Analyzing Student Problem-Solving Behavior in a Step-Based Tutor and Understanding the Effect of Unsolicited Hints

By

Rajagopalan Ranganathan

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved August 2011 by the
Graduate Supervisory Committee:

Kurt VanLehn, Chair
Robert Atkinson
Winslow Burleson

ARIZONA STATE UNIVERSITY

December 2011

ABSTRACT

Lots of previous studies have analyzed human tutoring at great depths and have shown expert human tutors to produce effect sizes, which is twice of that produced by an intelligent tutoring system (ITS). However, there has been no consensus on which factor makes them so effective. It is important to know this, so that same phenomena can be replicated in an ITS in order to achieve the same level of proficiency as expert human tutors. Also, to the best of my knowledge no one has looked at student reactions when they are working with a computer based tutor. The answers to both these questions are needed in order to build a highly effective computer-based tutor. My research focuses on the second question. In the first phase of my thesis, I analyzed the behavior of students when they were working with a step-based tutor Andes, using verbal-protocol analysis. The accomplishment of doing this was that I got to know of some ways in which students use a step-based tutor which can pave way for the creation of more effective computer-based tutors. I found from the first phase of the research that students often keep trying to fix errors by guessing repeatedly instead of asking for help by clicking the hint button. This phenomenon is known as hint refusal. Surprisingly, a large portion of the student's foundering was due to hint refusal. The hypothesis tested in the second phase of the research is that hint refusal can be significantly reduced and learning can be significantly increased if Andes uses more unsolicited hints and meta hints. An unsolicited hint is a hint that is given without the student asking for one. A meta-hint is like an unsolicited hint in that it is given without the student asking for it, but it just prompts the student to click on the hint button. Two versions of Andes were compared: the original version and a new version that gave more unsolicited and meta-hints. During a two-hour experiment, there were large, statistically reliable differences in several performance measures suggesting that the new policy was more effective.

DEDICATION

To my Amma and Appa

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Kurt VanLehn for all the guidance and encouragement. I would also like to thank Brett for all the help, without which this research could not have been finished. Last but not the least I want to thank my parents and all my friends for their constant support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

WHAT DO STUDENTS DO WHEN USING A STEP-BASED TUTORING SYSTEM?

Introduction

Developers of intelligent tutoring systems would like to know what students do when they are working with the tutor. This might help them develop tutors that are more effective in helping students learn. Although there have been many studies of human tutoring (discussed below), we are not aware of a similar study of students' behavior while being tutored by an intelligent tutoring system (ITS). In order to find out, at least in a preliminary way, what students do when being tutored by an ITS, we had 10 students work for 2 hours on Andes while talking out aloud. Initial observations and inferences are presented in this paper, along with suggestions for how to mine more insights from this rich corpus.

Background

One-on-one tutoring by humans is thought to be a much more effective method of instruction than intelligent tutoring systems (Bloom, 1984; Corbett, 2001; Evens & Michael, 2006; Graesser, VanLehn, Rose, Jordan, & Harter, 2001; VanLehn, et al., 2007; Woolf, 2009). Thus, there have been many studies of human tutoring that have attempted to find out why they are so effective (Cade, Copeland, Person, & D'Mello, 2008; Chi, Siler, Jeong, Yamauchi, & Hausmann, 2001; Cho, Michael, Rovick, & Evens, 2000; Core, Moore, & Zinn, 2003; Evens & Michael, 2006; Fox, 1991, 1993; Frederiksen, Donin, & Roy, 2000; Graesser, Person, & Magliano, 1995; Hume, Michael, Rovick, & Evens, 1996; Katz, Allbritton, & Connelly, 2003; McArthur, Stasz, & Zmuidzinas, 1990; Merrill, Reiser, Merrill, & Landes, 1995; Merrill, Reiser, Ranney, & Trafton, 1992; Ohlsson, et al., 2007; VanLehn, 1999; VanLehn, Siler, Murray, Yamauchi, & Baggett, 2003). In a recent review, VanLehn (in press) extracted from the literature, 9 hypothesis

for why human tutoring should be more effective than ITS. He argued that only two of them were viable:

- *Feedback*: Human tutors almost always give immediate feedback on the student's contribution, which may make it easy for the student to self-repair their knowledge.
- *Scaffolding*: Human tutors often provide prompts or hints that push the student to go a little further down a correct line of reasoning. Scaffolding does not present new information to the student, as that would be called a tutorial explanation. It instead gets the student to generate a bit more progress with existing knowledge.

Another theoretical framework comes from Chi et al. (2001), who found support for three hypothesis, a tutor-centered one, a student-centered one and an interactive one to explain the effectiveness of human tutoring. Her study suggests that the effectiveness of a tutor depends not only on the pedagogical skills of the tutor, but also on the response that they got from the students i.e. how well do the students construct knowledge after interacting with the tutor. She later generalized her findings and gave a conceptual framework for differentiating learning activities (Chi, 2009). According to this framework, there are four types of instructional activities namely (starting from the least effective to the most effective) passive, active, constructive and interactive. For example, reading a book can be considered to be a passive activity, reading a book and highlighting important points can be considered to be active, answering questions given at the back of the book can be called as constructive and discussing the questions with a peer or a tutor can be called an interactive activity. By interpreting results of various existing studies, she tested her hypothesis that interactive activities might be better than constructive activities, which in turn might be better than active activities, and which is better than the passive ones.

These theoretical frameworks agree that it is important to know how students behave when they work with a human tutor and, presumably, an ITS. Observations of student behavior correlate more strongly with learning gains than observations of tutors' behavior. However, to the best of our knowledge no one has looked into student reactions to an ITS, with one exception, namely the many studies that have explored "gaming the system," which is defined as a behavior where the student tries to exploit the properties of the tutoring system to succeed, rather than learn the content (R. S. J. d. Baker, Corbett, Koedinger, & Wagner, 2004; R. S. J. d. Baker, de Carvalho, Raspat, Corbett, & Koedinger, 2009; R. S. J. d. Baker, et al., 2008)(Aleven & Koedinger, 2000; R. S. Baker, et al., 2006; R. S. J. d. Baker, et al., 2009; R. S. J. d. Baker, et al., 2008; Hastings, Arnott-Hill, & Allbritton, 2010; Muldner, Burleson, van de Sande, & VanLehn, 2010; Murray & VanLehn, 2005; Shih, Koedinger, & Scheines, 2008;Walonoski et al. 2006).

The most common form of gaming is "hint abuse." When a tutoring system has a sequence of hints arranged from general to specific, student sometimes rapidly click through the general hints until they get to the last hint, which is called the bottom out hint. It states explicitly what the student must do to enter the next correct step. Although some students self-explain the bottom out hint (Shih, et al., 2008), most simply copy this information from the bottom out hint.

Because hints can be abused, some researchers have begun to question the value of hints (Muldner, Burleson, Van de Sande, & VanLehn, 2011; Ringenberg & VanLehn, 2006). They have wondered if reducing the student's use of hints, perhaps by providing alternative scaffolding such as a worked example, might be better.

This shows that improving an ITS might be easier if we knew how students used the system and in particular how they use its scaffolding. Unfortunately, as of now studies of student behavior while working on an ITS were confined to the gaming behavior of the

students and did not explore in general on how students used the overall system. This research is our initial attempt to study the behavior of students while working on computer-based tutoring systems, and we hope more studies are directed towards this neglected area, that can perhaps give more insights on how better tutors can be made.

Research Questions:

This research is an attempt to find out how students behave when using a step-based tutoring system (VanLehn, 2006), namely Andes (VanLehn, et al., 2005). Since something of this sort has never been done before, we focused on a more descriptive analysis of the data rather than one that focuses on testing hypotheses. Because learning with human tutors seems to occur mostly at impasses (VanLehn, 1999), we focused on episodes when students made an error and struggled to fix it. We characterized their behavior in two ways: (1) what did they do during the episode? And (2) what was the outcome of the episode, learning something new was found to be just one possible outcome.

Study Methods

This section describes the study and the methods used for collecting raw data. Discussion of the data analysis is deferred to the next section.

Participants

Ten paid volunteers, who had taken physics, participated in the two hour experiment. These volunteers were undergraduate students from a physics course having a working knowledge of solving problems in physics.

Materials

For the experiment, we used Andes 3, which is similar to Andes 2 (VanLehn et al., 1995) with an improved user interface, which is shown along with details in Figure 1. The user

interface of Andes 3 gave the users the same feel they would get when they are solving problems in pen and paper. They could draw bodies, axes, define quantities and write equations the same way they would do on a paper, except that Andes understood what they wrote and gave immediate feedback on each entry by turning it green if it was correct and red if it was incorrect. Andes would also provide hints when asked, and sometimes gave unsolicited hints as well.

