9%**	**65.1%**	**65.8%**	64.9%
		sd	15.0%	13.8%	17.4%	20.1%	16.5%
		LS Mean	65.3%	66.6%	67.2%	67.4%	66.6%
FB Main Effect F(1, 44)=4.24, p<0.05							

Table 11: Performance on tests of conceptual understanding (averaging across all pre/post tests)

cond	n		Day1-Post	Day2-Post	Day 10-Pre (Retention)	Day10-Post (Relearning)	Overall
IN-WT	12	Mean	**59.5%**	**70.5%**	**74.1%**	**79.8%**	71.0%
		sd	17.4%	18.6%	18.8%	11.3%	17.9%
		LS Mean	58.1%	69.0%	72.2%	78.3%	69.4%
IN-noWT	13	Mean	**75.3%**	**73.6%**	**82.7%**	**80.8%**	78.1%
		sd	16.7%	18.9%	15.4%	14.4%	16.4%
		LS Mean	73.7%	71.9%	80.6%	79.1%	76.3%
EX-WT	12	Mean	**65.5%**	**64.9%**	**67.8%**	**73.8%**	68.0%
		sd	15.2%	13.8%	14.6%	16.6%	15.0%
		LS Mean	66.4%	65.8%	69.0%	74.7%	69.0%
EX-noWT	12	Mean	**61.9%**	**64.9%**	**62.5%**	**57.7%**	61.8%
		sd	15.3%	14.4%	20.2%	20.8%	17.5%
		LS Mean	64.2%	67.4%	65.5%	60.1%	64.3%
FB Main Effect F(1, 44)= 4.06 p=0.05 WT Main Effect F(1, 44)= 0.13 p=0.72 FB*WT Interaction F(1, 44)= 3.66 p=0.06							

Table 12: Performance on tests of conceptual understanding (averaging across all pre/post tests)

Transfer Performance

cond	n	Performance on Transfer Test
IN	25	**74.3%** (Mean: 78.5% , sd: 27.9%)
EX	24	**59.8%** (Mean: 55.3% , sd: 33.4%)
FB Main Effect F(1,44)=5.66, p=0.02		

Table 13: Performance on transfer task across feedback conditions

cond	n	Transfer Performance
IN-WT	12	**77.0%** (Mean: 81.0%, sd: 28.8%)
IN-noWT	13	**71.7%** (Mean: 76.2%, sd: 28.0%)
EX-WT	12	**61.9%** (Mean: 59.4%, sd: 35.6%)
EX-noWT	12	**57.6%** (Mean: 51.3%, sd: 32.1%)
FB Main Effect F(1, 44)= 5.66, p=0.02 WT Main Effect F(1, 44)= 0.64, p=p=0.42 FB*WT Interaction F(1, 44)= 0.01, p=0.93		

Table 14: Performance on tests of conceptual understanding (averaging across all pre/post tests)

Retention Performance

cond	n	Performance on Retention Tests
IN	25	**81.2%** (Mean: 83.5% , sd: 20.6%)
EX	24	**72.5%** (Mean: 70.2% , sd: 25.6%)
		FB Main Effect F(1,44)=4.075, p<0.05

Table 15: Performance on tests of conceptual understanding (averaging across all pre/post tests)

cond	n	Performance on Retention Tests (following an eight day retention interval)
IN-WT	12	**79.1%** (Mean: 81.2%, sd: 20.7%)
IN-noWT	13	**83.2%** (Mean: 85.6%, sd: 20.7%)
EX-WT	12	**75.8%** (Mean: 74.5%, sd: 24.5%)
EX-noWT	12	**69.2%** (Mean: 65.8%, sd: 26.4%)
		FB Main Effect F(1,44)=4.08, p<0.05 WT Main Effect F(1,44)=0.09, p=0.76 FB*WT Interaction F(1,44)=1.62 , p=0.21

Table 16: Performance on tests of conceptual understanding (averaging across all pre/post tests)

Problems Completed

cond	n	Number of Problems Completed Over 3 Days
IN	25	**105.2** (Mean: 113.4 , sd: 61.4).
EX	24	**99.5** (Mean: 91.5 , sd: 64.1)
		FB Main Effect F(1,44)=0.17, p=.68

Table 17: Performance on tests of conceptual understanding (averaging across all pre/post tests)

cond	n	Number of Problems Completed Over 3 Days
IN-WT	12	**92.7** (Mean: 99.9, sd: 51.3)
IN-noWT	13	**117.7** (Mean: 125.8, sd: 69.2)
EX-WT	12	**102.7** (Mean: 98.25, sd: 80.1)
EX-noWT	12	**96.3** (Mean: 84.8, sd: 45.5)
		FB Main Effect F(1,44)=0.17, p=0.68 WT Main Effect F(1,44)=0.46, p=0.50 FB*WT Interaction F(1,44)=1.32 , p=0.26

Table 18: Performance on tests of conceptual understanding (averaging across all pre/post tests)

Learning Curves

We examined online training data to determine if performance during training mirrored the outcome measures just presented. The online data would also indicate at what point in the learning process these differences emerge.

We examined the number of attempts required to solve training problems as a function of the opportunity to practice six production rules associated with generating a solution to the six types of problems represented in the tutor (see Figure-21). For example, if a student wrote an incorrect formula (1st attempt), then modified it incorrectly (2nd attempt), then succeeded in entering a correct formula (at the 3rd attempt), we would count that performance as taking three attempts at a particular opportunity to apply a particular production. Figure-21 plots the average number of attempts required to correctly solve each of the six types of problems with each practice opportunity.

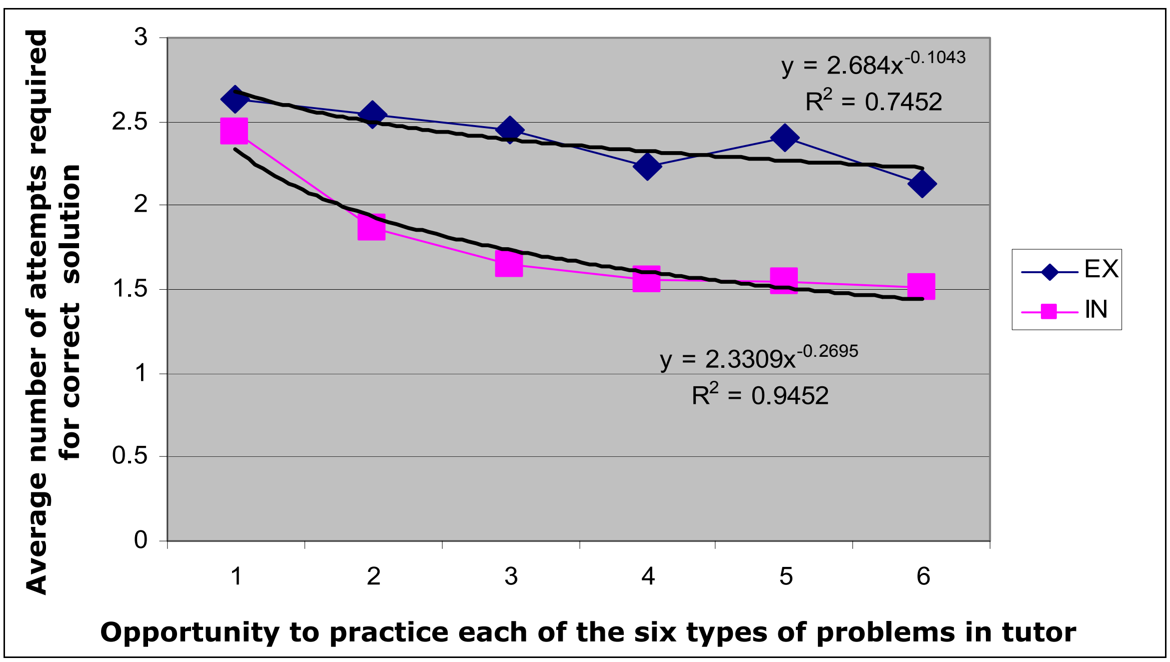


Figure 21: A comparison of learning curves associated with the two feedback condition. Students in both condition start at approximately the same level of performance. However, students in the intelligent novice condition learn at a faster rate.

Learning trends were estimated using best fitting power curves. A comparison of learning curves associated with the two tutorial conditions reveals that students in both groups start off performing at approximately the same level. However, over the course of successive attempts students in the intelligent novice condition learn at a faster rate.

We compared the difference in the number of attempts required between the first and second opportunity to practice the production rules among the two feedback conditions. A repeated measures ANCOVA showed a significant Feedback * Opportunity interaction ($F(1,43)=4.045$, $p = 0.05$). While students in both the expert and intelligent novice condition require approximately the same number of attempts to successfully solve the problem on the first opportunity of practice. Students in the intelligent novice condition do significantly better on the second opportunity to practice each of the production rules.

The learning curve analysis suggests that the opportunity to engage in the exercise of evaluative skill has a significant effect on student comprehension of cell referencing principles (discussed further in 4.6.1) . Furthermore, this analysis suggests that impact is most pronounced early in the learning process.

4.6 Discussion

4.6.1 Feedback

As discussed earlier, immediate feedback has been criticized on the grounds that it may prevent students from exercising skills that are important for performing tasks outside the training environment. These include error detection and error correction skills. However, as Corbett and Anderson (2001) have pointed out, merely delaying feedback may be necessary, but not sufficient to promote error detection and correction skills. Instead, they suggest, it may be necessary to provide direct feedback and support for these skills. Feedback based on an intelligent novice model does not simply provide students with an opportunity to engage in evaluative skills as delayed feedback would. Instead, an intelligent novice system explicitly models error detection and correction activities and supports students in the exercise of these skills.

Feedback based on an intelligent novice model provides a practical way for facilitating the exercise of evaluative skills in intelligent tutoring systems, while minimizing the potential for unproductive floundering. A comparison of student performance under the expert and intelligent novice conditions attests to the relative effectiveness of intelligent novice model feedback on a variety of different measures. During training, students in the intelligent novice condition learned at a faster rate. Intelligent novice condition students outperformed students in the expert condition on a variety of tasks – including: performance on isomorphs of training tasks, tests of conceptual understanding, transfer tasks, and retention tests following an eight day retention interval.

The analysis and results presented here suggest that the joint exercise of generative and evaluative skills can contribute to better learning outcomes than a focus on generative skills alone. Merrill, Reiser, Merrill, and Landes (1995) have theorized that a critical component of learning a new domain involves acquiring knowledge about the semantics of domain operators and their interactions. Productive engagement in reasoning about the causes and consequences of errors may provide students with a better model of domain operators. It is conceivable that the opportunity to observe the consequences of actions – whether successful in accomplishing a goal or not – may contribute to better declarative encodings of the effects of domain operators. This knowledge may be used interpretively to guide subsequent problem solving activity. The analysis presented here suggests that delaying feedback without explicitly engaging students in the exercise of evaluative skills will not produce as effective and transferable learning. This prediction can be tested by comparing a delayed feedback tutor with an intelligent novice tutor.

4.6.2 Example Walkthroughs

While learning outcomes associated with the expert walkthrough condition were not as pronounced as the results associated with the expert walkthrough in Study-1, the trend observed in Study-2 is in the same direction (see Figure-13). The interaction depicted in Figure-13, suggests that intelligent novice feedback and example walkthroughs provide complementary means to improve learning outcomes associated with existing expert model tutors. The results just presented suggest that the joint combination of example walkthroughs and intelligent novice feedback, add little to the benefits each would independently provide.

Chapter 5

Conclusion

5.1 Feedback

The research reported here presents experimental comparisons of feedback based on two cognitive models. These comparisons were carried out in the context of a spreadsheet tutor. One version of the tutor provided feedback on the basis of an expert model. The other version presented feedback on the basis of a model of an intelligent novice.

Study-1 showed that learning outcomes associated with the intelligent novice condition were not significantly different from the expert-no Walkthrough, control condition. It was hypothesized that the scaffolding associated with the exercise of error correction skills in the intelligent novice condition imposed a high working memory load on learners. This may have been the result of the fact that error correction activities were spread out over twenty steps — in a context detached from the original problem. Typos and other slips during error correction served to prolong the overall problem solving process and induce confusion among many learners.

The tutor used in Study-2 addressed some of the design problems identified with the intelligent novice tutor in Study-1:

- Error correction steps in the redesigned tutor were reduced to two.

- Students using the redesigned intelligent novice system reasoned about the causes and consequences of errors in the original problem solving context – as opposed to the detached context of the discrepancy table as required by the original intelligent novice tutor.
- Instead of reasoning about the causes and consequences of errors by comparing formulas (two abstract symbol strings in the discrepancy table), visual cues served to help students note discrepancies between the goal and actual formulas.
- While error correction steps in the tutor used in Study-1 required considerable typing – with the potential for introducing confusing typos – the intelligent novice tutor used in Study-2 reduced opportunities for errors by guiding students through error correction activities using multiple choice prompts.

Students using the intelligent novice tutor in Study-2 performed significantly better than students in the expert condition on a variety of measures: including, performance on close isomorphs of training tasks, tests of conceptual understanding, transfer tasks, and retention tests. Furthermore, an analysis of errors during training suggests that students in the intelligent novice condition start off requiring the same number of attempts to solve problems as students in the expert model feedback condition. However, over the course of training, students in the intelligent novice condition learn at a faster rate.

5.1.1 Acquiring the semantics of domain operators

The analysis and results presented here suggest that the joint exercise of generative and evaluative skills can contribute to better learning outcomes than a focus on generative skills alone. These results may be a result of the nature of problem solving activities supported by the intelligent novice model.

Newell and Simon (1972) have characterized problem solving as a search through a space of knowledge states. As Simon and Lea (1974) have noted, search through such a space is highly selective and is guided by the information made available at each knowledge state. Once a problem solver has visited certain number of points in a problem state, he or she can determine a direction to continue searching in one of two ways: by selecting a previously encountered knowledge state from which to continue the search or by selecting a particular operator that will allow the problem solver to reach a new knowledge state. Newell and Simon have identified means-end-analysis as a mechanism used by problem solvers to guide these selections. Problem solvers compare a given problem state with a goal state to discover one or more differences between them and pick operators from previous experience that are known to eliminate these differences. By applying operators selected in such a manner, problem solvers transition through problem states to arrive at a solution.

While the inappropriate application of an operator may be of little help with respect to an immediate problem solving objective, the opportunity to observe the effect of applying the inappropriate operator might prove to be helpful in later problem solving contexts where that operator and its associated effect may be relevant. With intelligent novice feedback, a student has the opportunity to learn both about an operator that might be immediately relevant to solving a problem, and also domain operators that might be relevant in subsequent problem solving contexts.

5.1.2 Role of the instructional interface

The exercise of error detection and correction skills required an environment where the discrepancy between desired and actual outcomes could be made salient to the student. The spreadsheet environment is rich in *internal feedback*. The action of copying and pasting a formula produces easily discernable consequences in the learning environment. Users can use this information to assess the effectiveness of their action and take remedial action if necessary. As such, the spreadsheet environment shares several attributes with environments that place the learner in the role of a diagnostician. In the words of Nathan (1998, pg 138), such systems may "reflect back to the learner observable and meaningful ramifications of the learners actions in such a way that the learner can use her prior knowledge to identify solution errors, re-examine prior misconceptions, and propose and test hypotheses about the causes of errors. "

Internal feedback may be absent or difficult to interpret in many domains. In such cases special attention may need to be placed on the design of learning environments. A strategy that has been adopted to deal with these deficiencies has been to link the use of formal expressions that a learner might be attempting to master with an animation or simulation. Students write mathematical or programming expressions that serve to control elements of a visual animation. The LOGO programming language (Papert, 1980) and the ANIMATE algebra learning environment (Nathan, 1998) represent instances of such an approach. Textual output of the intermediate and final results of a computer program or manipulation of mathematical expressions might serve a similar end. In order to facilitate better error detection and correction activities in the spreadsheet tutor described here, students were asked to create formulas that would multiply numerical values whose product would be easy for most students to predict and assess – (e.g. 2*5, 1500*10, 50*2).

5.2 Example Walkthroughs

The two studies reported here also evaluated the effectiveness of example walkthroughs. In both studies, students in the walkthrough conditions were actively engaged in the study of examples. Students were lead make a series of inferences associated with the selection of problem solving operators that may have been implicit in video and text examples.

In Study-1, the only condition that performed significantly better than the expert-no-walkthrough control condition was the expert-walkthrough condition. Walkthroughs did not have much of an impact on the intelligent novice condition of Study-1. We hypothesize that the overall working memory burden experienced by students working with the discrepancy table during the exercise of error detection and correction activities (both during walkthroughs and during procedural practice) is likely to have eliminated any likely benefit of using example walkthroughs for students working with the intelligent novice version of the tutor.