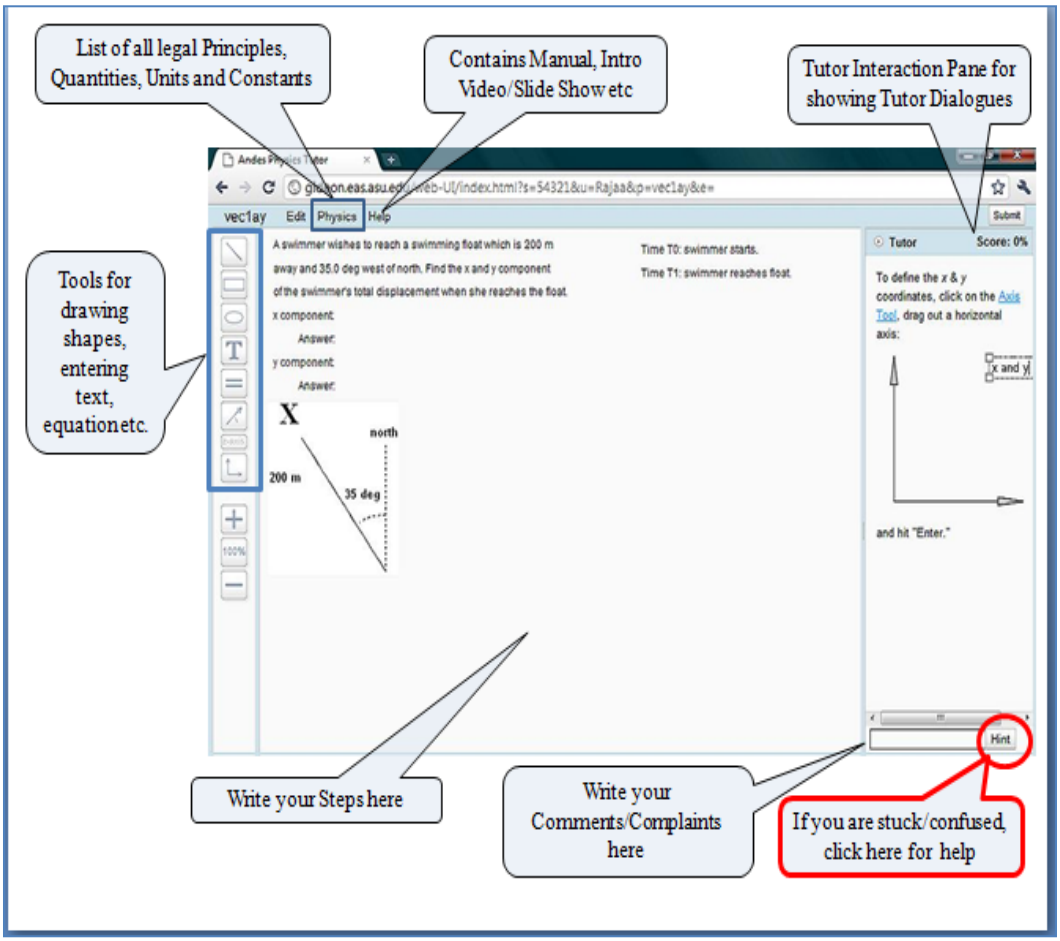

Figure 1.1: Andes3 User Interface

Unlike paper, Andes required students to follow certain conventions on what could be written. For instance, it required that dimensional numbers have units and that variables be defined before being used in an equation. These conventions clarify the solution and would be required, for instance, if the solution were published in a textbook. If the

students were working on paper, violating such conventions may cause a grader to take off points. Andes enforces these requirements not only because its designers believe that doing so improves student learning, but also because Andes can do a better job of understanding the student's writing if they follow the conventions. As the data we collected made clear, students were already familiar with many of Andes' conventions but were not in the habit of obeying them. Much of their learning turned out to be simply discovering that Andes' required them to follow these conventions.

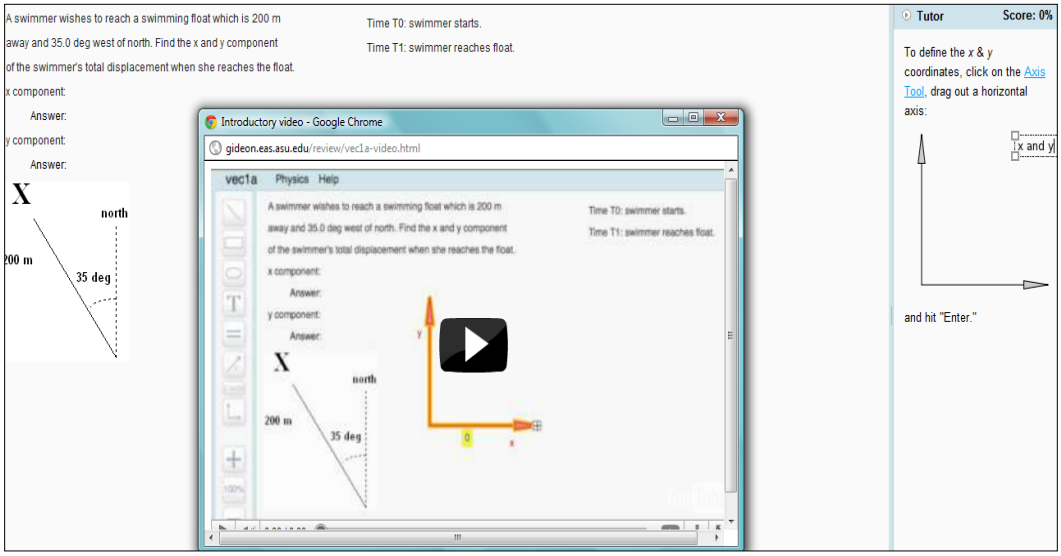

Figure 1.2: Introductory problem with the instruction video; on the right side, we can see the step by step instructions for solving the problem.

Students were given a sequence of physics problems to solve. All the problems used the concepts of Work and Energy. Examples of problems are shown in Figures 3 and 4. The Andes Help menu contained brief summaries of the relevant concepts and principles, including the relevant equations. Students did not have a textbook or any other source of physics information other than Andes.

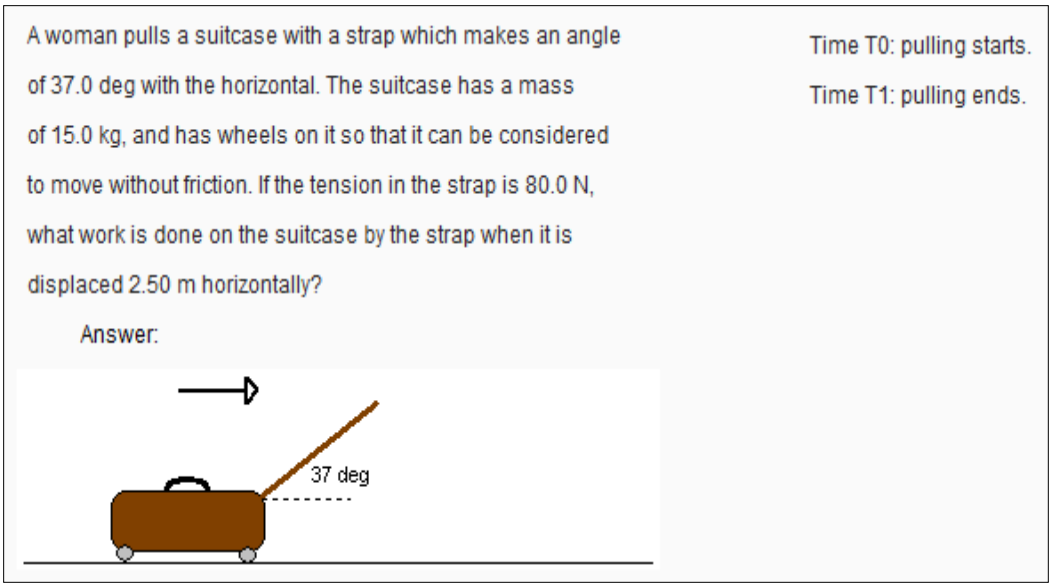

Figure 1.3: Problem 1

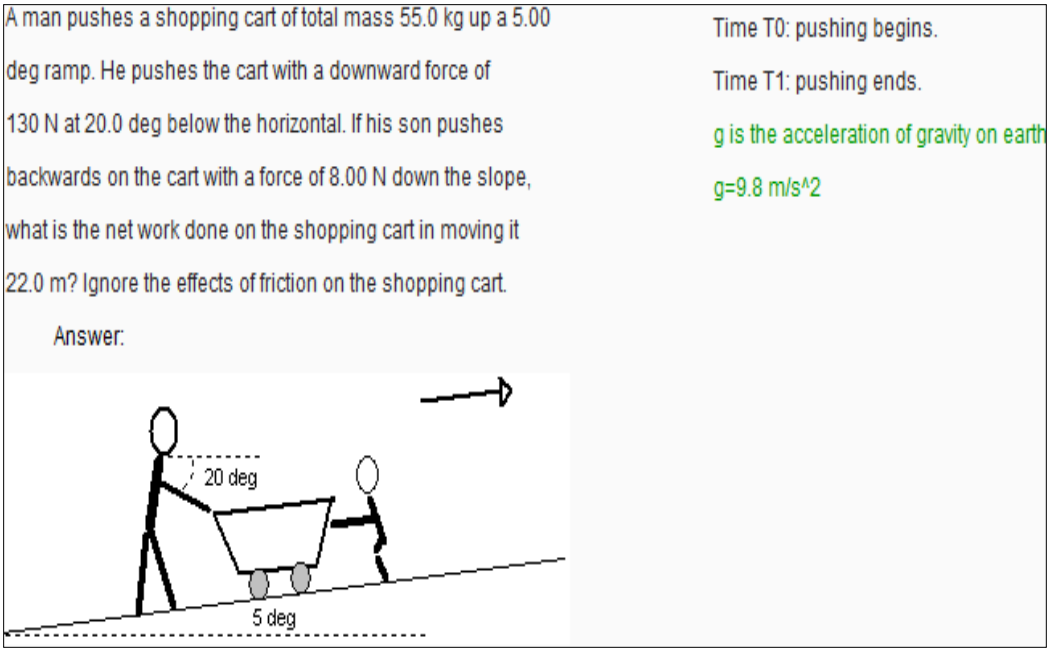

Figure 1.4: Problem 2

Procedure

After obtaining informed consent, students were trained on the user interface and the Andes' conventions. First they watched a short video that showed a step by step solution for a simple vector based problem (shown in fig 2) while explaining the tools and conventions of Andes. Next they were walked through solving the same vector problem.

That is, they were required to solve the problem with instructions given at every step. The video and the walk through comprised the whole user interface training. The video remained available under the “Help” menu item at the top of the screen, but students seldom referred back to it.

The students’ remaining time was spent solving physics problems. Students were asked to solve as many problems as they could in the remainder of the 2 hour period. They all worked through the same sequence of problems.

As they solved problems, students were asked to provide information on their cognitive and affective state by talking continuously about what they were thinking and how they were approaching the problem. The computer screen where the problem was being solved and the verbal protocol given by the students were recorded. We coded directly from the screen videos of the student solving problems while speaking aloud, and did not transcribe them.

Data Analysis Methods

The data analysis consisted of first segmenting the videos into episodes, then coding each episode, and finally counting the codes and aggregating.

In order to segment the video into episodes, we used the fact that Andes provides immediate feedback by coloring a student’s entry green if it is correct and red if it is incorrect. An episode was defined as starting from an incorrect (red) entry and ending when that entry was either entered correctly (green) or deleted. If the episode ended with correction of the initial entry, then it is called a “Correction” episode; if it ended with deletion of the initial entry, it is called a “Deletion” episode.

There was one special case of deletion which occurred rather frequently. Andes has three tools that involve typing: The *equation* tool is for typing in equations; the *vector* tool

is for drawing a vector and typing in its name and definition (e.g., “Let d be the displacement of the crate.”); and the *text* tool is for defining scalar variables (e.g., “Let m be the mass of the crate”). Students sometimes entered text that the coder recognizes as correct, but into the wrong tool. Andes turned the text red. Usually, the students deleted the red entry, clicked on the correct tool and retyped the original text into the new tool’s entry box. This was counted as a Correction episode. If they simply deleted the red entry and did not retype it into a different tool, then the episode was counted as a Deletion episode.

An example of an actual *Correction episode* is given in Table 1 and Figure 5. The student, who was working on the problem shown in Figure 3, made an incorrect entry (action 1 in the Table). This created a red text “d = 2.50 m” that stayed on the screen until action 6, when the student clicked on the red text, typed nothing, and pressed the Enter key, thus turning the text green and ending the episode. In between, the student defined a variable, thus fixing the error that made “d = 2.50 m” incorrect. Defining the variable was initially done incorrectly (action 2) and deleted (action 3), but the student asked for a hint (action 4) which enable the student to enter a correct definition (action 5). Although actions 2 through 5 fit the definition for an episode, we decided to keep the coding simple and not use nested episodes, so actions 2 through 5 were not considered an episode, but were instead consider to be part of the episode that starts with action 1 and ends with action 6.

Action No.	Student Action	Tutor Response
1.	“d = 2.50 m” (Using equation tool)	Red Color (Incorrect Entry)+ Unsolicited Hint: Undefined Variable d
2.	“Let d be the displacement of the crate” (Using text tool)	Red Color (Incorrect Entry)
3.	Delete “Let d be the displacement of the crate”	None
4.	Click on Hint Button	Hint Given: “Variables must be defined before being used in an equation. Vectors are defined with the vector tool and Scalars are defined with the Text tool”
5.	“Let d be the displacement of the crate” (Using vector tool)	Green Color (Correct Entry)
6.	“d = 2.50 m” (re-entered by clicking on the equation to edit it, making no changes, and pressing the Enter key)	Green Color (Correct Entry)

Table 1.1: Student Episode

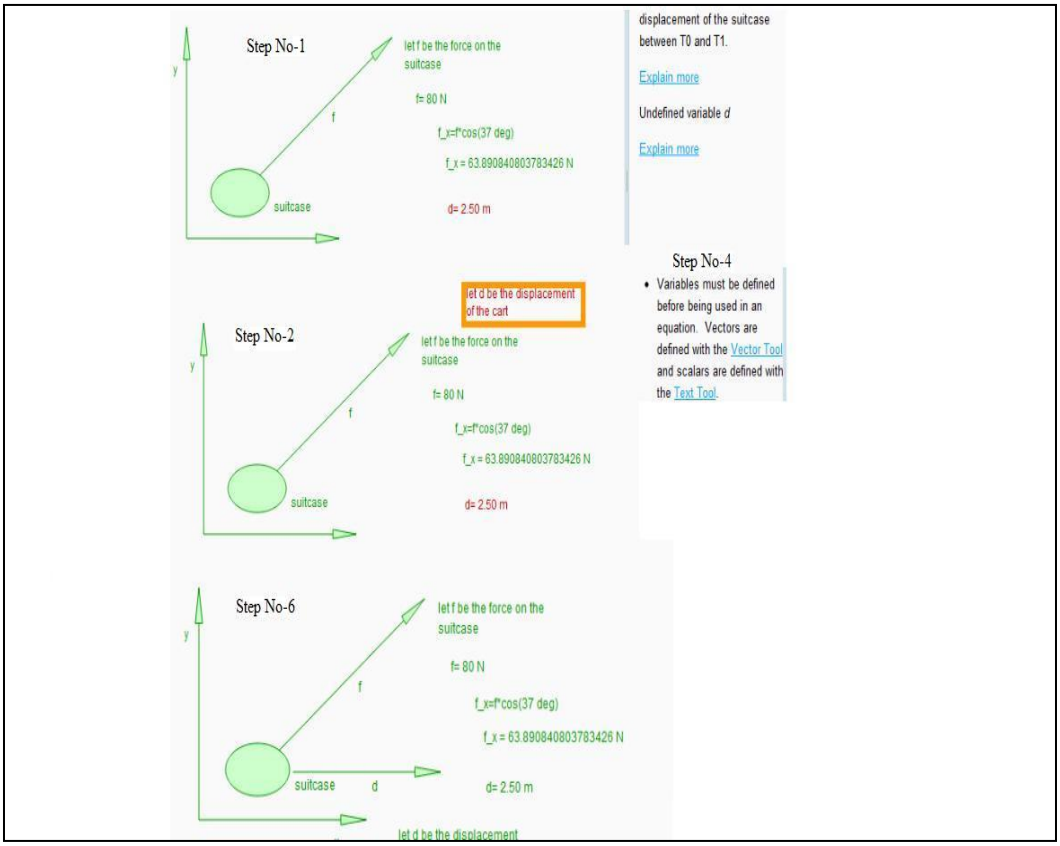

Figure 1.5: A Student Episode

An example of a *deletion episode* is shown in Table 2, which shows a student's initial actions when solving the problem, *"A man pushes a 20.0 kg crate across a frictionless floor with a horizontal force of 24.0 N. What work is done by the man on the crate in displacing it by 5.00 m?"* Action 1 is incorrect, thus starting an episode. The episode ends at action 7, when the initiating entry is deleted. This episode indicates again how several red entries can be part of an episode without themselves starting an episode.

Action No.	Student Action	Tutor Response
1.	"W=F*d" (Entered using the equation tool)	Red Color (Incorrect Entry), Unsolicited Hint: Undefined variables W,F and d.
2.	"W: Work" (Entered with the text tool)	Green Color (Correct Entry)
3.	"F: Force" (Editing the above entry)	Red Color (Incorrect Entry)
4.	"F: 24 N" (Editing the above entry)	Red Color (Incorrect Entry)
5.	"F: 24.0 N" (Editing the above entry)	Red Color (Incorrect Entry)
6.	Delete "F: 24.0 N"	None
7.	Delete "W=F*d"	None

Table 1.2: A Deletion Episode

Coding Conventions

We defined four categories for coding the episodes:

1. Outcome
2. Behavior
3. Struggling
4. Deletion type

An episode could have at most one code from each coding category. Every *Correction episode* received a code for its Outcome and the students' Behavior and was also given a yes/no code for Struggling. Every *Deletion episode* was given a code for Deletion type as well as the other three categories. A fifth category, Bug, was used to indicate whether an episode displayed a bug or other problem with the Andes software that should be fixed. It will not be considered further here. Each subsection below describes a coding category and its codes.

Outcome

The Outcome coding category represents the outcomes of the student for a particular episode, including either learning or the reason that they failed to learn anything.

Learning New Physics: This code represents that the students learned some physics that they appear not to have known earlier. As we had the audio/video of the students solving the problems, their comments often indicated when they were learning new physics, which is very difficult to determine by using log file analysis.

For example, Figure 6 shows a scene where the student must find the magnitude of the final velocity (v), given the initial velocity (u), the acceleration (a) and the displacement (d). Now, this can be done directly using the equation v^2 = u^2 + 2*a*d. The student had defined all the necessary quantities for solving this problem but he was not aware of the equation. This equation is not always taught, because it can be derived from other kinematics equations, which may be why this student was unfamiliar with it. After entering an incorrect entry and fussing around for awhile, he opened the Andes Help menu, navigated to the list of principles (shown on the left side of Figure 6), and saw the equation. After the state shown in Figure 6, he applied the equation and the entry turned green, ending the episode. In this episode, he seems to have learned a new kinematics equation. We are sure of this because we could hear him saying that he wanted an equation that connects velocity, acceleration and displacement. And the moment he opened the principles menu and saw the equation, he made a remark “Oh, v^2 = u^2 + 2*a*d”, which implies that he got to know of a new equation in physics that he did not know before.