Study-2 provided another opportunity to evaluate the efficacy of example walkthroughs. Once again, the worst outcomes were associated with the expert no-walkthrough condition. An analysis of results showed a marginally significant feedback by walkthrough interaction. The pattern of the interaction was consistent with a *terminative interaction*[3]. Both walkthroughs and intelligent novice feedback appeared to have a positive impact on student performance. However, the combination of the two added no benefit over the effect of each alone. Two aspects of the outcome observed in Study-2 are worth noting. First, outcomes associated with the expert-walkthrough were not as pronounced as the results observed in Study-1. Second, Walkthroughs had virtually no impact among students using the intelligent novice tutor.

It is not possible to determine exactly why students in the expert walkthrough condition in Study-1 outperformed expert walkthrough students in Study-2. It is possible that modifications to the walkthroughs in the expert condition prior to Study-2 may have contributed to the observed outcome.

Prior to Study-2, the expert model example walkthrough was modified to eliminate the opportunity to observe the consequences of copying and pasting a formula with relative references in a problem that required absolute references. In other words, the expert walkthrough condition in Study-1 had an element of the intelligent novice condition — that is, an opportunity to experience error detection and correction. In order to make the walkthrough in the expert condition more consistent with the underlying model of desired performance, the example walkthrough in the expert condition was modified to focus exclusively on the steps involved in generating a solution. It is plausible that the opportunity to observe the consequences of copying and pasting a formula without the appropriate absolute references may have given students in the Study-1 expert-walkthrough condition a better understanding of cell referencing concepts prior to practice with the

---

[3] A **terminative interaction** is one in which two or more variables are clearly effective, but when combined their effect is not increased over that of either of the variables alone. (Neale and Liebert, 1986)

tutor. The possibility that the opportunity to observe the consequences of copying and pasting incorrect formulas may have contributed superior learning outcomes among expert walkthrough students in Study-1 will have to be examined closely in future studies.

When considered together, results from the two studies point to example walkthroughs as a promising way to improve on learning outcomes associated with expert model tutors. In Study-1, the expert-walkthrough condition was the only group to significantly outperform the expert-noWalkthrough control condition. While the outcomes associated with the expert-walkthrough condition in Study-2 were not as pronounced as in Study-1, all conditions did better than the expert-no-Walkthrough control condition. Considering their relative ease of implementation, example walkthroughs may provide an economical and efficient way to improve on learning outcomes associated with intelligent tutoring systems.

5.3 Future Work

The following issues need closer examination in future studies:

- Do the pedagogical benefits associated with the intelligent novice condition stem from the opportunity to observe the consequences of an error? Or, are these benefits tied to engaging in error detection and correction activities? These questions could be answered by comparing the intelligent novice tutor used in Study-2 with a tutor that simply allows students to see the consequences of pasting an incorrect formula. While the proposed tutor would allow students to see the consequences of their errors, the emphasis in the tutor would still be on generative skills – students would not be guided through the process of analyzing errors.

- The intelligent novice tutor might also be compared to a delayed feedback tutor. Based on the theoretical analysis presented earlier, one would predict that the intelligent novice tutor would produce superior learning outcomes by productively engaging students in error detection and correction activities. As Corbett and Anderson have noted, delay may be necessary, but not sufficient condition for students to productively engage in self monitoring skills.

- Future work could try to determine whether the opportunity to observe the consequence of copying and pasting incorrect formulas during example walkthroughs in the expert-walkthrough condition in Study-1 may have contributed to the better learning outcomes among expert-walkthrough students than their counterparts in Study-2.

- It remains to be determined whether the benefits of example walkthroughs and intelligent novice feedback may be realizable in other academic domains — particularly in domains, such as algebra and computer programming that may be deficient in internal feedback.

- Error detection and correction activities are a relatively straight forward matter in the spreadsheet cell referencing domain. As soon as a formula is entered, it can be tested quite easily by copying and pasting the formula across relevant cells. Similarly, in the spreadsheet tutor, identifying the source of the error involves scrutinizing a fairly compact formula. However, in many domains, the solution to a problem may require a sequence of several steps. As a result identifying the source of an error may be considerably more difficult. Future work may examine ways to support the exercise of error detection and correction skills in such domains.

5.4 Contributions of this thesis

- Reviews and summarizes research on the timing of feedback as it relates to intelligent tutoring system design.

- Presents a novel theoretical perspective that offers the potential for jointly realizing many of the benefits offered by immediate and delayed feedback in intelligent tutoring systems.

- Provides empirical support for the theoretical perspective just mentioned. Students receiving feedback on the basis of a cognitive model of an intelligent novice (representing error detection, error correction, and solution generation skills) demonstrate better learning than tutors that embody expert models. Most intelligent tutors are based an expert model (emphasizing error free and efficient task performance). Expert model tutors intervene as soon as a student deviates from the solution path.

- Introduces example walkthroughs, a technique for helping students attend to, and accurately encode operator selection inferences implicit in problem examples. Example walkthroughs have shown promise in conjunction with expert model tutors. Provides a cost effective way to improve performance associated with expert model tutors.

- Demonstrates the application of the theoretical ideas outlined in the thesis in the context of a model tracing tutor aimed at teaching students spreadsheet cell referencing concepts – an area that has been shown to pose difficulties for experienced and novice spreadsheet users.

References

Anderson, J. R. (1993). Rules of the mind. L. Erlbaum Associates:Hillsdale, N.J.

Anderson, J.R., Conrad, F., and Corbett, A. (1989). Skill acquisition and the Lisp Tutor. Cognitive Science, 13, 467-505.

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

Baxter, I. & Oatley, K. (1991). Measuring the learnability of spreadsheets in inexperienced users and those with previous spreadsheet experience. Behaviour & Information Technology 10(6): 475-490

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

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

Brown, P. & Gould, J. (1987). An Experimental Study of People Creating Spreadsheets, ACM Transactions on Office Information Systems 5 (3), 258-272,

Bruer, J. (1993). The Mind’s Journey from Novice to Expert. American Educator, 6-46

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

Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. Cognition and Instruction, 8(4), 293-332

Chandler, P., & Sweller, J. (1992). The split-attention effect as a factor in the design of instruction. British Journal of Psychology, 62, 233-246

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.

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

Corbett, A. T., Anderson, J. R. (2001) Locus of feedback control in computer-based tutoring: impact on learning rate, achievement and attitudes. Proceedings of CHI 2002, Human Factors in Computing Systems (March 31 - April 5, 2001, Seattle, WA, USA), ACM, 2001 245-252

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

Doyle, J. R. (1990). Naive users and the Lotus interface: a field study. Behaviour & Information Technology 9(1): 81-89

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

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.

Jones, R. M., Fleischman, E. S. (2001). Cascade Explains and Informs the Utility of Fading Examples to Problems. Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society (pp. 459-464). Mahwah, NJ: Lawrence Erlbaum

Judd, C. H. (1908). The relation of special training to general intelligence. Educational Review, 36, 28-42.

Klahr, D. and Carver, S.M. (1988). Cognitive objectives in a LOGO debugging curriculum: Instruction, learning, and transfer. Cognitive Psychology, 20, 362-404

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

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

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

Lovett, M. C., & Greenhouse, J. B. (2000). Applying cognitive theory to statistics instruction. The American Statistician, 54, 196-206.

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.

Nardi, B. (1993). A Small Matter of Programming: Perspectives on End User Computing. Cambridge. Cambridge: MIT Press.

Nathan, M. J. (1998). Knowledge and situational feedback in a learning environment foralgebra story problem solving. Interactive Learning Environment5, 161-180

Neale, J.M. & Liebert, R.M. (1986). Science and Behavior: An Introduction to Methods of Research (3rd edition). Englewood Cliffs, NJ: Prentice-Hall.

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

Palincsar, A.S., & Brown, A.L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. Cognition and Instruction, 1, 117-175.

Pirolli, P.L., & Anderson, J.R., (1985). The role of learning from examples in the acquisition of recursive programming skills. | Canadian Journal of Psychology| , 39, 2, pp.240-72

Reif, F. & Allen, S. (1992). Cognition for interpreting scientific concepts: A study of acceleration. Cognition and Instruction, 9 1-44

Reif, F., Scott, L. A. (1999) Teaching scientific thinking skills: Students and computers coaching each other. American Journal of Physics, volume 67, pp. 819-831

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: Lawrence Erlbaum.

Rittle-Johnson, B., Siegler, R.S. & Alibali, M.W. (2001). Developing conceptual understanding and procedural skill in mathematics: An iterative process. Journal of Educational Psychology, 93, 346-362

Schoenfeld, A. (1987). What's all the fuss about metacognition? In A. Schoenfeld (Ed.), Cognitive science and mathematics education (pp. 189-215). Hillsdale, NJ: Lawrence Erlbaum Associates

Schmidt, R.A., Bjork, R.A. (1992). New conceptualizations of practice: common principles in three paradigms suggest new concepts for training. Psychological Science, 3 (4), 207-217.

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

Schommer, M. (1993). Epistemological development and academic performance among secondary students. Journal of Educational Psychology, 85, 1-6

Schooler, L. J. & Anderson, J. R. (1990). The disruptive potential of immediate feedback. Proceedings of the Twelfth Annual Conference of the Cognitive ScienceSociety, 702-708, Cambridge, MA

Singley, M. K. & Anderson, J. R. (1989). Transfer of Cognitive Skill. Cambridge, MA: Harvard University Press

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

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. Cognition and Instruction, 2(1), 59-89

Ward, M. & Sweller, J. (1990). Structuring effective worked examples. Cognition and Instruction, 1990, 7(1), 1-39.

Zhu, X. & Simon, H. A. (1987). Learning Mathematics from Examples and by Doing. Cognition and Instruction, 4, 137-166

Appendix 1: Expert Model English Rules

P1(a) absolute-column-reference

if =paste-cells lie in different columns

and =variable1 lies in a single column

and =variable2 lies in different columns

=>

Add an absolute reference ahead of the column reference for =variable1

and set goal ready-to-copy

P1(b) absolute-row-reference

if =paste-cells lie in different rows

and =variable1 lies in a single row

and =variable2 lies in different rows

=>

Add an absolute reference ahead of the row reference for =variable1

and set goal ready-to-copy

P1(c) relative-reference-case1

if =paste-cells lie in different rows

and =variable1 lies in different rows

and =variable2 lies in different rows

=>

omit use of any absolute references

and set goal ready-to-copy

P1(d) relative-reference-case2

if =paste-cells lie in different columns

and =variable1 lies in different columns

and =variable2 lies in different columns

=>

omit use of any absolute references

and set goal ready-to-copy

P1(e) constant-value

if =paste-cells lie in different rows and columns

and =variable1 lies in a single row and column

and =variable2 lies in different rows and columns

=>

add absolute references ahead of the row and column reference of =variable1

and set goal ready-to-copy

P1(f) constant-value

if =paste-cells lie in different rows and columns

and =variable1 lies different colimns

and =variable2 lies in different rows

=>

add absolute references ahead of the row of =variable1 and column reference of =variable2

and set goal ready-to-copy

P2 copy-formula

if the goal is ready-to-copy

=>

copy the value in formula-entry-cell

and set-goal ready-to-paste

P3 paste-formula

if the goal is ready-to-paste then

=>

select =paste-cells

and paste

set goal-complete

P4 problem-done

if the goal is goal-complete

=>

click on 'done button'

Appendix 2: Intelligent Novice Model English Rules

P1(a) absolute-column-reference

if =paste-cells lie in different columns

and =variable1 lies in a single column

and =variable2 lies in different columns

=>

Add an absolute reference ahead of the column reference for =variable1

and set goal ready-to-copy

P1(b) absolute-row-reference

if =paste-cells lie in different rows

and =variable1 lies in a single row

and =variable2 lies in different rows

=>

Add an absolute reference ahead of the row reference for =variable1

and set goal ready-to-copy

P1(c) relative-reference-case1

if =paste-cells lie in different rows

and =variable1 lies in different rows

and =variable2 lies in different rows

=>

omit use of any absolute references

and set goal ready-to-copy

P1(d) relative-reference-case2

if =paste-cells lie in different columns

and =variable1 lies in different columns

and =variable2 lies in different columns

=>

omit use of any absolute references

and set goal ready-to-copy

P1(e) constant-value

if =paste-cells lie in different rows and columns

and =variable1 lies in a single row and column

and =variable2 lies in different rows and columns

=>

add absolute references ahead of the row and column reference of =variable1

and set goal ready-to-copy

P1(f) constant-value

if =paste-cells lie in different rows and columns

and =variable1 lies different columns

and =variable2 lies in different rows

=>

add absolute references ahead of the row of =variable1 and column reference of =variable2

and set goal ready-to-copy

P2 copy-formula

if the goal is ready-to-copy

=>

copy the value in formula-entry-cell

and set goal ready-to-paste

P3 paste-formula

if the goal is ready-to-paste then

=>

select =paste-cells

and paste

set goal check-accuracy

EVALUATIVE SKILLS

p5 check-accuracy

if the actual value in a =paste-cell does not equal the anticipated value

=>

examine formula underlying cell with discrepancy

set goal examine-and-correct-underlying formula

p6 remove-incorrect-dollar-sign

if goal examine-and-correct-underlying-formula

and =pastecell has a reference that has not changed when it should have

=>

remove corresponding '$' sign from original formula

p7 add-dollar-sign-at-incorrectly-omitted-location

if goal examine-and-correct-underlying-formula

and pastecell has a reference that has changed when it not should have

=>

add '$' sign to corresponding reference in original formula

p8 all references checked

if goal examine-and-correct-underlying-formula

and unanticipated references changes have been corrected

=>

set goal ready-to-copy

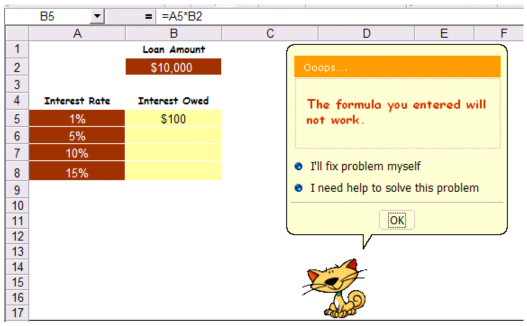
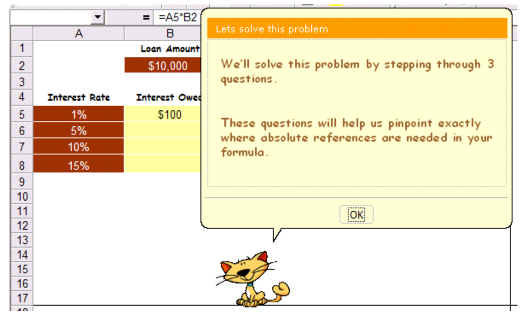
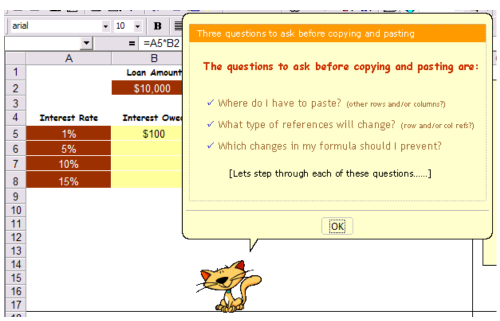
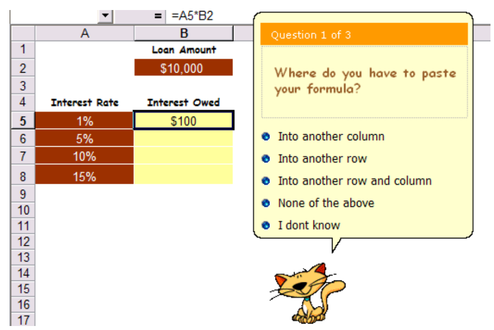
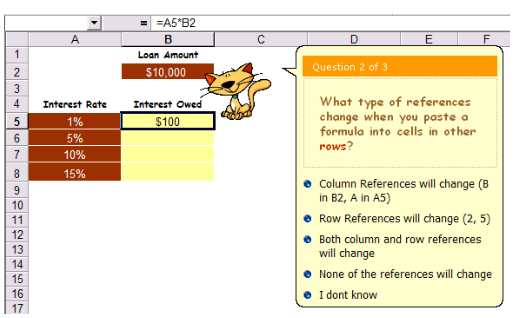
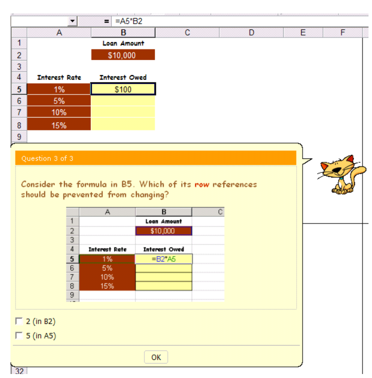
P4 problem-done