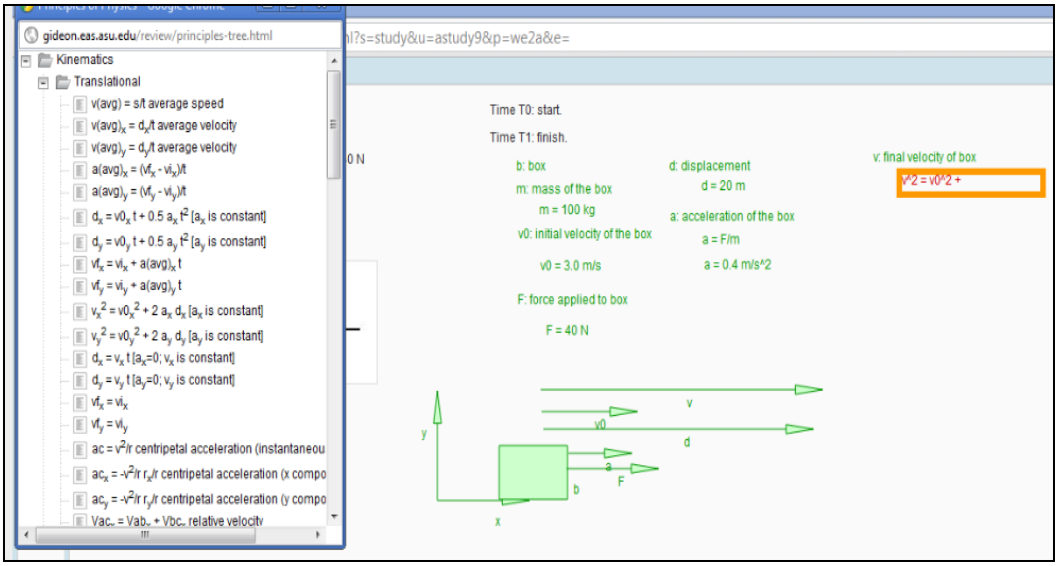

Figure 1.6: Learning new Physics

It should be pointed out that we did not expect physics learning to be frequent, because the students had already studied the appropriate physics content. In fact, they had already taken the midterm exam prior to this study, which covered Work and Energy.

Learning Andes Conventions: As mentioned earlier, Andes expects students to follow certain conventions while solving problems. For instance, it expects the students to define a variable before using it in equations. Moreover, vectors must be defined using the vector tool; scalar quantities must be defined using the text tool; and equations must be written using the equation tool. This coding category represents episodes in which students appear to learn that Andes requires complying with a convention.

As an example, consider the problem shown in Figure 3 where students must define the displacement of the suitcase using the vector tool. Figure 7 (which has multiple parts) shows the key parts of a scene in which the student defined displacement using the text tool (Figure 7.1). The first hint (Figure 7.2) was “Note that displacement is a vector quantity,” which doesn’t help the student because the student is not yet aware of its implications for selecting a tool. The student clicked on the “Explain more” button, which

told him to use the vector tool for defining the displacement since it is a vector quantity. The student deleted the original red entry and defined a vector for displacement (Figure 7.3). In this process of correcting his red entry, he seems to have learned something new about Andes conventions. Note that this convention was mentioned in the introduction video shown to the students before they start solving.

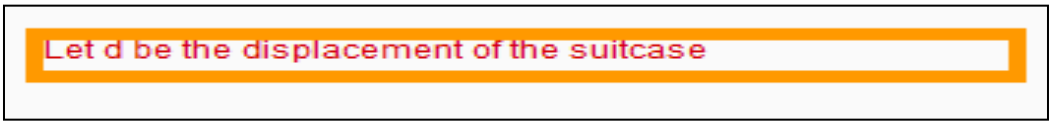

Figure 1.7.1: Learning Andes Conventions: This shows the displacement defined by the student using a text tool.

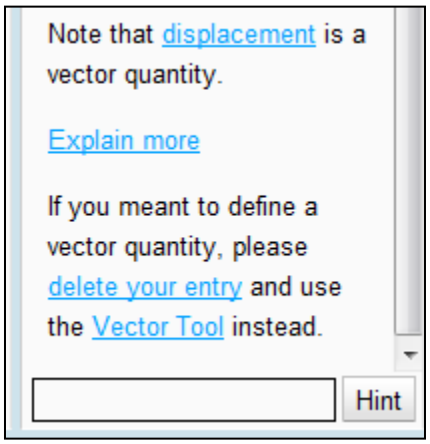

Figure 1.7.2: Learning Andes Conventions: Hints provided by Andes

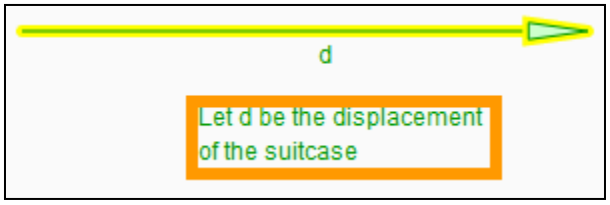

Figure 1.7.3: Learning Andes Conventions: Correction made by the student, redefining displacement using vector tool.

Apply Existing Knowledge: This code represents episodes where students already knew something, but forgot to apply it. Students often mention this, even when they must ask for a hint and it reminds them of the knowledge they "knew" but failed to apply. For example, consider Figure 8. Here the student has solved a problem completely, but while entering his final answer in the answer box, he forgot to put units to the answer, which is required according to the conventions followed in Andes. He got an unsolicited

hint which said “Forgot to put units on a quantity.” In this case, he knew about the Andes convention that he was supposed to have put the units on the quantity, but somehow he forgot about the same, but realizes it and quickly corrects the answer. The entire episode takes less than 8 seconds.

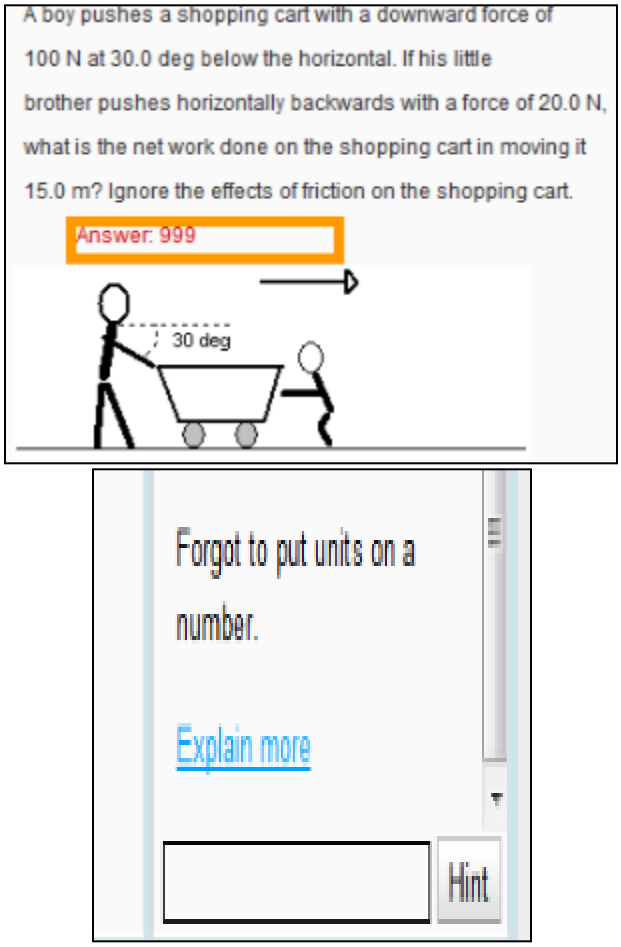

Figure 1.8: Applying existing knowledge: Did not put units to a Quantity and the corresponding hint given by Andes.

Guessing the correction: This code represents episodes in which students correct the red entry that started the episode, but they seem not to know why their revised entry was correct. This can occur with Andes conventions that are less commonly applied to paper-based work and thus more likely to be unfamiliar to students. However, our example shows an instance where a bug in Andes is worked around by guessing.

Figure 9.1 shows an episode that starts when the student has drawn coordinate axes at an angle of 5 degrees to the horizontal. Although this is a reasonable choice for this problem, Andes does not recognize it as such (this is a bug in Andes that has since been corrected). When the student's axes turn red, he does not ask for help from Andes; instead he just changes the axes to zero degrees (Figure 9.2). It is likely that he has no idea why he must do that, so this episode is has a Learning Outcome code of Guessing.

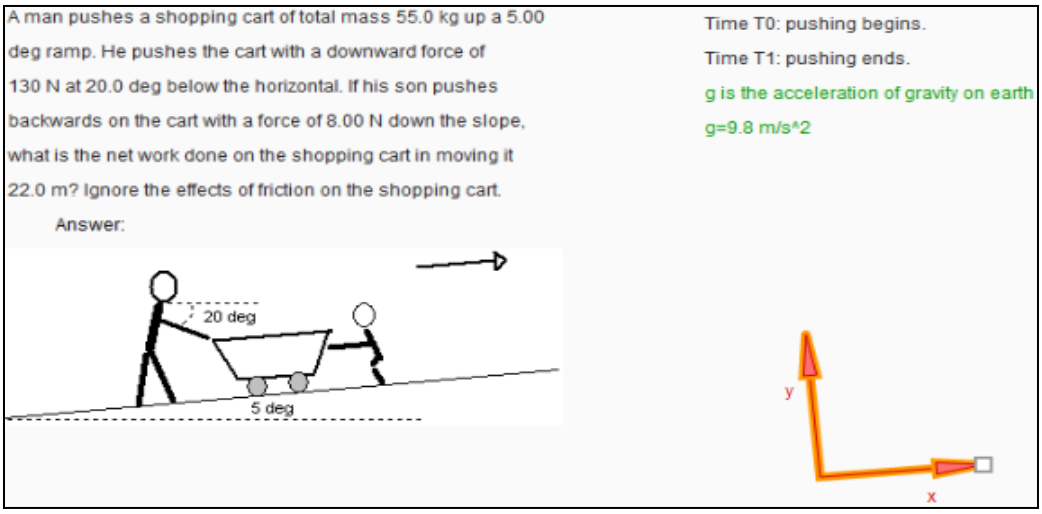

Figure 1.9.1: Guessing: Axes drawn incorrectly

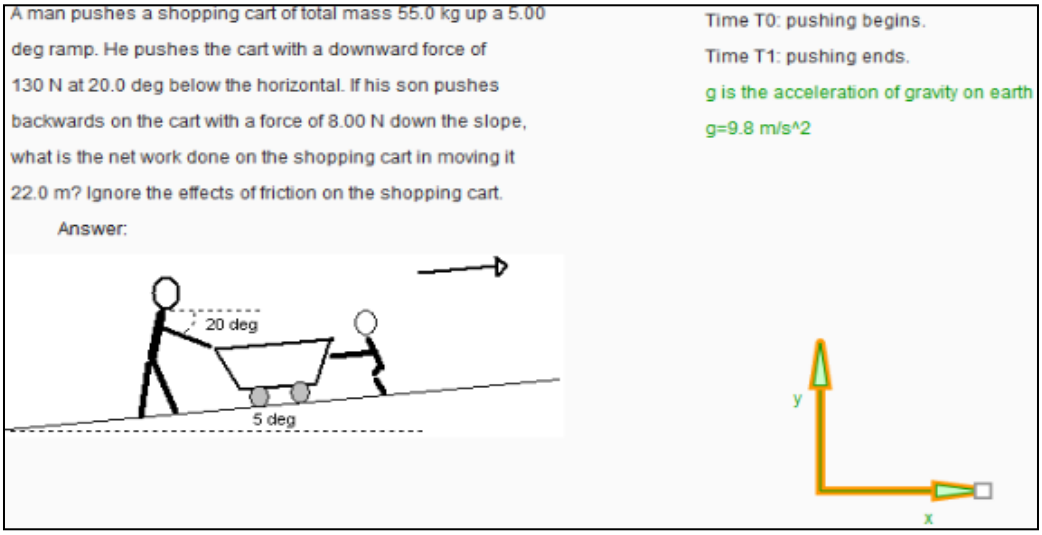

Figure 1.9.2: Guessing: Correction

Guessing or applying knowledge: There are some episodes in which it is not clear even from the audio/video if they were guessing or using the knowledge that they already have. All such episodes fall into this category.