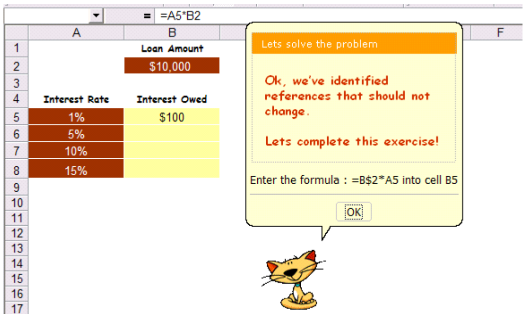
if the goal is goal-complete

=>

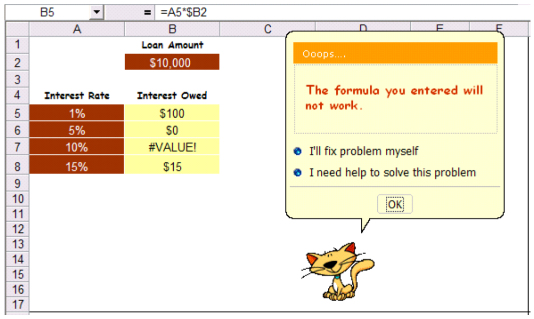
click on 'done button'

Appendix 3: Screen Shots of Expert Model Feedback in Study-1

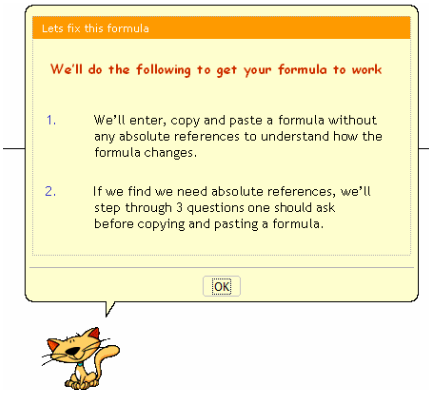
	
1. System intervenes as soon as a student enters an incorrect formula. Student has choice of fixing formula on one's own or doing so with help.	**2. If help is sought, the student is stepped through a three step process to generate a solution**
	
3. The three step process is outlined.	**4. The first step asks the student to consider the direction in which the formula is to be pasted.**
	
5. In the second step, the student is asked to consider the type of reference that will change.	**6. Student is shown references in the formula that will change. The system asks the student to select references that must remain fixed.**

		
7. The student is prompted to enter a formula that will hold the reference identified in 6 fixed.		

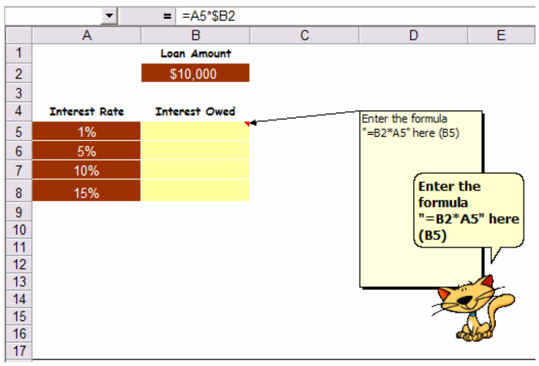
Appendix 4: Screen Shots of Intelligent Novice Model Feedback in Study-1



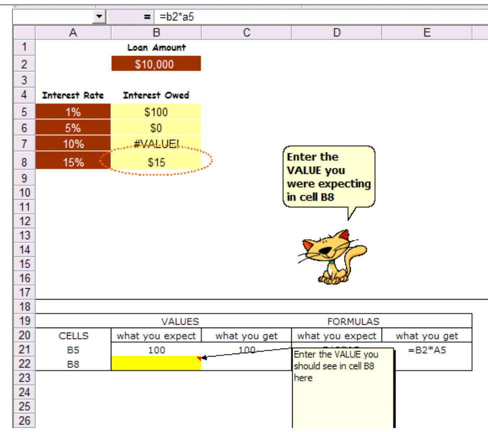
1. Student makes an error during repair attempt. Tutor notifies student and offers help.



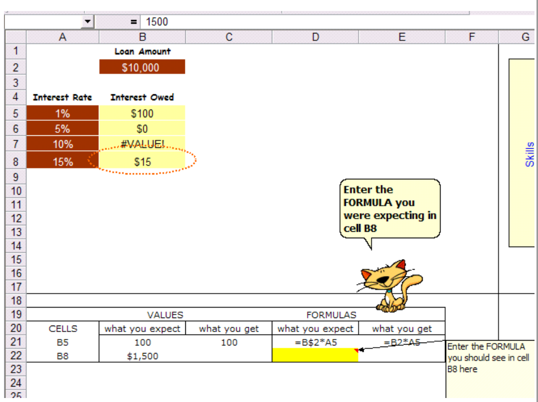
2. Tutor provides overview of error detection and correction process.



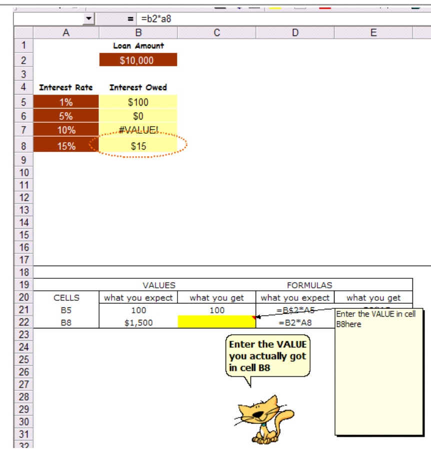
3. Student is prompted to enter, copy, and paste a formula with no absolute references to observe consequences.



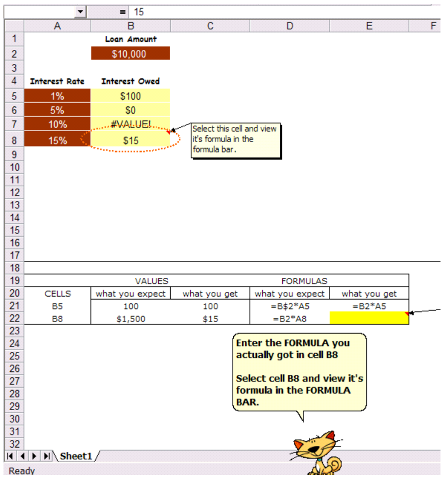
4. System picks a cell that was pasted into for close examination. The student is asked to enter the *value* that should have resulted in the selected cell had the formula worked.



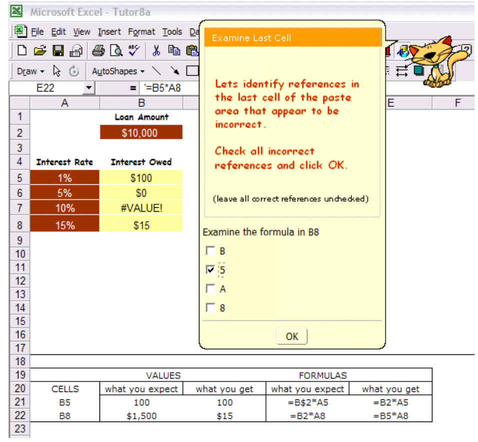
5. Student is asked to enter the formula that should have resulted in the selected cell had the formula worked.



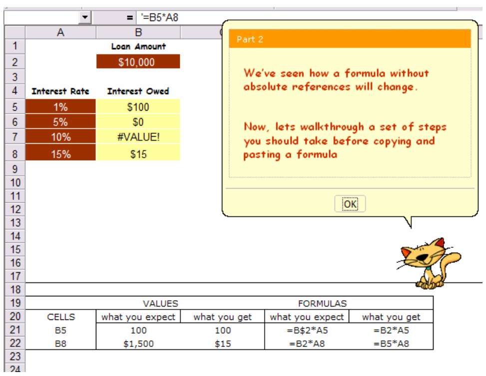
6. Student is asked to enter the value actually produced in the focus cell.



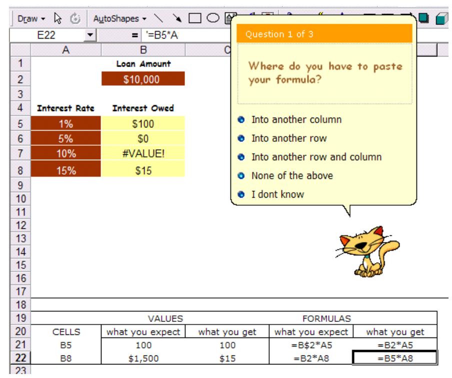
7. Student is asked to enter the *formula* actually produced in the focus cell.



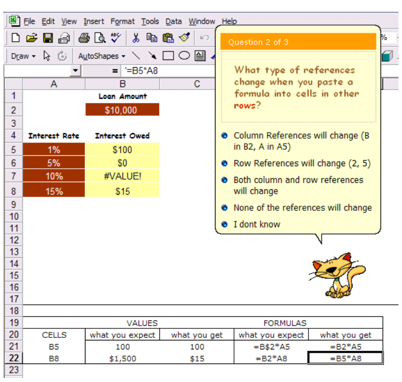
8. Student is asked to identify the incorrect reference in formula underlying the focus cell.



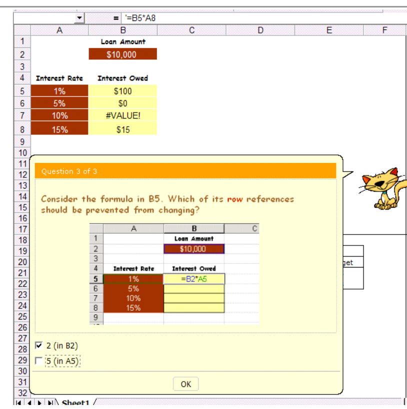
9. Having detected the source of an error, student is stepped through the process of generating a solution.



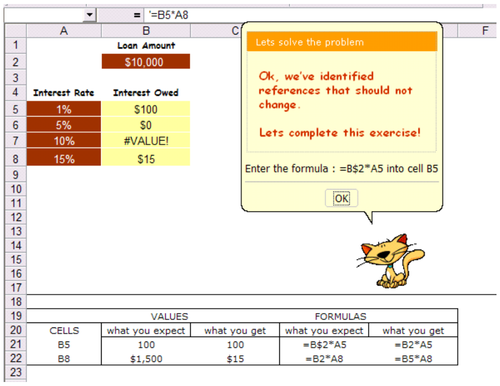
10. Student is asked to identify the direction in which the formula is to be copied and pasted



11. Student is asked to identify the types of references that will be affected when copied.

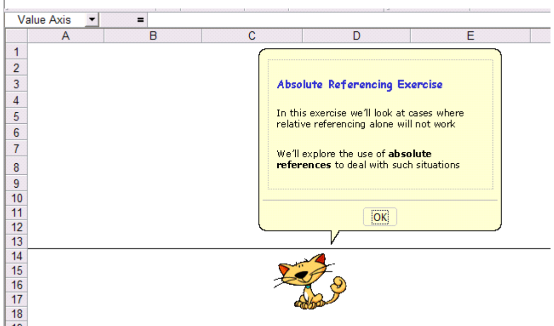


12. Student is shown references in the formula that will change. The system asks the student to select references that must remain fixed.

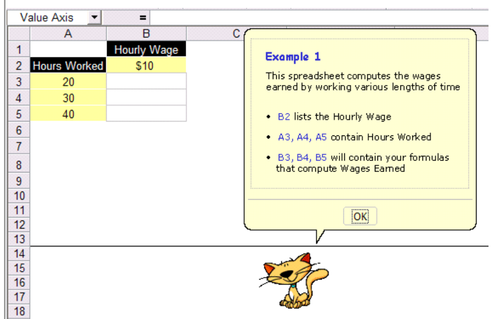


13. Student is asked prompted to enter the correct formula.

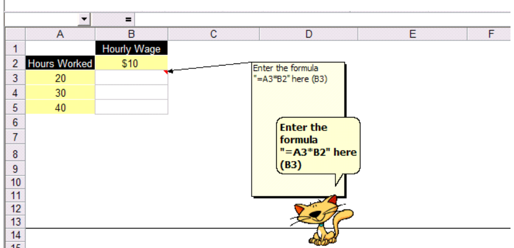
Appendix 5: Screen Shots of Expert Model Walkthrough in Study-1



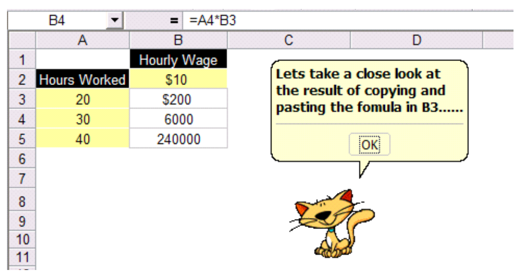
1. Introductory balloon in the Expert version of the example walkthrough



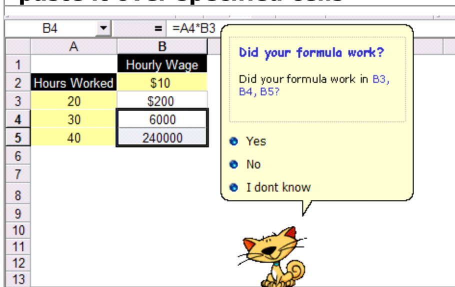
2. Overview of the problem that will be solved



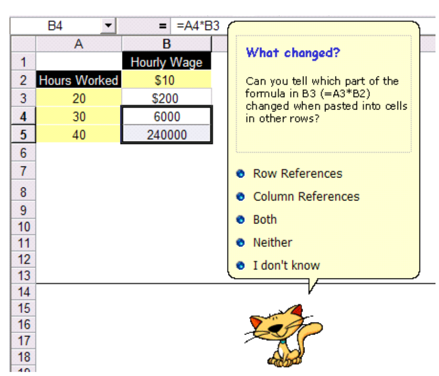
3. Student is asked to enter a formula without absolute references and copy and paste it over specified cells



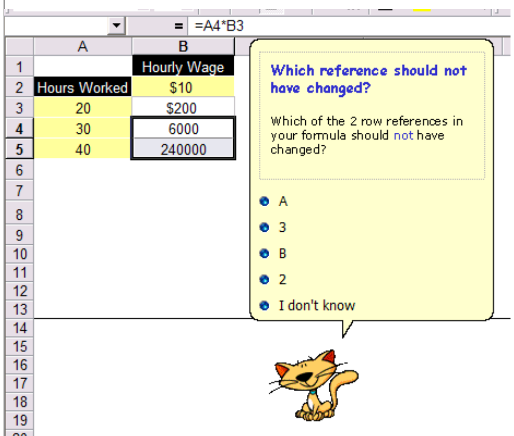
4. Student prompted to consider result of copy and paste in the next few steps



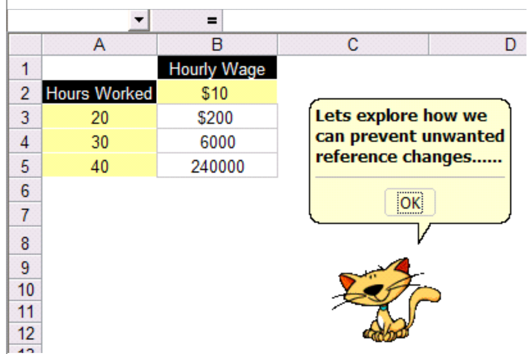
5. Student asked to indicate whether the formula worked appropriately.



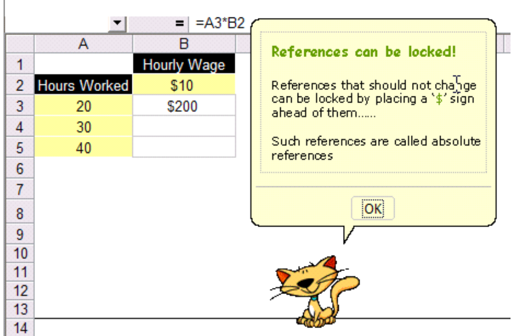
6. Student asked to indicate the type of reference that changed



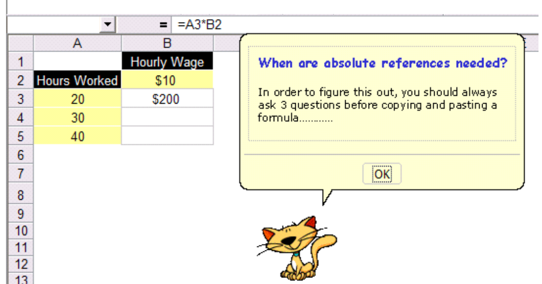
7. Student is asked to identify the reference in the formula that should not have changed



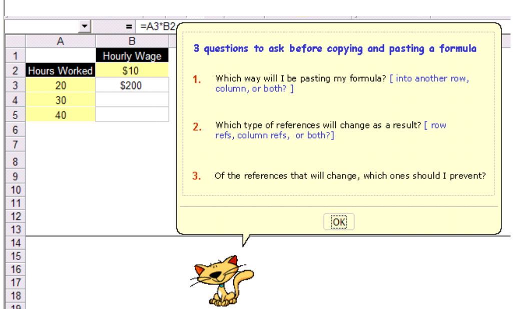
8. Focus now changes to generating a solution



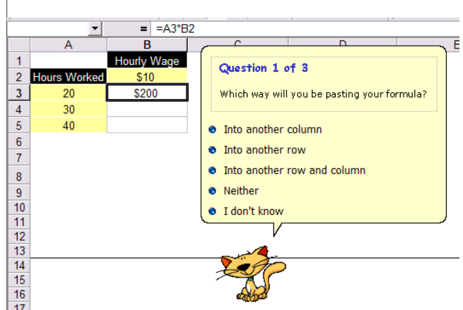
9. Student reminded about the absolute reference



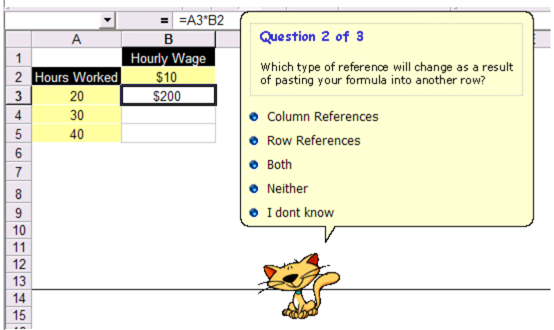
10. Student introduced to the solution generation process



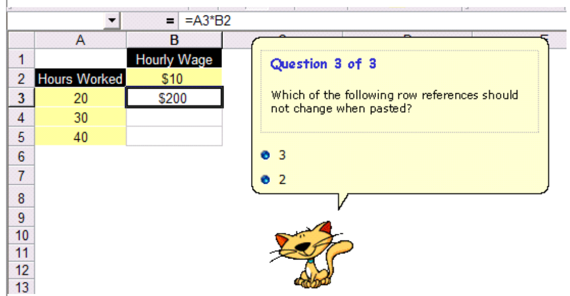
11. Student is shown an overview of a three step process that will allow the student to generate a solution to the problem.



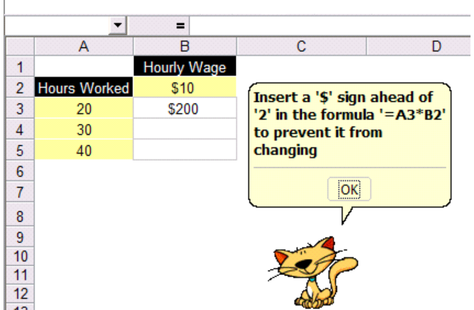
12. The first step asks the student to consider the direction in which the formula is to be pasted.



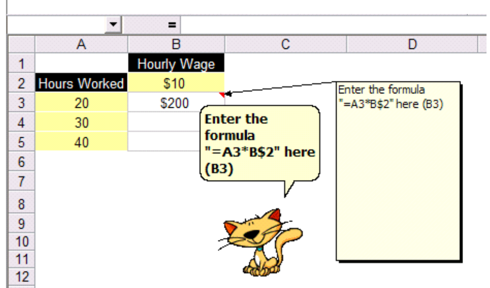
13. **In the second step, the student is asked to consider the type of reference that will change.**



14. **Student is shown references in the formula that will change. The system asks the student to select references that must remain fixed.**

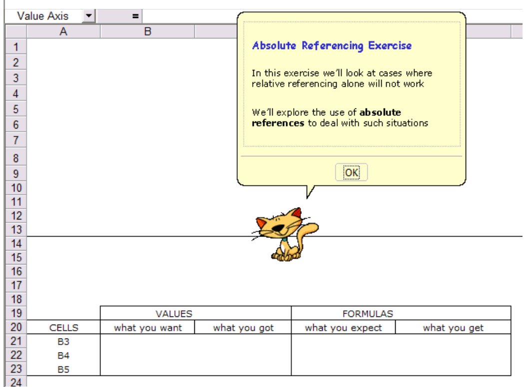


15. **Student is told that the reference identified in 14 can be held fixed with a '$' sign.**

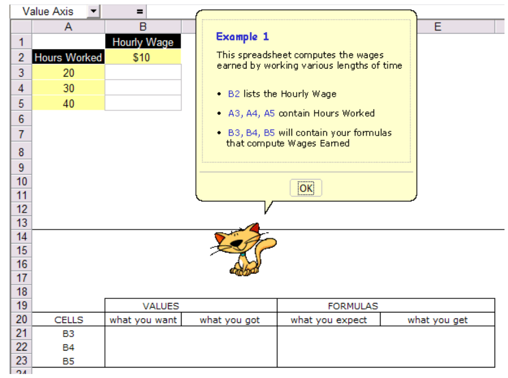


16. **Student is prompted to enter the solution. The student is subsequently asked to copy and paste the formula.**

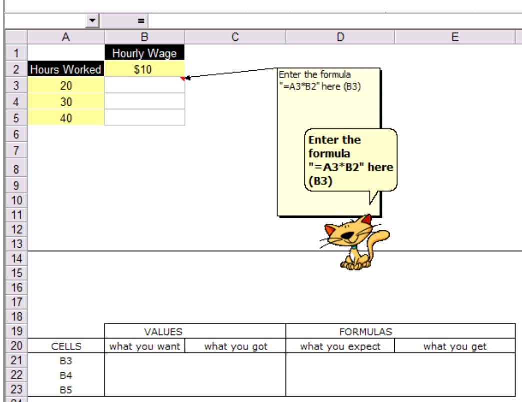
Appendix 6: Screen Shots of Intelligent Novice Model Walkthrough in Study-1



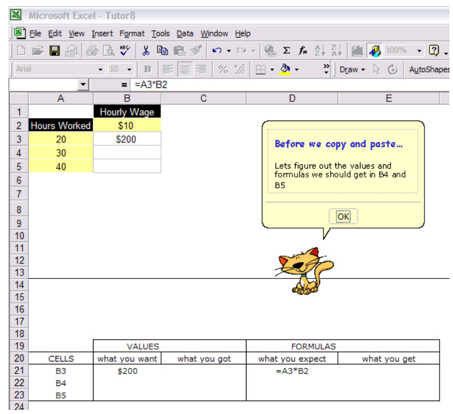
1. **Introduction to the walkthrough**



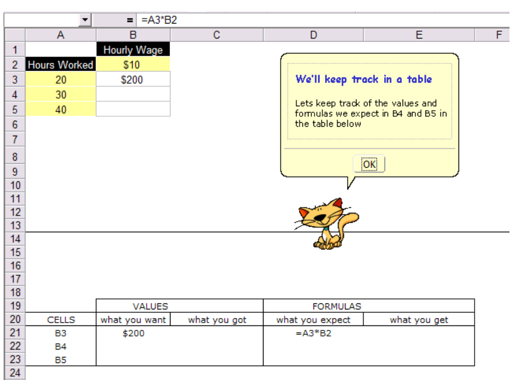
2. **Student is given a description of the problem to be solved.**



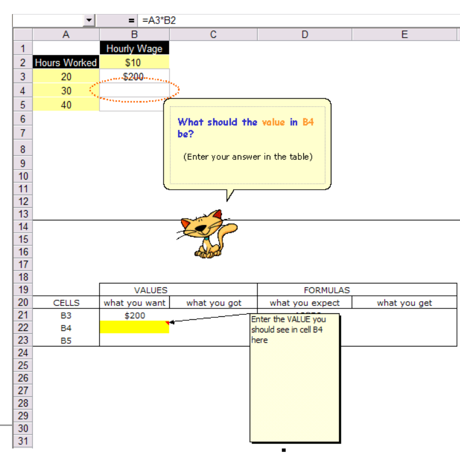
3. **Student is asked to enter a formula without absolute references**



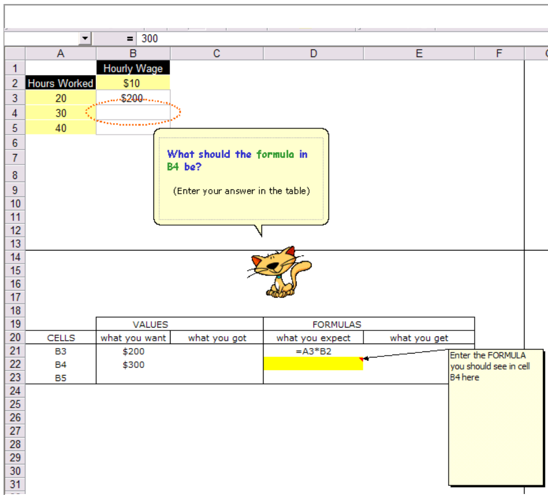
4. **Student will be prompted to predict values and formulas that will result from pasting the formula**

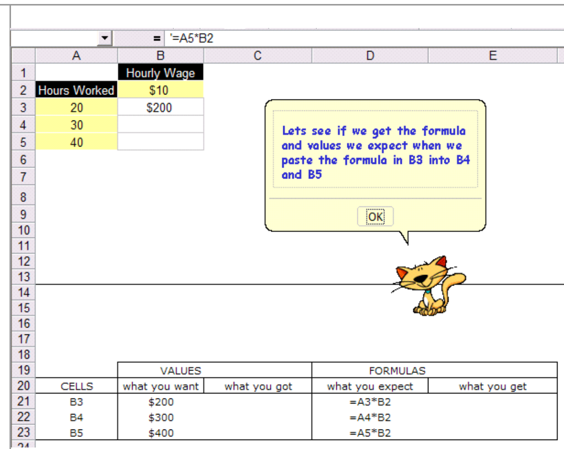


5. **Student is introduced to the discrepancy table where predicted and actual values and formulas will be compared.**



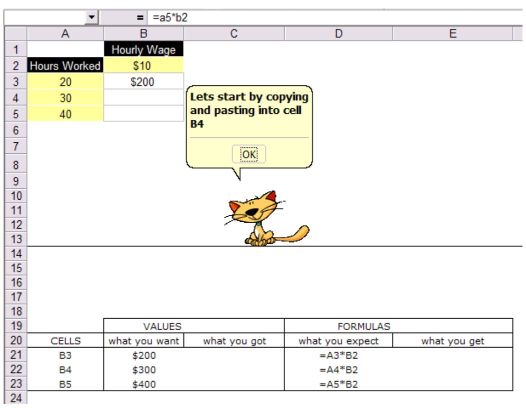
hat should result if the formula were to work correctly.

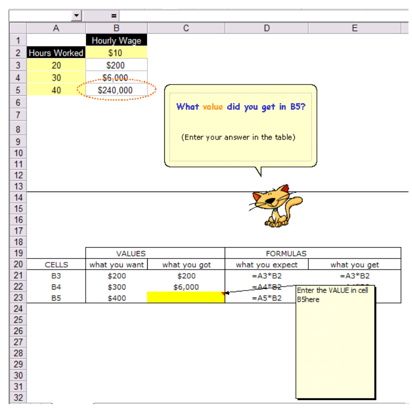




7. Student is prompted to specify formulas that should result in the paste locations if the formula were to work

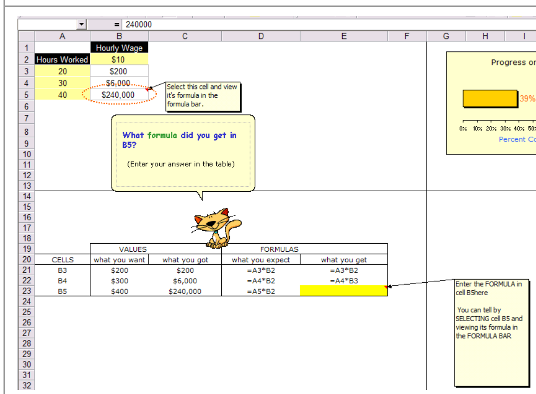
8. Student will judge the actual outcome of copying and pasting the formula in subsequent steps.

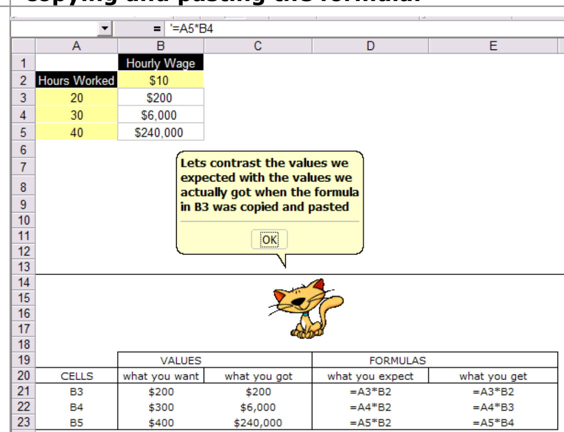




9. Student is prompted to copy and paste the formula one cell at a time

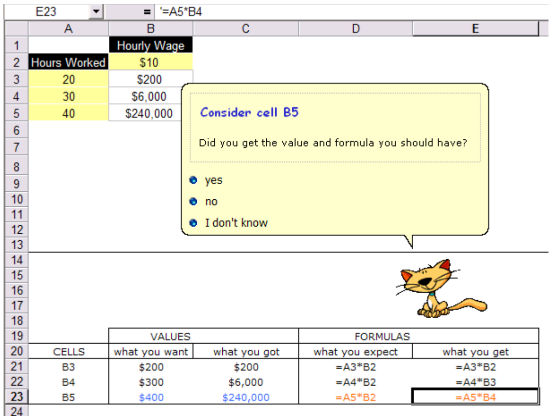
10. After each paste, student is asked to enter the value that actually results from copying and pasting the formula.



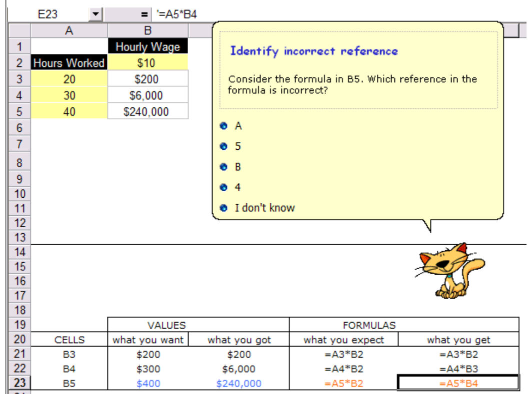


11. After each paste, student is asked to enter the formula that resulted from the paste operation.

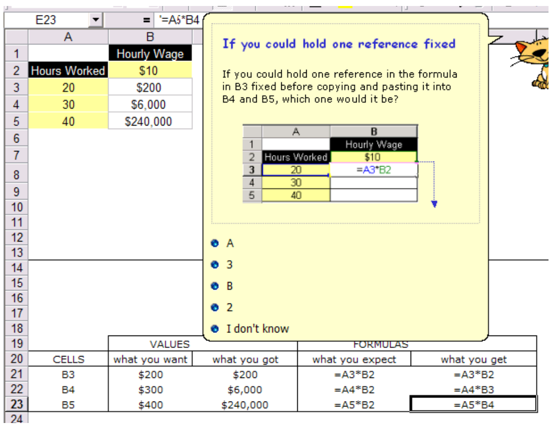
12. Student is prompted to analyze the discrepancy between actual and intended outcomes in subsequent cells.



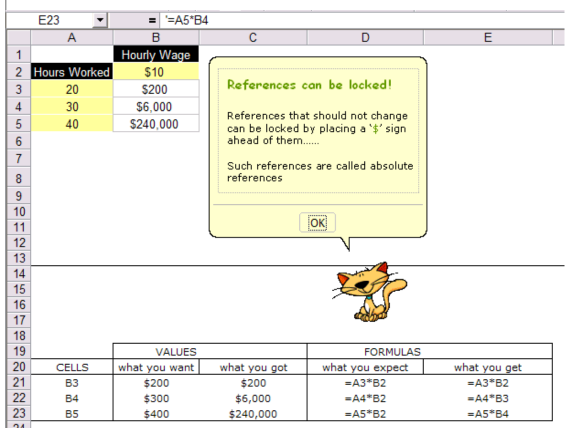
13. student is asked whether the formula worked



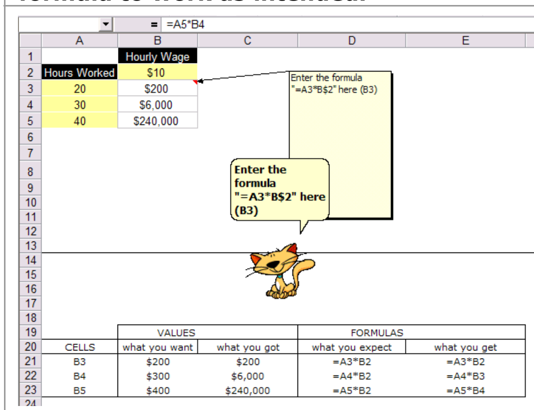
14. Student is asked to identify incorrect part of formula that resulted from copy and paste operation.



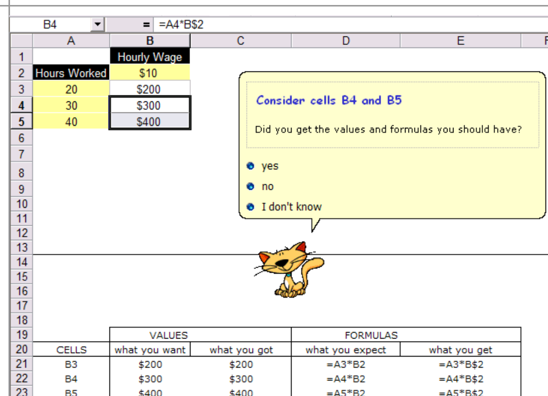
15. Student is asked to identify the reference that must be held fixed in order for the formula to work as intended.