Copying a bottom-out hint: Help is provided in Andes as a sequence of hints, starting from the most general hints and moving towards more specific ones. The purpose of the hint sequence is to make the student think for some time after every hint and see if he can correct his mistakes. The last hint in this sequence known as bottom-out hint, would normally give away the next step to the student. There are some episodes in which the student would just copy the bottom-out hint without putting any effort himself thinking about what could have gone wrong. Those episodes come under this category.

Miscellaneous corrections: There are some episodes which would somehow not fit into any of the above categories. They have been put in this category.

Struggling

In the process of solving problems, whenever the student makes a mistake and encounters a red entry, there are a series of steps involved in correcting the entry. Sometimes, they realizes their mistake on their own and corrects it in no time; sometimes the unsolicited hints give them a clue on what could have gone wrong and then they manage to rectify it themselves rather quickly; but there are times when they struggle for quite some time either making many attempts on their own or asking for help continuously and trying to correct their mistake. Some of these cases may be due to unclear hints or bugs in Andes, and other times it is the student's mistake that they don't ask for help when they are stuck at some place for a long time. (This phenomena is known as Hint Refusal in the ITS literature). Therefore, we decided to have a coding category that represents the effort of the student in correcting his mistakes, i.e. if he realized and rectified the mistake quickly or if he struggled for a long time either due to hint refusal or because of unclear hints/bugs in Andes. This can actually help us find the reasons why students struggled and so we can take actions to reduce this.

For every Correction episode, this code will represent if the student struggled to make his red entry green, or if he could do it easily. For simplicity, we used only two values: yes or no.

Behavior

On encountering an incorrect entry the student is expected to correct it. There are different ways in which he can attempt to correct the incorrect entry. This category represents them.

Asked for all hints, rapidly: This is the code for hint abuse, in which students rapidly keep clicking on the hint button till they reach the bottom-out hint, without thinking at any stage about what could have gone wrong. The moment they reach the bottom-out hint, the answer is right in front of their eyes and they simply copy it. This category is given to all such episodes. One such example is shown below in Fig 10. Consider the problem shown in Fig 3. The screen-shot below shows an intermediate state when a student is solving the problem.

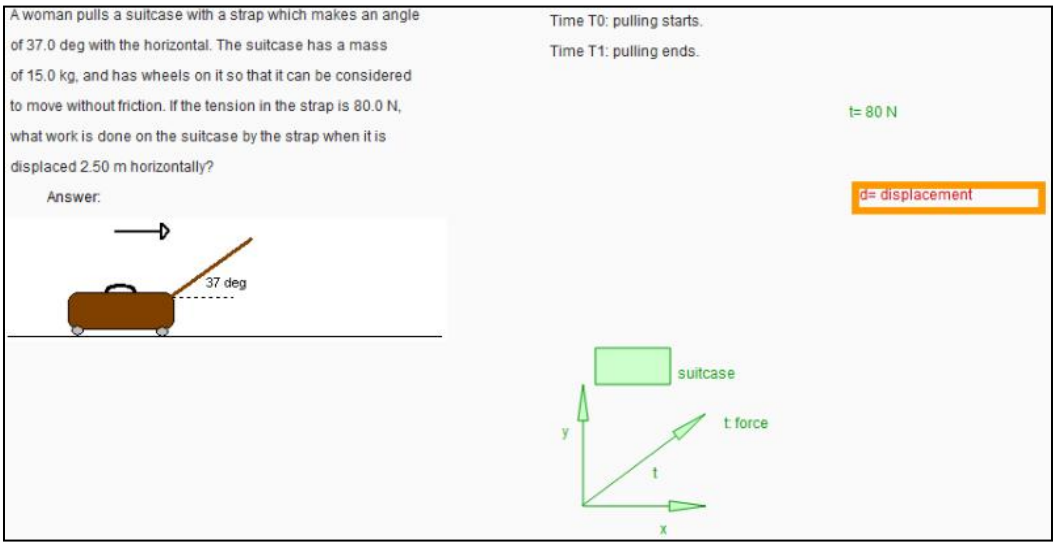

Figure 1.10.1: Hint Abuse - Initial problem scene

The goal of this problem is to find the work done (say, W) when given the force (F), the displacement (D) and the angle (theta) between F and D. This is as simple as applying the equation W=F*D*cosine (theta). At this point (shown above), the student has defined the force and he is attempting to define the displacement. But he uses the equation tool for doing so which is incorrect since displacement is a vector and it has to be defined using a vector tool. At this point, he gets an unsolicited hint and he keeps clicking his way through till he reaches the bottom-out hint as shown in fig 10.2 below. After he sees the bottom-out hint, he defines a vector for the displacement. Since we had the videos we could see that he did not even bother to read the intermediate hints; he just kept clicking on the hints till he reaches the last hint which gave away the answer.

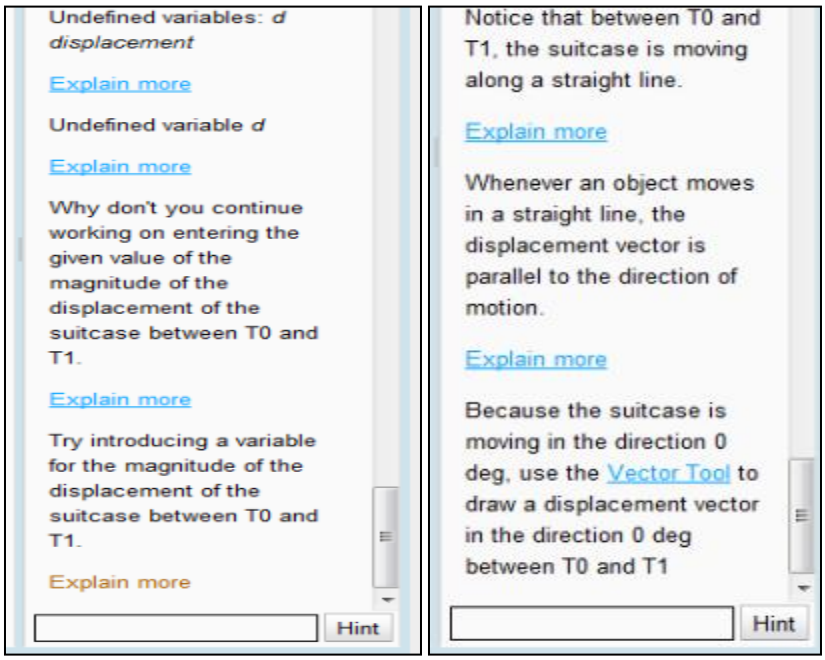

Figure 1.10.2: Hint Abuse – Sequence of hints till the bottom-out hint.

No hints, much guessing: Sometimes when students get a red entry, instead of asking for help, they keep editing the answer till the point when it becomes correct. This is sometimes referred to as hint refusal. Consider the episode shown below (Fig 11):