16. Student is reminded of the absolute reference operator.

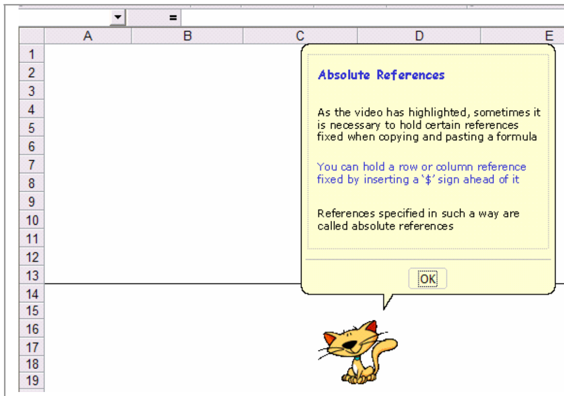


17. Student enters the correct formula and is prompted to copy and paste it.

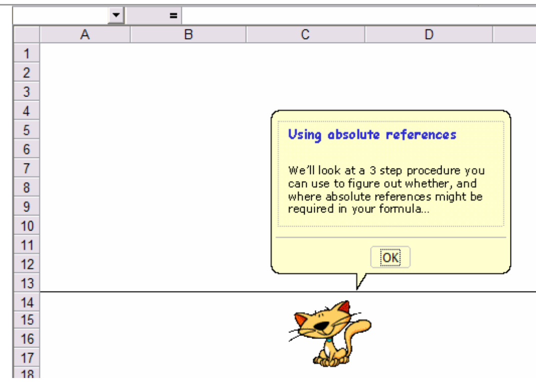


18. Student is asked to verify that the formula worked.

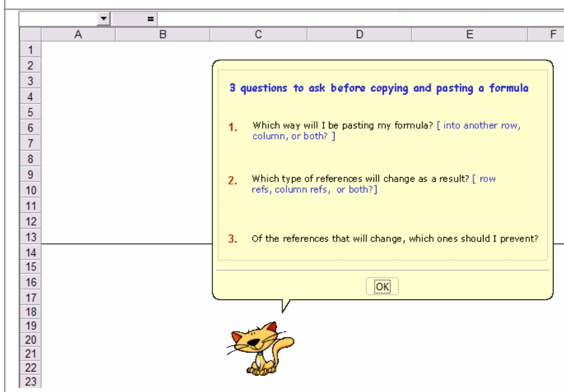
Appendix 7: Screen Shots of Expert Walkthrough in Study-2



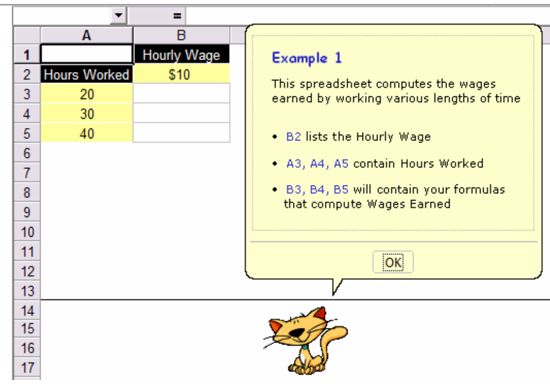
1. Overview balloon 1



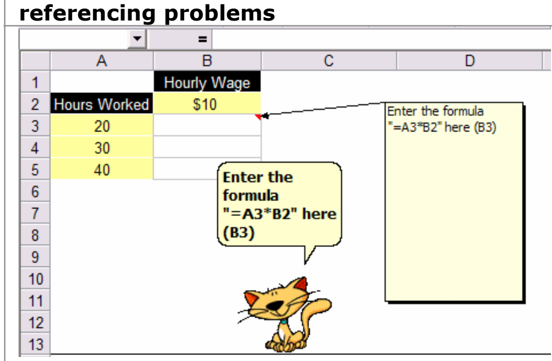
2. Overview balloon 2



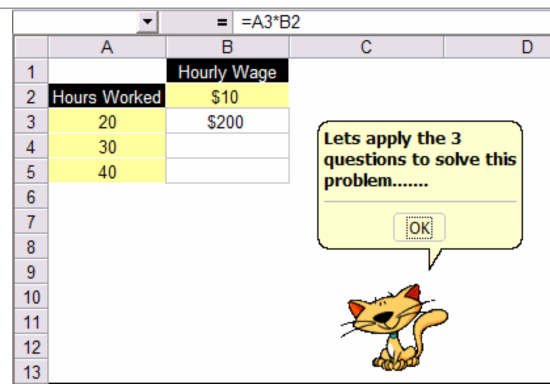
3. Student is introduced to a 3-step procedure to generate solutions to cell referencing problems



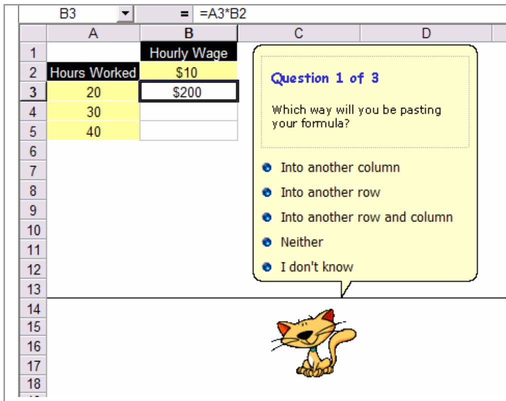
4. Student is given an overview of problem to be solved



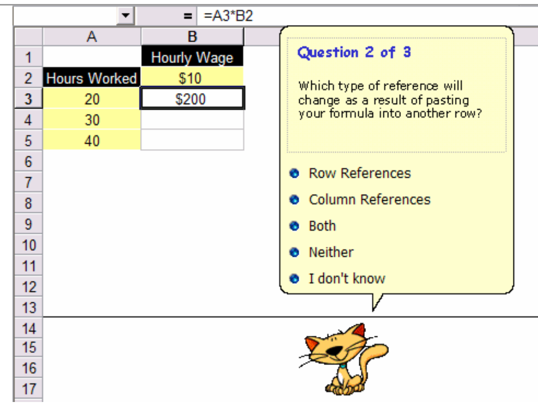
5 Student is asked to type in a formula without any absolute references.



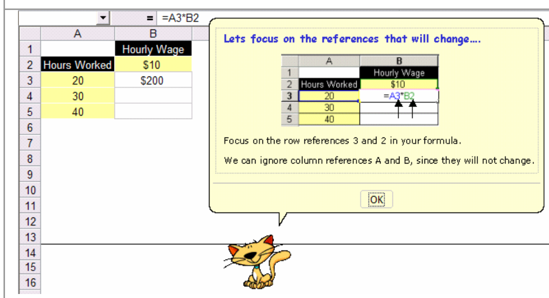
6. Student is led through a 3-step process to determine where, if any, absolute references may be necessary\



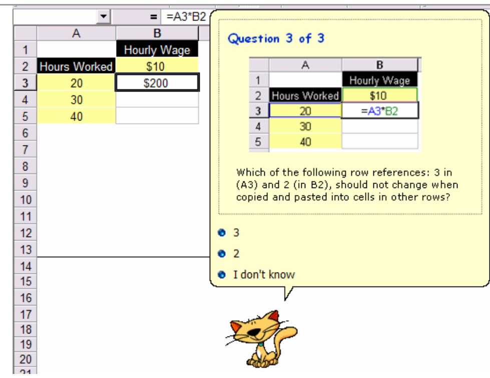
7. Student is asked to consider the direction in which formula is to be copied.



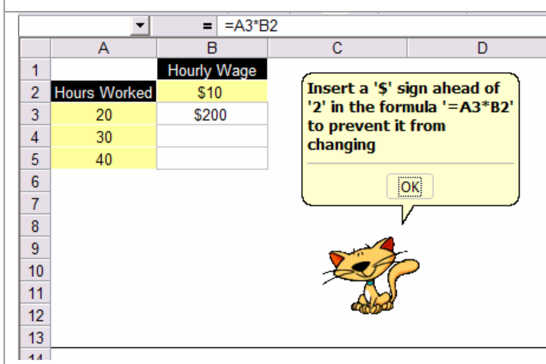
8. Student is asked to consider the type of reference that will change



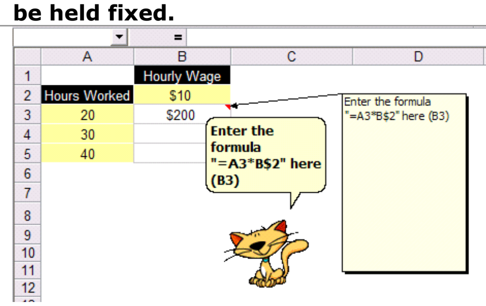
9. Student is asked to focus on the references that will change



10. Of the references that will change, student is asked to indicate ones that should be held fixed.

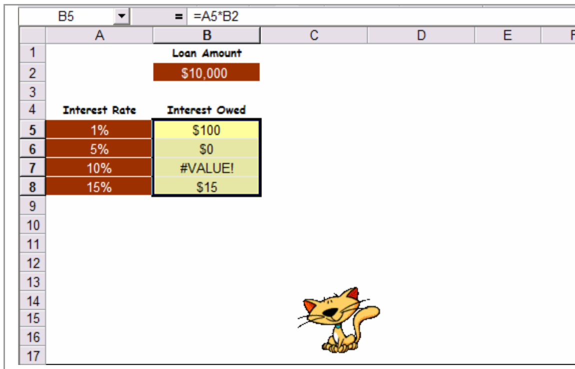


11. Student is told how the reference just identified as requiring an absolute reference can be held fixed

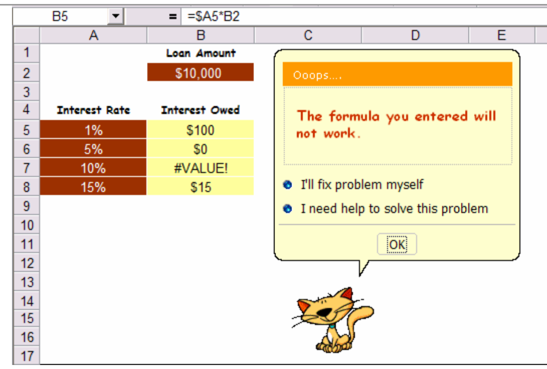


12. Student is prompted to enter the solution

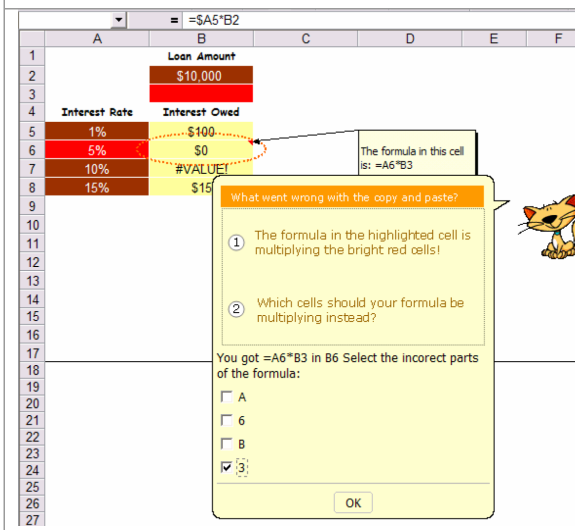
Appendix 8: Screen Shots of Intelligent Novice Feedback in Study-2



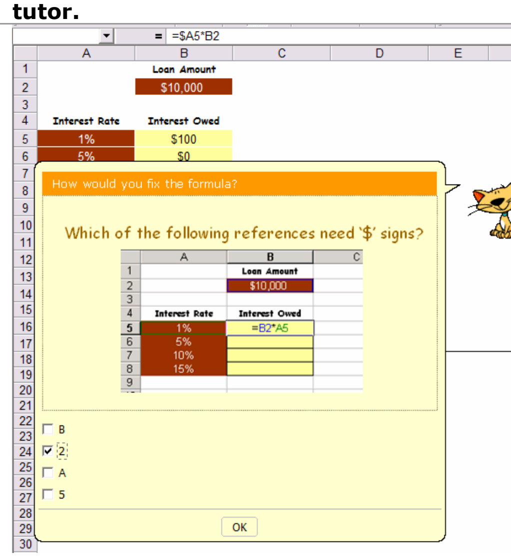
1. Student enters, copies, and pastes an incorrect formula. Student sees the consequence of incorrect action.



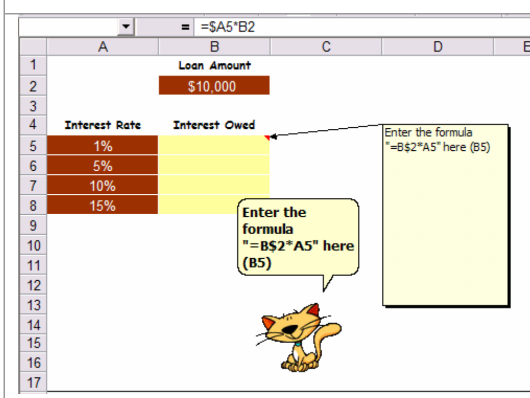
2. Student attempts to make repair and fails. Student is notified of error and given option to solve problem with help from the tutor.



3. Student is asked to identify the incorrect reference in formula.

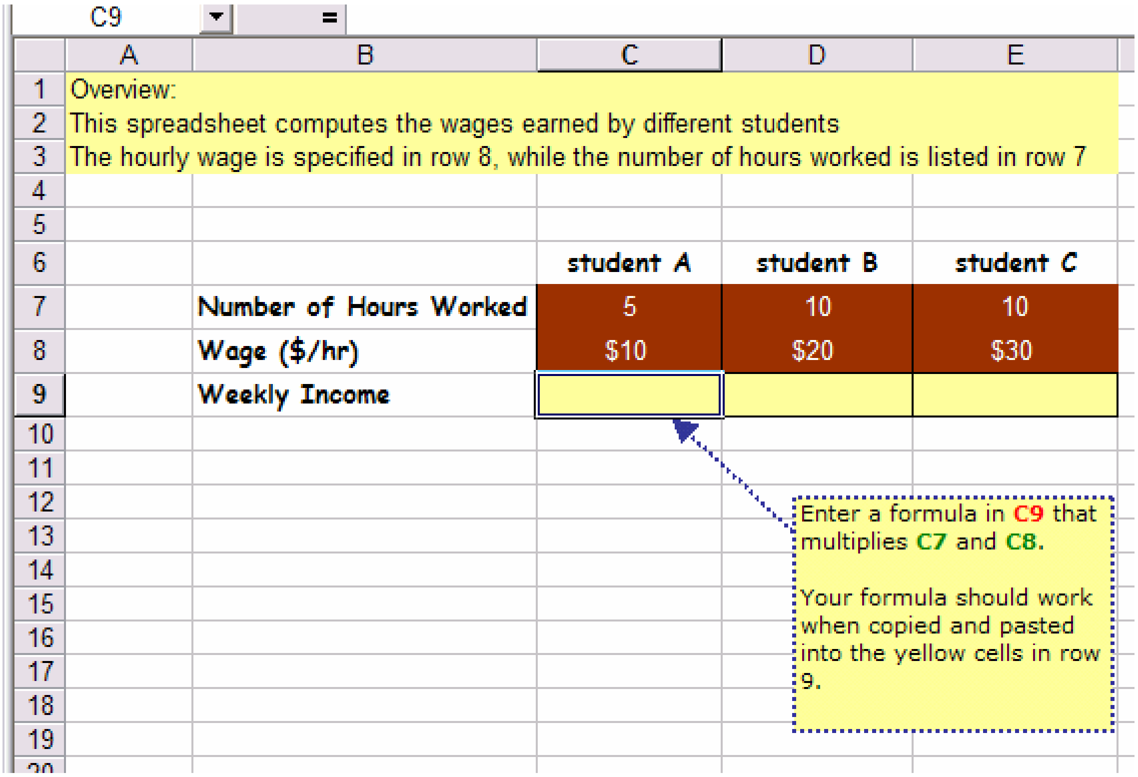


4. Student is asked to indicate the appropriate fix to prevent error just observed.



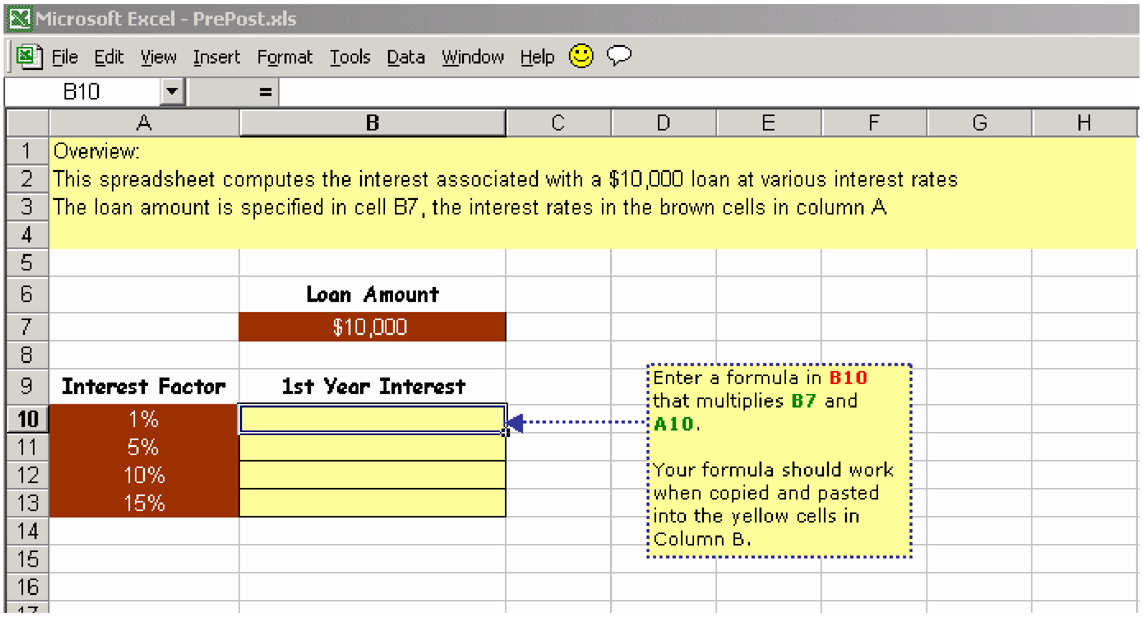
5. Student is prompted to enter correct formula.