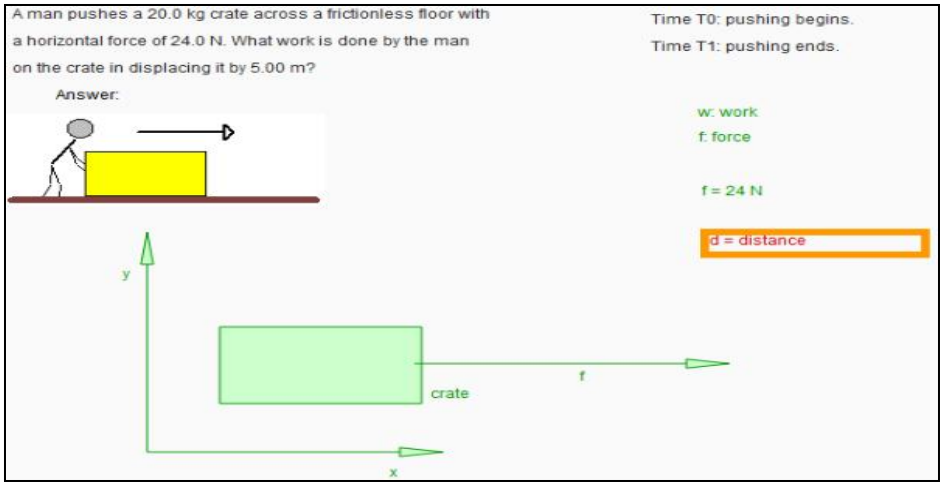

Figure 1.11.1: Hint Refusal-Attempt 1

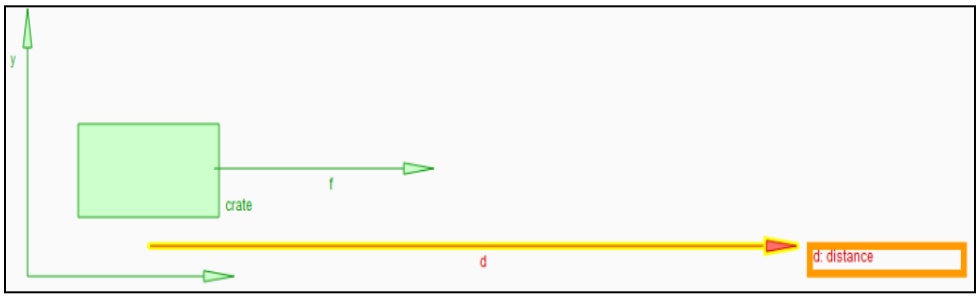

Figure 1.11.2: Hint Refusal-Attempt 8

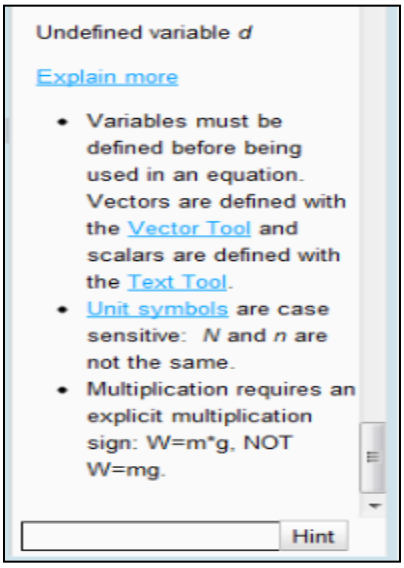

Figure 1.11.3: Hint Refusal-Useful hint shown to the student

Here the student is trying to define the displacement. Displacement has to be defined using the vector tool. Here the student has made two mistakes, one is trying to define displacement as distance and the other is using the equation tool instead of a vector tool.

The student gets an unsolicited hint which he seems not to understand. Although he could have clicked on “Explain more,” he instead keeps attempting on his own to correct his mistake. He made 7 attempts using the incorrect tools and before finally using the vector tool (fig 11.2). However, he still got it wrong because he used distance instead of displacement. Eventually he got frustrated (in fact he himself mentioned this as heard from the audio captures) and decided to read some of the hints provided. He finally he got a hint (fig 11.3) which gave him the idea that he was supposed to have used the vector tool for defining the displacement, and consequently he gets the correct answer. Overall, he made more than ten attempts to correct his mistake and eventually got totally frustrated. All this could have been avoided if he had asked for more hints in the initial stages instead of guessing so many times.

Asked for some hints: This code represents those cases in which the student gets a red entry and rectifies it using the hints given by Andes. An example has already been mentioned in the ‘Apply Existing Knowledge’ section above. There, the student forgot to put units for the work done and he got an unsolicited hint saying ‘Forgot to put units on a number’. On seeing this, he realized his mistake and corrected it. Apparently, the hint was effective in allowing him to correct his mistake.

Referred to example: As already mentioned, the very first Andes problem given to all the participants of this study is a walkthrough problem i.e. it gave step by step instructions to the student. If a student made any mistake, all he had to do is to look at the step by step instruction given to him and he would know his mistake. Since the problem is solved by the tutor itself (the solution is shown in the form of the walk-through), and the student refers to this information we decided to name this category as ‘Referred to Example’. Consider the episode shown below (Fig 12). Here instead of displacement, the student used the term ‘distance’ which is incorrect. In the tutor pane, we can see the step by step instructions for solving this problem. There is a step that tells him to draw a vector and

label it as‘d is the displacement of the swimmer’. The student reads this and corrects his mistake.

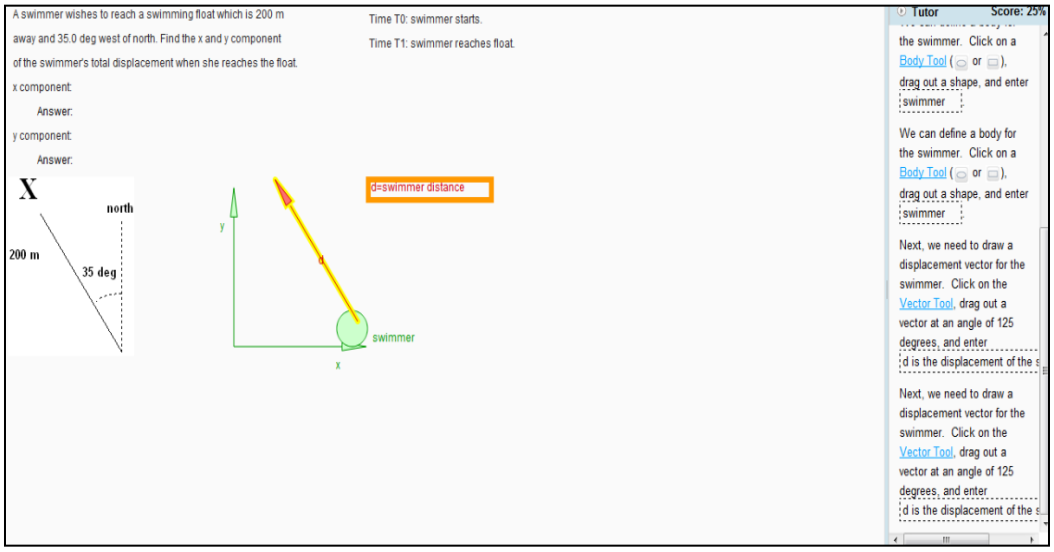

Figure 1.12: Refer to Example

No hints, little guessing: This category represents cases where after getting an incorrect entry, the students themselves correct it almost immediately without taking any kind of help.

Asked expert for help: There are some episodes in which the student is stuck at some point without knowing how to proceed or without knowing how to rectify a red entry. If he gets very frustrated and is in the verge of giving up, it is better for the experimenter to provide some help so that he can at least move forward. In most of these cases, students get stuck because of some bugs or bad hints. Therefore it is mandatory for the experimenter to provide help; else it would be extremely difficult for the student to proceed on his own. All such episodes fall under this category.

Andes Defect: These represent episodes where it is impossible for the students to correct their mistake due to some defect present in Andes. This code is applicable only to the Deletion episode, because the only way to proceed is to delete the entry and work around it. Consider an example given in fig 13. Here the student tried to define the force

of gravity due to the earth using a vector tool. Instead of recognizing what the student was trying to do, Andes recognized this force wrongly as the external horizontal force applied on the box (which had already been defined by the student) and marked it red. The student tried to change this entry several times thinking that he might have used incorrect wordings, but the system always recognized the student's entry as the external force only, finally forcing him to delete his entry. Hence, we can say that the cause of deletion was the defect present in Andes.

Figure 1.13.1: Andes Defect 1

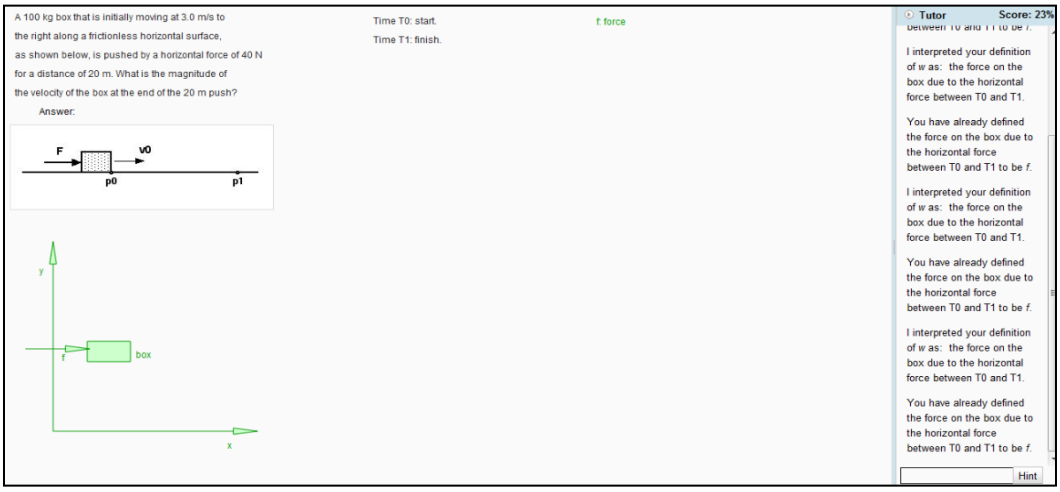

Figure 1.13.2: Andes Defect 2

Others: This code is for Correction episodes that do not fall into any of the above categories come under this one.

More examples of Correction episodes

In order to illustrate some more details in the coding scheme, below are some episodes showing some Correction episodes and their corresponding codes.

Consider the episode shown below (Fig 14):

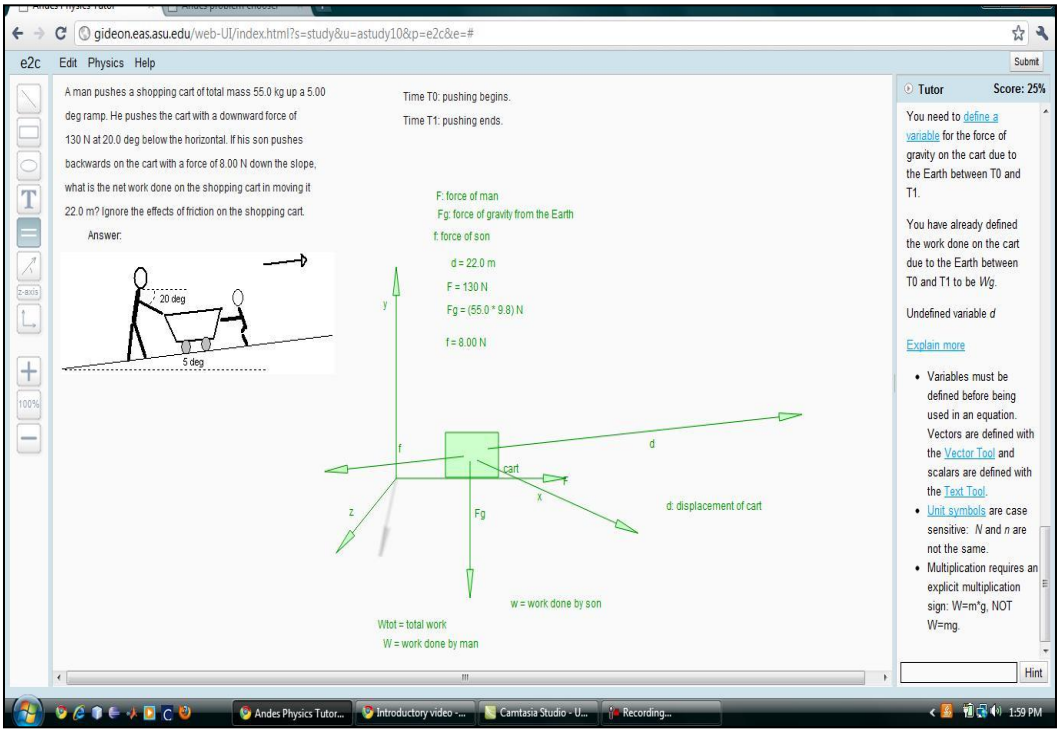

Figure 1.14.1: Learning New Physics

Here, the student has defined everything needed for solving the problem, and just has to write the equation for the net work done on the shopping cart. The equation for work is w=f*d*cos(Theta), where f is the force exerted by the man on the cart, d is the displacement of the cart and Theta is the angle between f and d. Many students do not have a clear idea of what Theta represents, and end up making mistakes here. In this case Theta is in fact 25 degrees, whereas most of the students write it as 20 degrees or 340 degrees, thus leading to a red entry. But, in this case, the student made use of the hints well, and landed up in the Principles menu (Fig 14.2), which gave an alternate equation for work as w=fx*dx + fy*dy, as shown below:

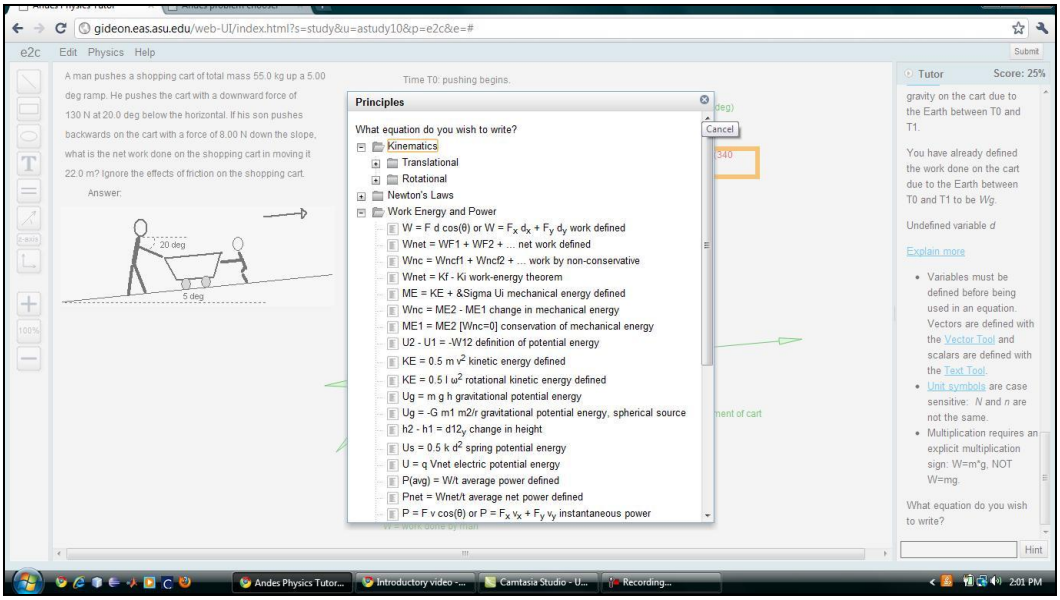

Figure 1.14.2: Principles Menu

On using this form of the work equation, he got the correct step. Therefore, the code for the Outcome category is 'Learning new Physics'. Here he went through the hints very smoothly, therefore the code for the 'Struggling' category is 'No', which means that he did not have to struggle to get the correct result, and finally the code for the category 'Behavior' is 'Asked for some hints', since it was the hints that made him see an alternate form of the work and energy equation and get things correct.

Next, consider the episode shown below (Fig 15):

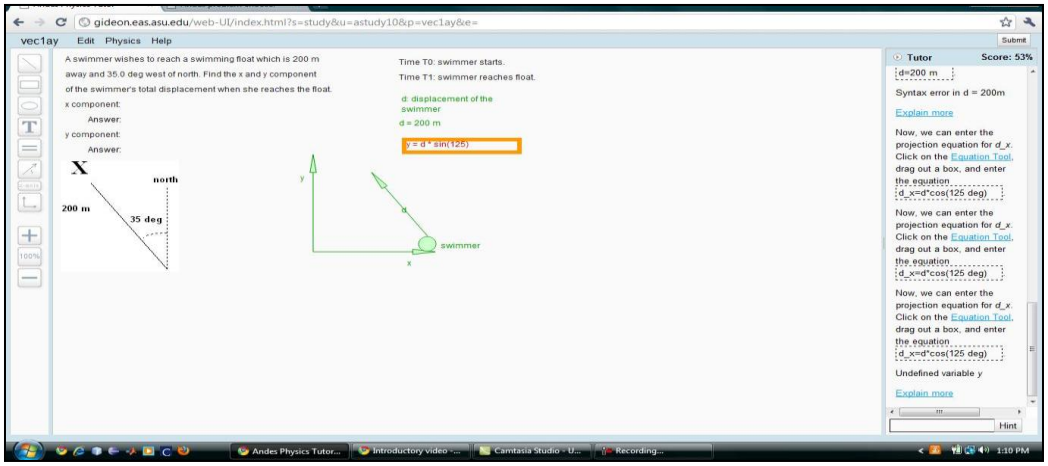

Figure 1.15: Learning Andes Conventions

Here, the student is working on the introductory problem where the system gives him the step by step instructions for solving the problem. The student should have written d_y = d*sin (125 deg), but instead he wrote it y=d*sin (125), which is incorrect according to the Andes conventions. Now, in this case, on the right hand side, an unsolicited hint was given to him, which says that variable y is undefined. Also, the correct way to write the equation is shown in the instructions. He looks at the example and corrects his equation. He did not spend much time, since the answer was right before his eyes. Therefore the code for the 'Outcome' is 'Learning Andes Conventions', 'Struggling' was 'No', and the 'Behavior' is 'Refer to Example'.

Next consider the episode shown in Fig 16: Here, the student wrote the equation as W=F*d*cos(330) instead of the correct equation W=F*d*cos(330 deg). He forgot to mention 'deg' which is the unit for degrees. Because he already had written many equations prior to this, he realized his mistake himself and quickly corrected the error. Therefore the code for 'Outcome' is 'Apply Existing Knowledge', 'Struggling' is 'No' and 'Behavior' is 'No hints, little guessing.'

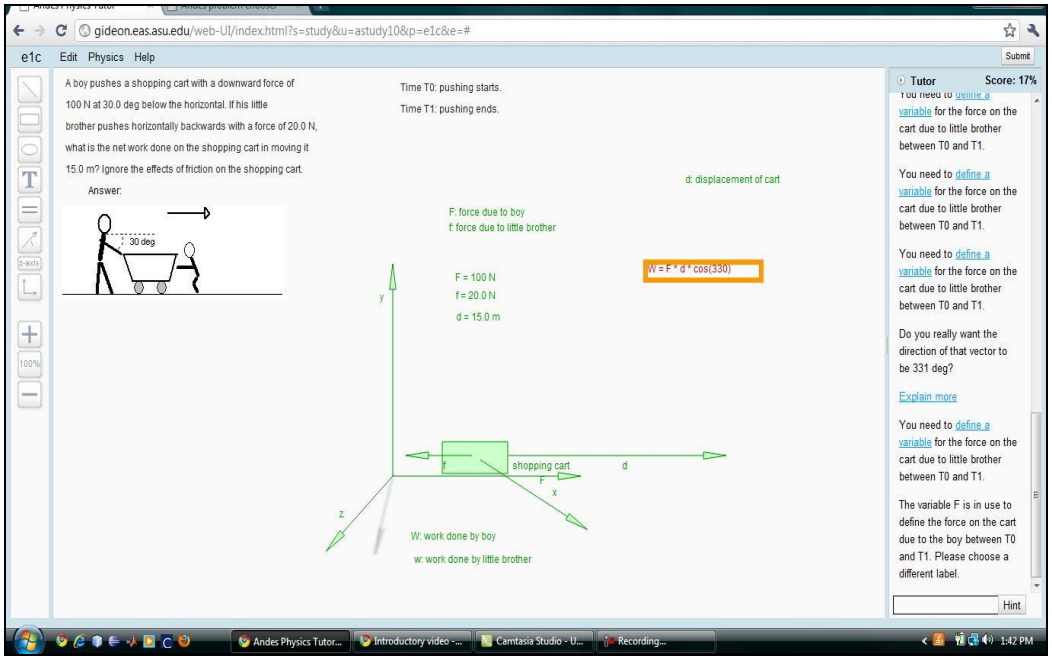

Figure 1.16: Apply Existing Knowledge

Next consider the episode shows below (Fig 17):