Appendix 9: Problem Solving Test



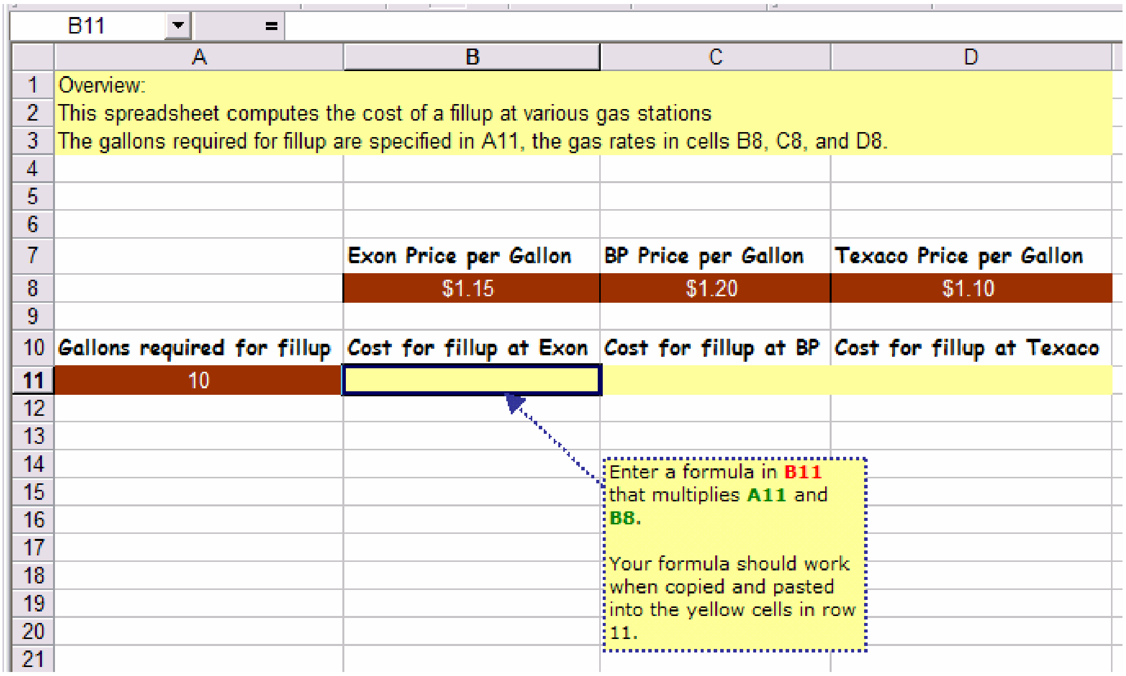
Correct Response: = C7*C8 (1point)

Penalty of 0.25 points for each redundant '$' sign ahead of the row references references 7 or 8



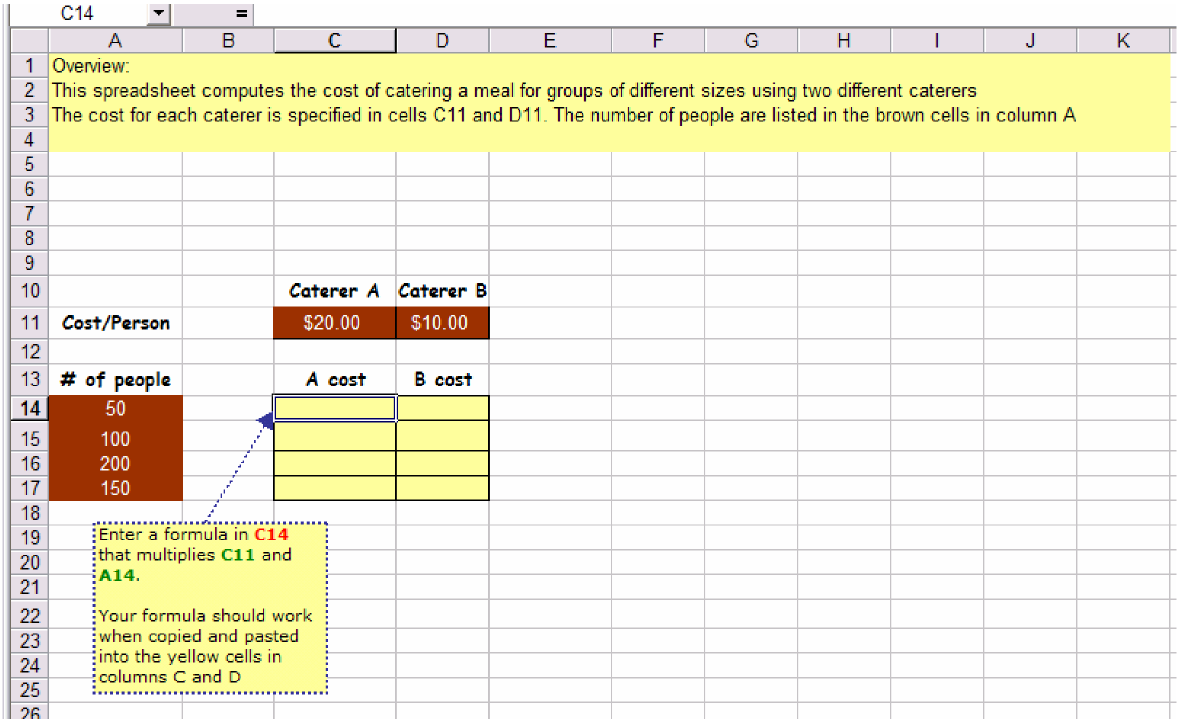
Correct Response: = B$7*A10 (2 points)

Penalty of 0.50 points for each redundant '$' sign ahead of the column references B or A



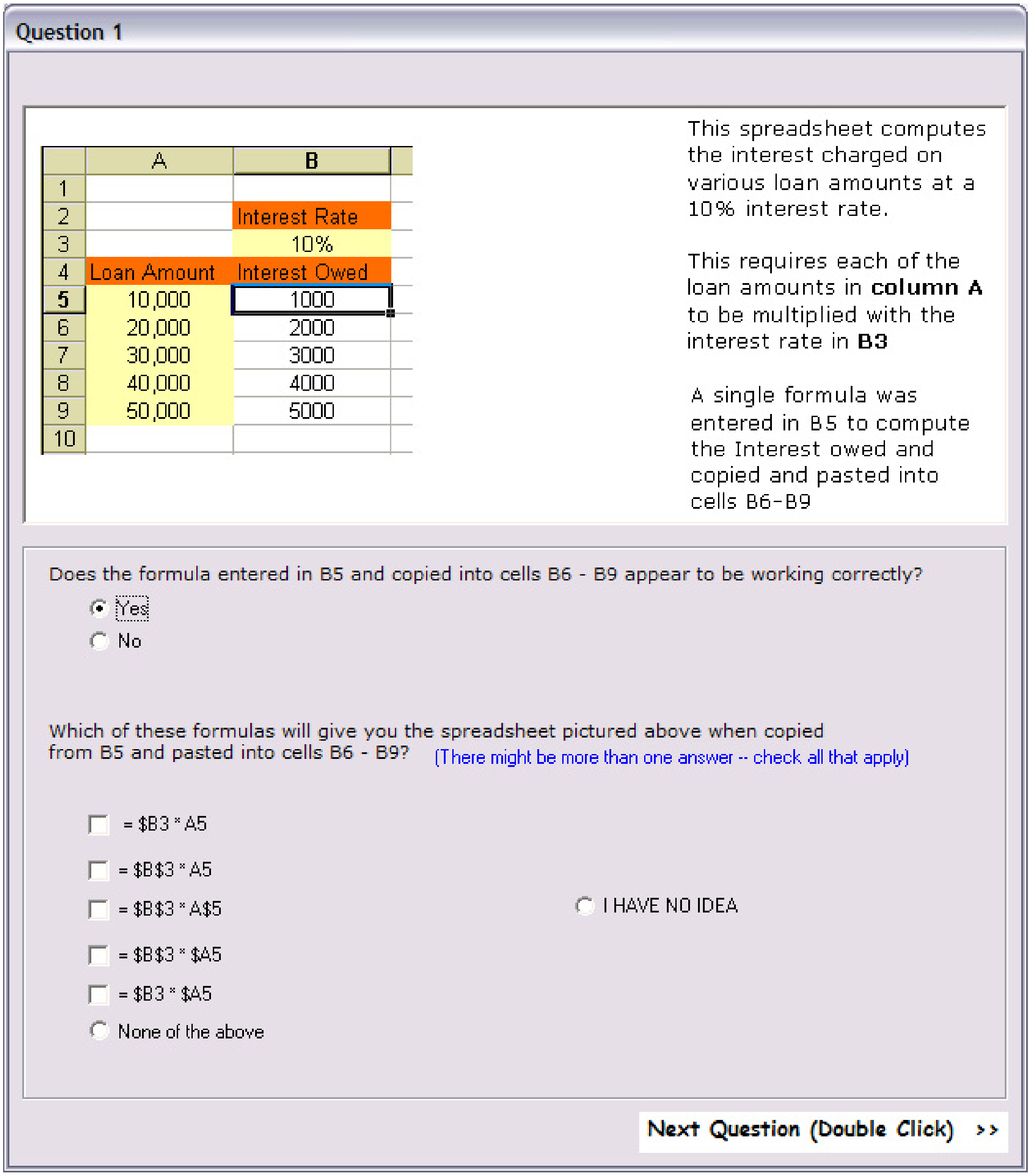
Correct Response: = $A11*B8 (2 points)

Penalty of 0.50 points for each redundant '$' sign ahead of the row references 11 or 8



Correct Response: = $A14*C$11 (5 points)

Appendix 10 Conceptual Test (Attribution Items)



Formula working: yes (1 point)

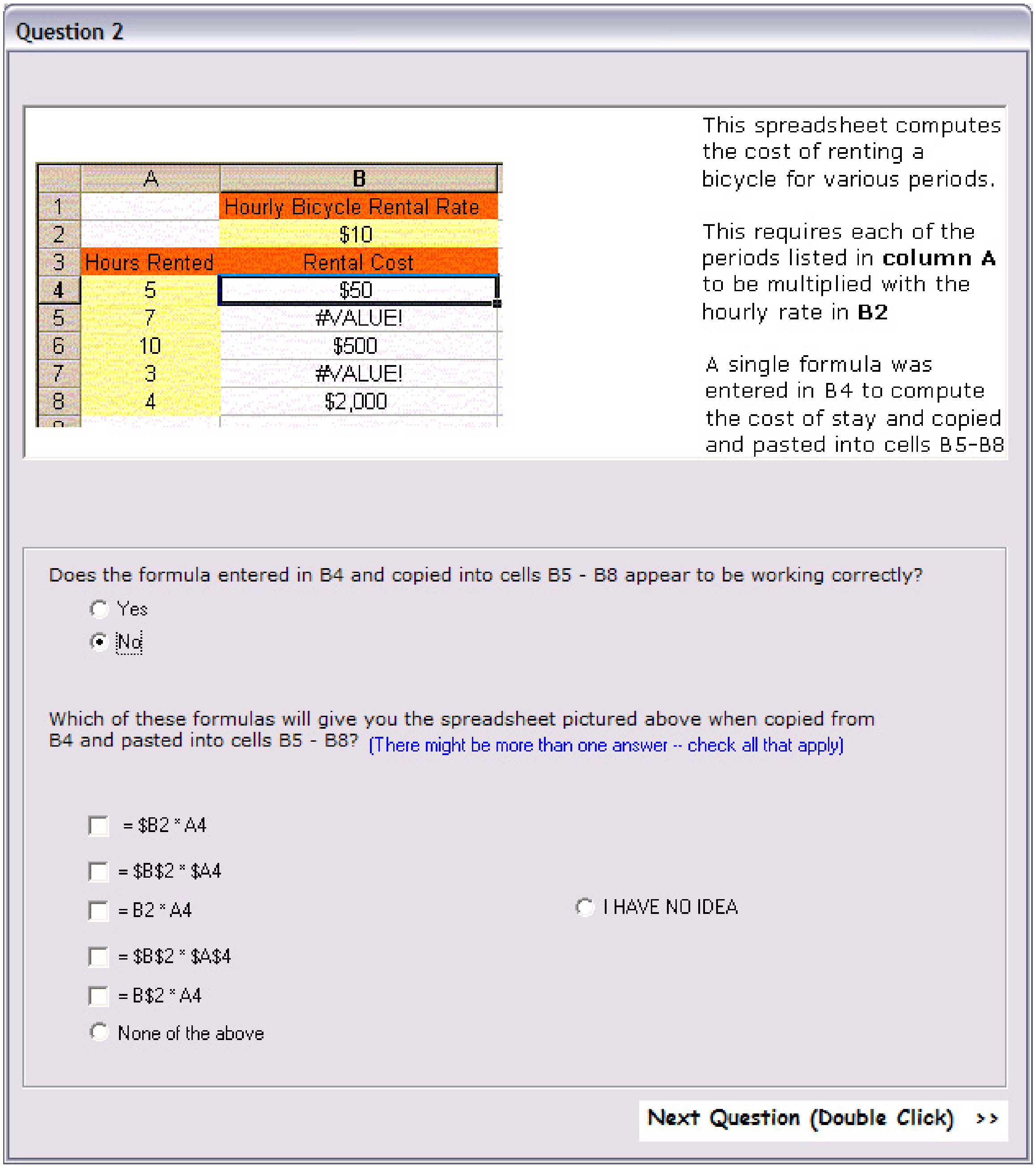
=$B3*A5 (false) (1point)

=$B$3*A5 (true) (1 point)

=$B$3*A$5 (false) (1 point)

=$B$3*$A5 (true) (1 point)

=$B3*$A5 (false) (1 point)



Formula working: no (1 point)

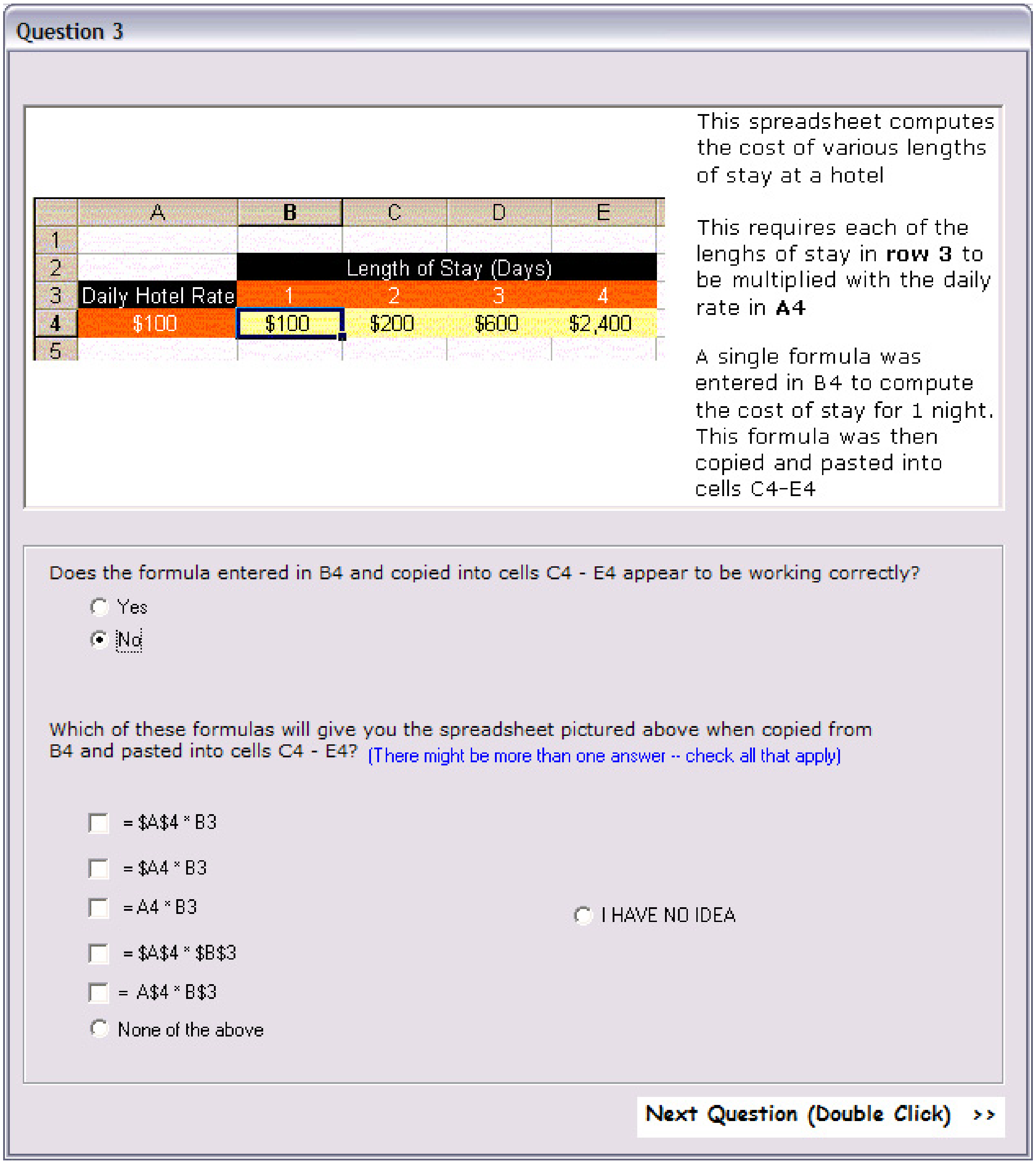
=$B2*A4 (true) (1point)

=$B$2*$A4 (false) (1 point)

=B2*A4 (true) (1 point)

=$B$2*$A$4 (false) (1 point)

=B$2*A4 (false) (1 point)



Formula working: no (1 point)

=$A$4*B3 (false) (1point)

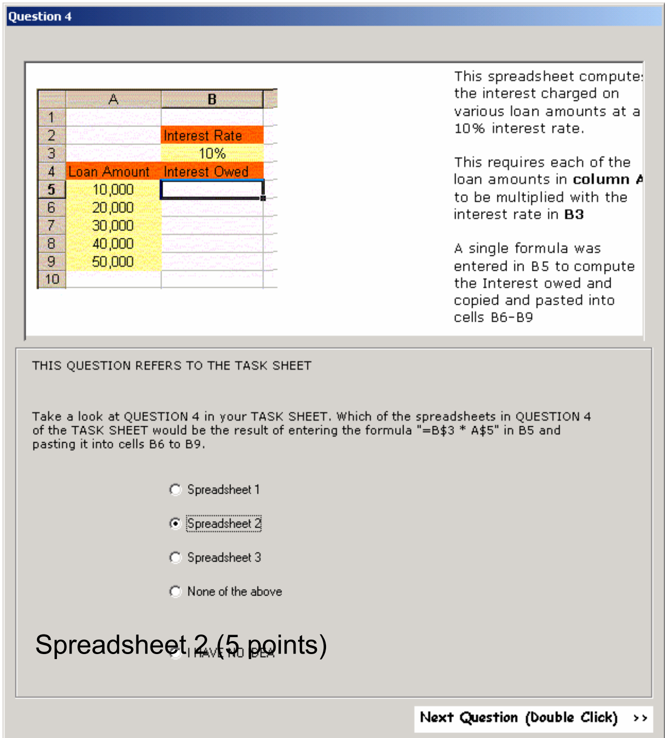
=$A4*B3 (false) (1 point)

=A4*B3 (true) (1 point)

=$A$4*$B$3 (false) (1 point)

=A$4*B$3 (true) (1 point)

Appendix 11: Conceptual Test (Predictive Items)


Question 4

	A	B
1		
2		Interest Rate
3		10%
4	Loan Amount	Interest Owed
5	10,000	
6	20,000	
7	30,000	
8	40,000	
9	50,000	
10		

This spreadsheet compute: the interest charged on various loan amounts at a 10% interest rate.

This requires each of the loan amounts in **column A** to be multiplied with the interest rate in **B3**

A single formula was entered in B5 to compute the Interest owed and copied and pasted into cells B6-B9

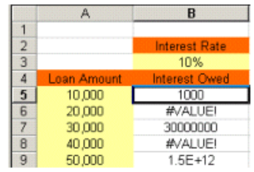
THIS QUESTION REFERS TO THE TASK SHEET

Take a look at QUESTION 4 in your TASK SHEET. Which of the spreadsheets in QUESTION 4 of the TASK SHEET would be the result of entering the formula "=B$3 * A$5" in B5 and pasting it into cells B6 to B9.

- Spreadsheet 1
- Spreadsheet 2
- Spreadsheet 3
- None of the above

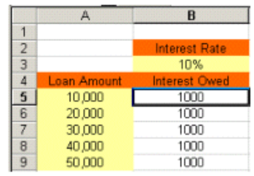
Spreadsheet 2 (5 points)

Next Question (Double Click) >>



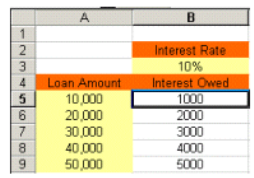
	A	B
1		
2		Interest Rate
3		10%
4	Loan Amount	Interest Owed
5	10,000	1000
6	20,000	#VALUE!
7	30,000	30000000
8	40,000	#VALUE!
9	50,000	1.5E+12

Spreadsheet 1



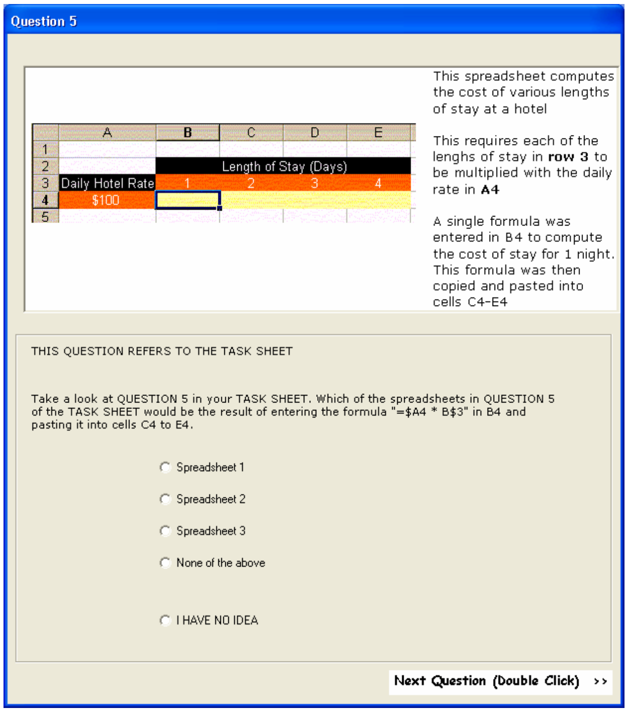
	A	B
1		
2		Interest Rate
3		10%
4	Loan Amount	Interest Owed
5	10,000	1000
6	20,000	1000
7	30,000	1000
8	40,000	1000
9	50,000	1000

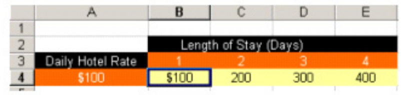
Spreadsheet 2



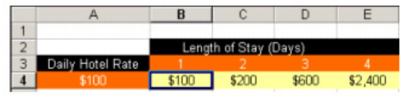
	A	B
1		
2		Interest Rate
3		10%
4	Loan Amount	Interest Owed
5	10,000	1000
6	20,000	2000
7	30,000	3000
8	40,000	4000
9	50,000	5000

Spreadsheet 3

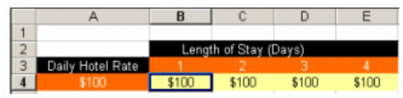




Spreadsheet 1



Spreadsheet 2



Spreadsheet 3

Spreadsheet 1 (5 points)

Appendix 12: Computer Experience Questionnaire

Name ______________________________ Date __________

Frequency of use of various computer applications	
Never (1) Once or Twice a Year (2) Monthly (3) Weekly (4) Daily (5)	
Application	**Your Rating**
Games	
Word Processing	
Spreadsheets	
Presentations (Power Point etc.)	
Programming	
Database Creation	
Multimedia Content Creation (animation, digital audio/video creation and editing)	
Web Browsing	
e-mail	
Graphics (Photoshop etc.)	
Web Page Creation	

Expertise With Various Applications	
I always need help with this (1) I sometimes need help with this (2) I rarely need help with this (3) I never need help (4) I can help others, I am an expert (5)	
Application	**Your Rating**
Games	
Word Processing	
Spreadsheets	
Presentations (Power Point etc.)	
Programming	
Database Creation	
Multimedia Content Creation (animation, digital audio/video creation and editing)	
Web Browsing	
e-mail	
Graphics (Photoshop etc.)	
Web Page Creation	

How comfortable do you feel using computers, in general?

Very comfortable

Somewhat comfortable

Neither comfortable nor uncomfortable

Somewhat uncomfortable

Very uncomfortable

Prior to this study, which of the following tasks (if any) have you performed using spreadsheets? Circle all that apply.

Graphs

Data Entry

Formulas

Functions

Cell Referencing

Data Analysis

Macros

Which operating system do you primarily use? (circle one)

UNIX Macintosh Windows

Please indicate the highest level of education completed (circle one):

Grammar School

High School or equivalent

Vocational/Technical School (2 year)

Some College Years: Major:

College Graduate (4 year) Major:

Master's Degree (MS/MA) Major:

Doctoral Degree (PhD) Major:

Professional Degree (MD,JD, etc.)

Please indicate your age:

15 -19

20-24

25-29

30-34

35-39

40-44

45-49

50-54

55-59

60-64

over 65

Appendix 13: Transfer Test

TASK OVERVIEW

A travel agency with several corporate accounts maintains a spreadsheet to keep track of commissions and other charges.

Your job is to specify a set of formulas that will compute these charges.

INSTRUCTIONS

Construct the required formulas as quickly as you can.

If any of your formulas require the use of absolute references, try to use the **least number** of absolute references possible.

INSTRUCTIONS (continued)

Prior to every task you will be prompted to read a task description.

Double click the ‘Perform Task’ button as soon as you are ready to begin.

Double click on the ‘Done’ button as soon as you have completed a task.

TASK 1	Enter a formula in **E4** that computes the travel expenses for the month of January. Your formula will have to add the contents of **B4**, **C4**, and **D4** Your formula should compute the total expenses for other months when copied from **E4** and pasted into cells **E5 to E15**.

Double click on the 'Done' button as soon as you have completed task.

Correct Response: =B4+C4+D4 (1 point)
0.25 point penalty for each redundant $ ahead of column references B, C or D

Spreadsheet used by the student for this task:

E4 =

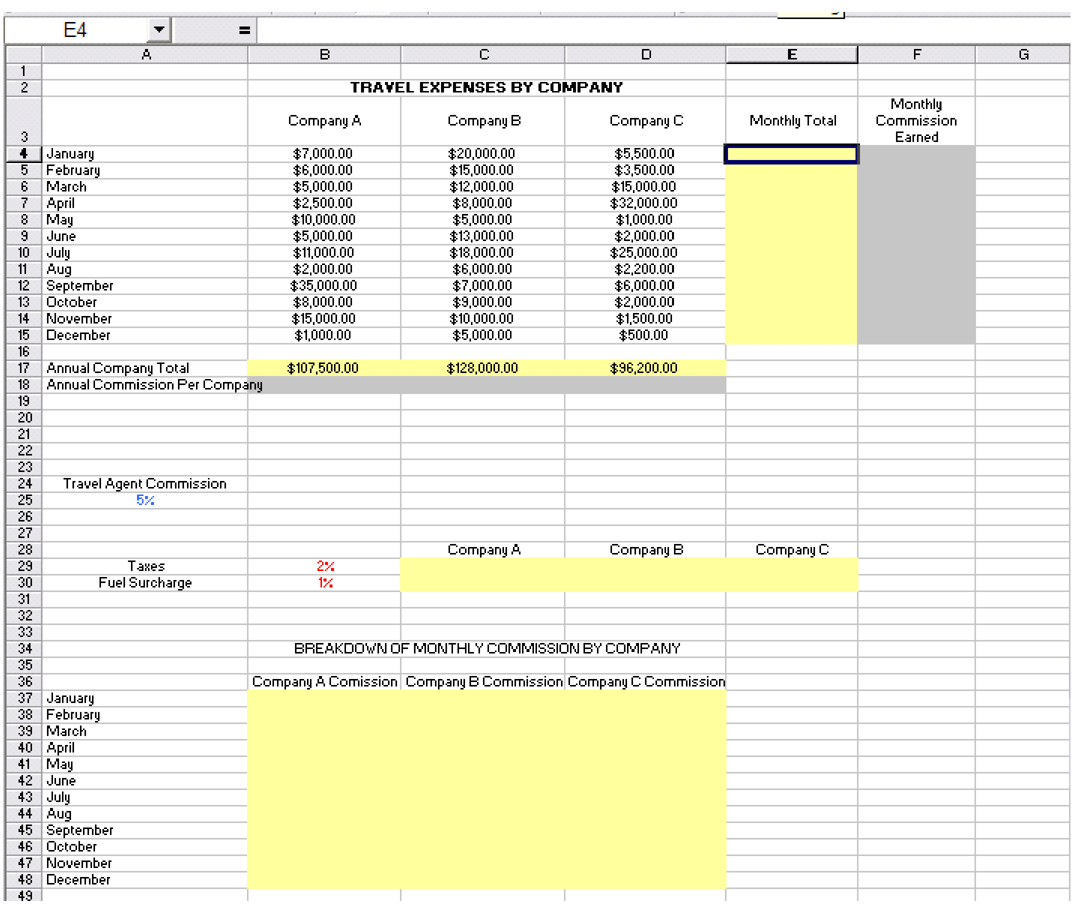
	A	B	C	D	E	F	G
1							
2			**TRAVEL EXPENSES BY COMPANY**				
3		Company A	Company B	Company C	Monthly Total	Monthly Commission Earned	
4	January	$7,000.00	$20,000.00	$5,500.00			
5	February	$6,000.00	$15,000.00	$3,500.00			
6	March	$5,000.00	$12,000.00	$15,000.00			
7	April	$2,500.00	$8,000.00	$32,000.00			
8	May	$10,000.00	$5,000.00	$1,000.00			
9	June	$5,000.00	$13,000.00	$2,000.00			
10	July	$11,000.00	$18,000.00	$25,000.00			
11	Aug	$2,000.00	$6,000.00	$2,200.00			
12	September	$35,000.00	$7,000.00	$6,000.00			
13	October	$8,000.00	$9,000.00	$2,000.00			
14	November	$15,000.00	$10,000.00	$1,500.00			
15	December	$1,000.00	$5,000.00	$500.00			
16							
17	Annual Company Total	$107,500.00	$128,000.00	$96,200.00			
18	Annual Commission Per Company						
19							
20							
21							
22							
23							
24	Travel Agent Commission						
25	5%						
26							
27							
28			Company A	Company B	Company C		
29	Taxes	2%					
30	Fuel Surcharge	1%					
31							
32							
33							
34			BREAKDOWN OF MONTHLY COMMISSION BY COMPANY				
35							
36		Company A Comission	Company B Commission	Company C Commission			
37	January						
38	February						
39	March						
40	April						
41	May						
42	June						
43	July						
44	Aug						
45	September						
46	October						
47	November						
48	December						
49							

TASK 2	Enter a formula in **F4** that computes the *Monthly Commission* earned in the month of January. You'll have to multiply the *Monthly Total* amount in **E4** with the *Commission Rate* in **A25**. Your formula should compute the *Monthly Commission* for other months when copied from **F4** and pasted into cells **F5 to F15**.

Double click on the 'Done' button as soon as you have completed task.

Correct Response: =A$25*E4 (2 points)
0.5 point penalty for each redundant $ ahead of column references A or E

Spreadsheet used by the student for this task:



E4 =

	A	B	C	D	E	F	G
1							
2			TRAVEL EXPENSES BY COMPANY				
3		Company A	Company B	Company C	Monthly Total	Monthly Commission Earned	
4	January	$7,000.00	$20,000.00	$5,500.00			
5	February	$6,000.00	$15,000.00	$3,500.00			
6	March	$5,000.00	$12,000.00	$15,000.00			
7	April	$2,500.00	$8,000.00	$32,000.00			
8	May	$10,000.00	$5,000.00	$1,000.00			
9	June	$5,000.00	$13,000.00	$2,000.00			
10	July	$11,000.00	$18,000.00	$25,000.00			
11	Aug	$2,000.00	$6,000.00	$2,200.00			
12	September	$35,000.00	$7,000.00	$6,000.00			
13	October	$8,000.00	$9,000.00	$2,000.00			
14	November	$15,000.00	$10,000.00	$1,500.00			
15	December	$1,000.00	$5,000.00	$500.00			
16							
17	Annual Company Total	$107,500.00	$128,000.00	$96,200.00			
18	Annual Commission Per Company						
19							
20							
21							
22							
23							
24	Travel Agent Commission						
25	5%						
26							
27							
28			Company A	Company B	Company C		
29	Taxes	2%					
30	Fuel Surcharge	1%					
31							
32							
33							
34		BREAKDOWN OF MONTHLY COMMISSION BY COMPANY					
35							
36		Company A Comission	Company B Commission	Company C Commission			
37	January						
38	February						
39	March						
40	April						
41	May						
42	June						
43	July						
44	Aug						
45	September						
46	October						
47	November						
48	December						
49							

TASK 3	Enter a formula in **B18** that computes the *Annual Commission* earned from *Company A*. You'll have to multiply the *Annual Company Total* amount for Company A in **B17** with the commission rate in **A25**. Your formula should compute *Annual Commissions* for other companies when copied and pasted into cells **C18 and D18**.

Double click on the 'Done' button as soon as you have completed task.

Correct Response: =$A25*B17
0.5 point penalty for each redundant $ ahead of row references 25 or 17

Spreadsheet used by the student for this task:

E4 =

	A	B	C	D	E	F	G
1							
2			TRAVEL EXPENSES BY COMPANY				
3		Company A	Company B	Company C	Monthly Total	Monthly Commission Earned	
4	January	$7,000.00	$20,000.00	$5,500.00			
5	February	$6,000.00	$15,000.00	$3,500.00			
6	March	$5,000.00	$12,000.00	$15,000.00			
7	April	$2,500.00	$8,000.00	$32,000.00			
8	May	$10,000.00	$5,000.00	$1,000.00			
9	June	$5,000.00	$13,000.00	$2,000.00			
10	July	$11,000.00	$18,000.00	$25,000.00			
11	Aug	$2,000.00	$6,000.00	$2,200.00			
12	September	$35,000.00	$7,000.00	$6,000.00			
13	October	$8,000.00	$9,000.00	$2,000.00			
14	November	$15,000.00	$10,000.00	$1,500.00			
15	December	$1,000.00	$5,000.00	$500.00			
16							
17	Annual Company Total	$107,500.00	$128,000.00	$96,200.00			
18	Annual Commission Per Company						
19							
20							
21							
22							
23							
24	Travel Agent Commission						
25	5%						
26							
27							
28			Company A	Company B	Company C		
29	Taxes	2%					
30	Fuel Surcharge	1%					
31							
32							
33							
34			BREAKDOWN OF MONTHLY COMMISSION BY COMPANY				
35							
36		Company A Comission	Company B Commission	Company C Commission			
37	January						
38	February						
39	March						
40	April						
41	May						
42	June						
43	July						
44	Aug						
45	September						
46	October						
47	November						
48	December						
49							

TASK 4	This task requires you to figure out the annual taxes and fuel surcharge expenses for companies A, B, and C. Enter a formula in **C29** that multiplies the tax rate in **B29** with B17 -- the *Annual Company Total* amount for *Company A*. Your formula should compute taxes and fuel surcharge on annual travel expenses by *Company A*, *Company B*, and *Company C* when copied from **C29** and pasted into cells **D29, E29, C30, D30 and E30**.

Double click on the 'Done' button as soon as you have completed task.

Correct Response: =$B29*B$17 (5 points)

Spreadsheet used by the student for this task:

E4 =

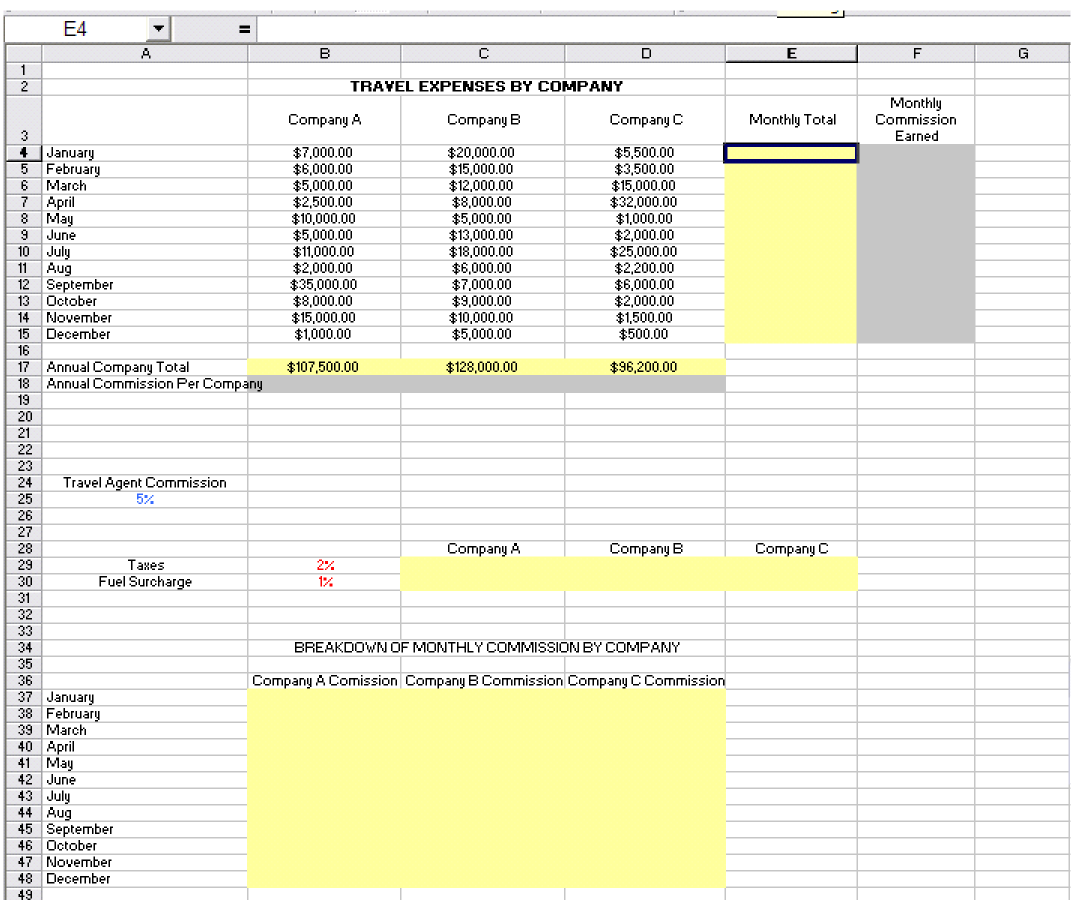
	A	B	C	D	E	F	G
1							
2			**TRAVEL EXPENSES BY COMPANY**				
3		Company A	Company B	Company C	Monthly Total	Monthly Commission Earned	
4	January	$7,000.00	$20,000.00	$5,500.00			
5	February	$6,000.00	$15,000.00	$3,500.00			
6	March	$5,000.00	$12,000.00	$15,000.00			
7	April	$2,500.00	$8,000.00	$32,000.00			
8	May	$10,000.00	$5,000.00	$1,000.00			
9	June	$5,000.00	$13,000.00	$2,000.00			
10	July	$11,000.00	$18,000.00	$25,000.00			
11	Aug	$2,000.00	$6,000.00	$2,200.00			
12	September	$35,000.00	$7,000.00	$6,000.00			
13	October	$8,000.00	$9,000.00	$2,000.00			
14	November	$15,000.00	$10,000.00	$1,500.00			
15	December	$1,000.00	$5,000.00	$500.00			
16							
17	Annual Company Total	$107,500.00	$128,000.00	$96,200.00			
18	Annual Commission Per Company						
19							
20							
21							
22							
23							
24	Travel Agent Commission						
25	5%						
26							
27							
28			Company A	Company B	Company C		
29	Taxes	2%					
30	Fuel Surcharge	1%					
31							
32							
33							
34			BREAKDOWN OF MONTHLY COMMISSION BY COMPANY				
35							
36		Company A Comission	Company B Commission	Company C Commission			
37	January						
38	February						
39	March						
40	April						
41	May						
42	June						
43	July						
44	Aug						
45	September						
46	October						
47	November						
48	December						
49							

TASK 5	Compute the *Monthly Commission* charged to each of the 3 companies Enter a formula in **B37** that multiplies January travel expenses for *Company A* in **B4** with the commission amount in **A25.** Your formula should calculate the commission charges for each of the companies for each of the 12 months when copied and pasted into cells **C37 to D48**.

Double click on the 'Done' button as soon as you have completed task.

Correct Response: =$A$25*B4 (5 points)

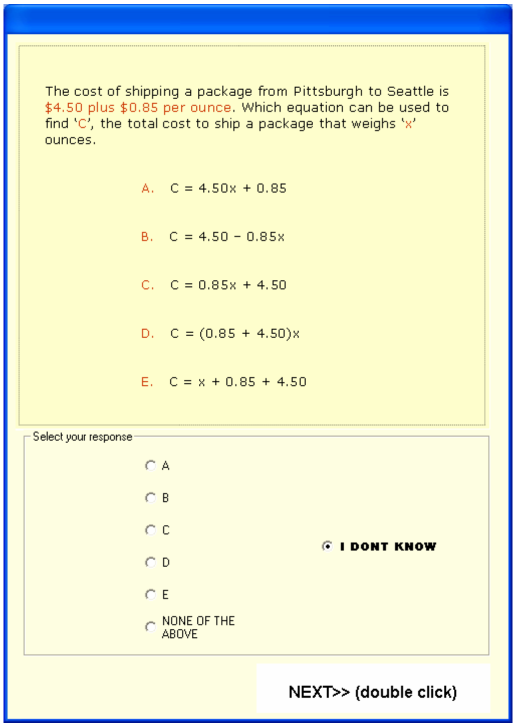
Spreadsheet used by the student for this task:



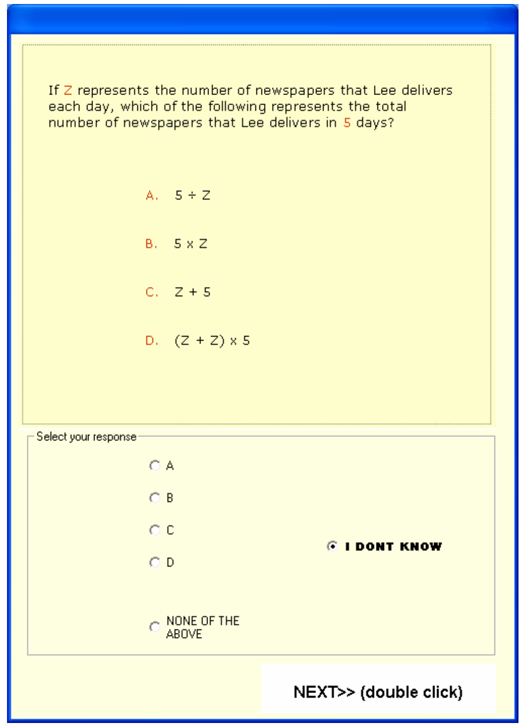
E4 =

	A	B	C	D	E	F	G
1							
2		TRAVEL EXPENSES BY COMPANY					
3		Company A	Company B	Company C	Monthly Total	Monthly Commission Earned	
4	January	$7,000.00	$20,000.00	$5,500.00			
5	February	$6,000.00	$15,000.00	$3,500.00			
6	March	$5,000.00	$12,000.00	$15,000.00			
7	April	$2,500.00	$8,000.00	$32,000.00			
8	May	$10,000.00	$5,000.00	$1,000.00			
9	June	$5,000.00	$13,000.00	$2,000.00			
10	July	$11,000.00	$18,000.00	$25,000.00			
11	Aug	$2,000.00	$6,000.00	$2,200.00			
12	September	$35,000.00	$7,000.00	$6,000.00			
13	October	$8,000.00	$9,000.00	$2,000.00			
14	November	$15,000.00	$10,000.00	$1,500.00			
15	December	$1,000.00	$5,000.00	$500.00			
16							
17	Annual Company Total	$107,500.00	$128,000.00	$96,200.00			
18	Annual Commission Per Company						
19							
20							
21							
22							
23							
24	Travel Agent Commission						
25	5%						
26							
27							
28			Company A	Company B	Company C		
29	Taxes	2%					
30	Fuel Surcharge	1%					
31							
32							
33							
34		BREAKDOWN OF MONTHLY COMMISSION BY COMPANY					
35							
36		Company A Comission	Company B Commission	Company C Commission			
37	January						
38	February						
39	March						
40	April						
41	May						
42	June						
43	July						
44	Aug						
45	September						
46	October						
47	November						
48	December						
49							

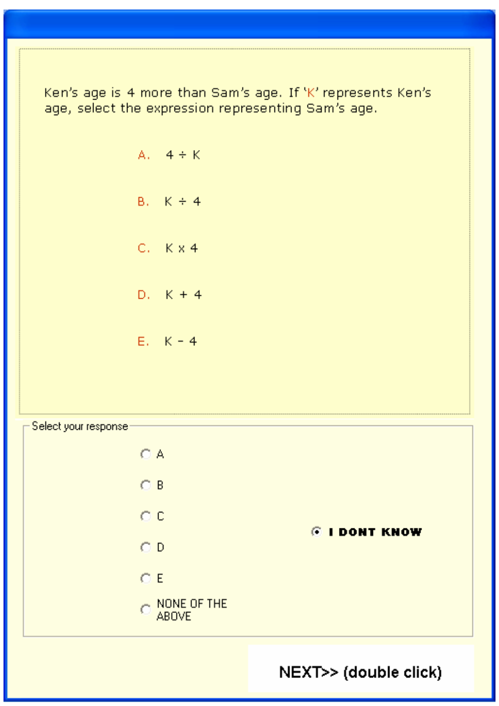
Appendix 14: Math Test



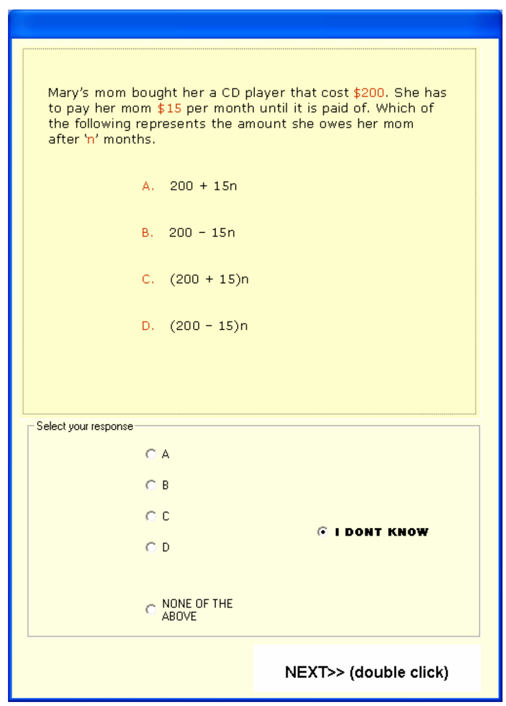
response: c (1 point)



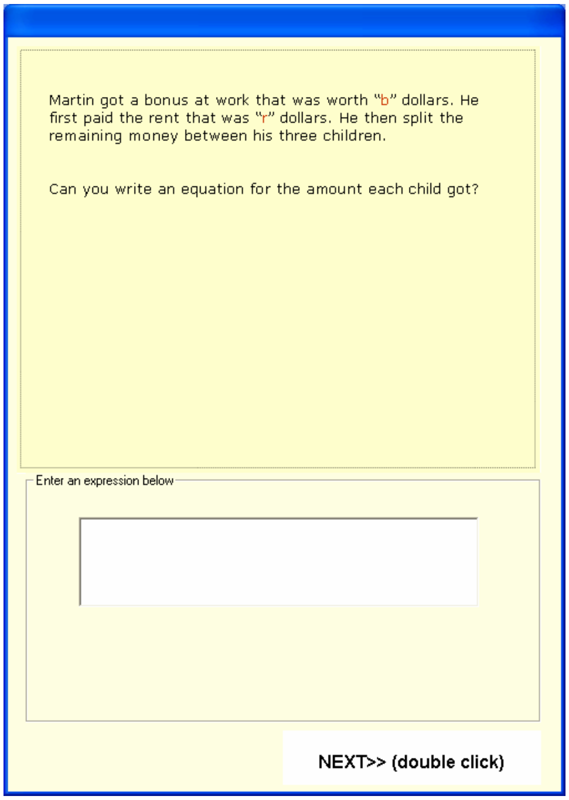
response: b (1 point)



response: e (1 point)



response: b (1 point)



response: (b-r)/3 (1 point)