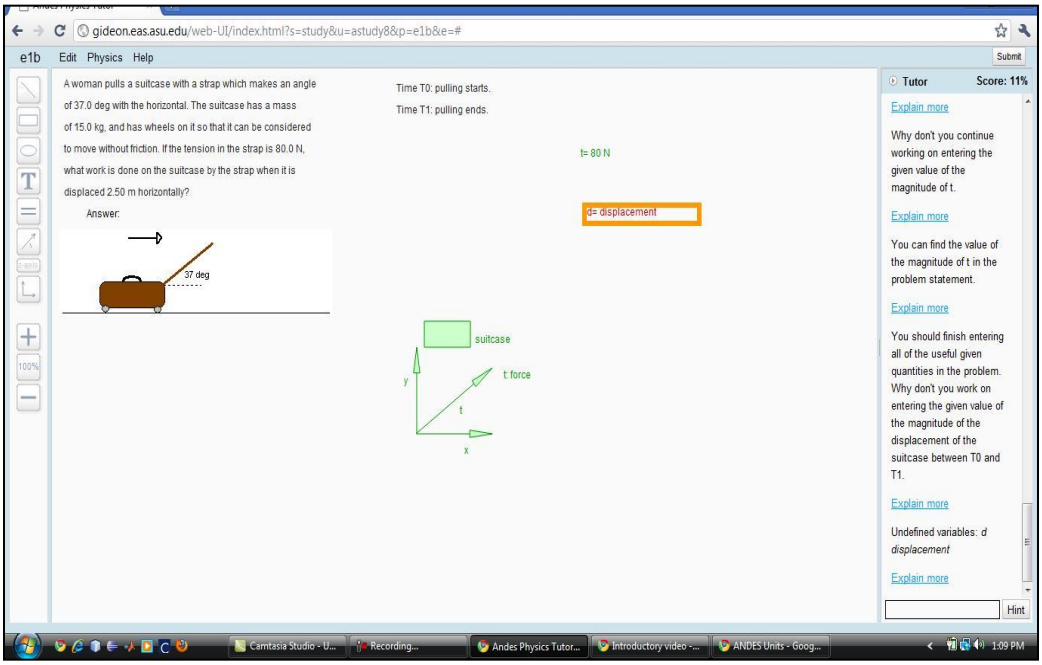

Figure 1.17.1: Copying the Hint

Here, the student used the wrong tool to define the displacement of the shopping cart. Displacement is a vector and must be defined using the vector tool. Instead of trying to find out what was wrong, he kept clicking on the hint button (shown in the screen shot

below), which eventually led him to the bottom-out hint and then he used the vector tool and wrote the equation correctly. Therefore, the code for the 'Outcome' is 'Copying the bottom-out Hint', 'Struggling' is 'No', and 'Behavior' is 'Asked for all hints, rapidly'.

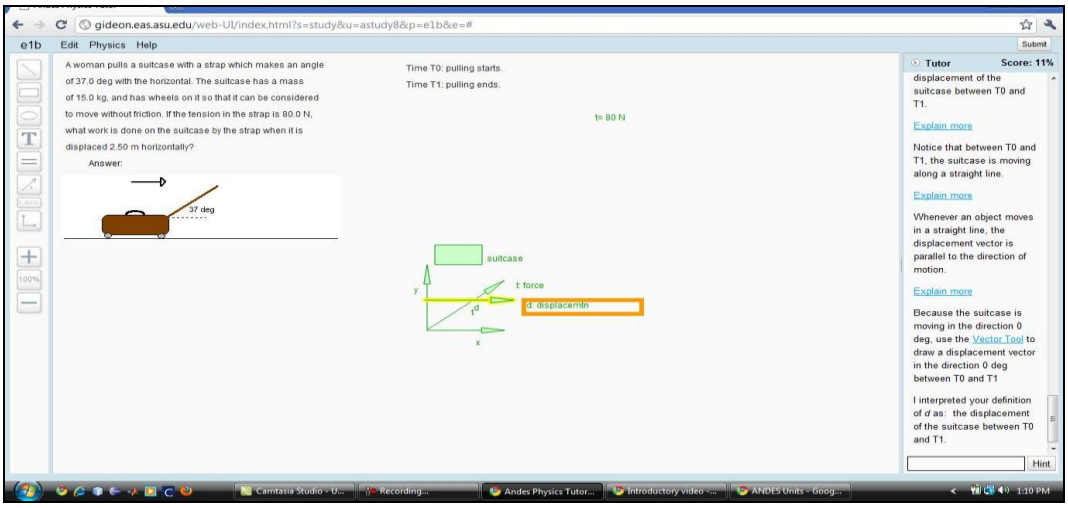

Figure 1.17.2: Copying the Hint

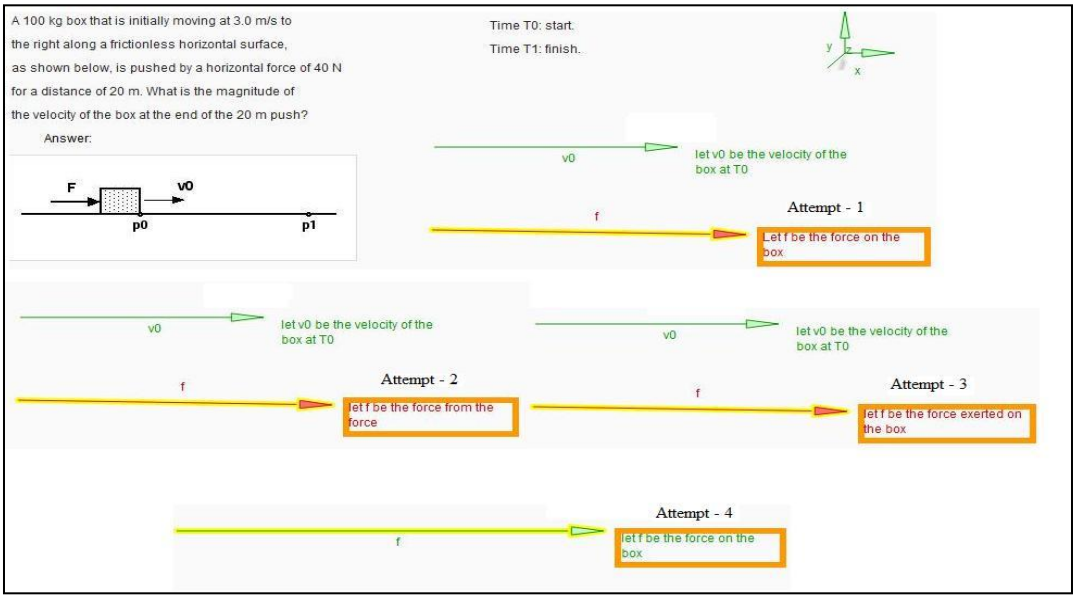

Figure 1.18: Guessing

Next consider the episode shown in Fig 18. The student was trying to define a horizontal force, but accidently he drew it slightly tilted. The student tried to fix it on his own and got it right in his 4th attempt. Instead of taking a hint, he kept trying out different definitions

and finally one of them turned green. Therefore the code for the 'Learning-Outcomes' is 'Guessing the Correction', 'Struggling' is 'Yes' and 'Type' is 'No hints, much guessing'.

The Deletion Type category

There are many episodes in which the student encounters a red entry and instead of correcting it, he deletes it. The Deletion Type category represents the reason the entry was incorrect.

Incorrect Tool Used: As already mentioned, Andes expects students to follow some conventions while solving problems, one of which is to define quantities before using them in equations for solving the problem. Vector quantities are defined using the vector tool and scalar quantities are defined using the text tool, and equations are using the equation tool. Since there are different tools for different purposes, students sometimes get confused and end up using an incorrect tool. In those cases the student is forced to delete the entry and then use the correct tool and write the same entry again. These cases fall into this category. An example of this was already shown in Table 2.

Incorrect: This category represents all other cases. Typically, the student's entry was conceptually wrong, i.e. it was a physics error, and he deleted the entry instead of taking a help and correcting those.

Data analysis and Results

The screen capturing and audio recording were done using Camtasia Studio. The coding was done using Elan. Elan allowed the creation of different tiers so each tier was used for coding different category. An episode could easily be identified and the boundaries marked, and an annotation given to each tier thus representing the code for that particular category.

A total of 283 episodes were coded for the 10 students, or 28.3 episodes per student on average. Table 3 displays all the episodes and how they were coded along the most important two categories. The columns show the process code and the rows show the outcome code. The table includes both Correction and Deletion episodes, and distinguishes them by including "Deleting the red entry" as an outcome. All the other Outcome codes (rows) correspond to Correction episodes.

Table 3: Classification of behaviors (columns) and outcomes (rows)

	No hints, much guessing	No hints, little guessing	Asked for all hints, rapidly	Asked for some hints	Referred to example	Asked expert for help	Other	Andes defect	total
Learning new physics	0	2	0	1	0	4	0		7
Learning Andes conventions	5	6	0	30	11	17	3		72
Applying existing knowledge	2	18	0	56	2	0	0		78
Guessed the correction	4	14	0	2	0	1	0		21
Guessing or applying knowledge	1	9	0	5	0	0	0		15
Copying a bottom-out hint	0	0	11	1	2	3	1		18
Deleting the red entry	32	0	0	6	0	0	17	9	64
Miscelaneous correction	2	1	0	3	0	2	0		8
Total	46	50	11	104	15	27	21	9	283

Table 1.3: Classification of behaviors (columns) and outcomes (rows)

In order to draw some conclusions based on proportions, we aggregated some of the coding categories (shown as rectangular boxes with heavier lines in table 3). "Learning new physics" and "Learning Andes conventions" implies that the student learned something. "Guesses the correction", "Guessing or applying knowledge", "Copying a bottom-out hint" and "Deleting the red entry" implies that the student lost the opportunity to learn something. "No hints, much guessing", "No hints, little guessing" mean that the student never asked for help, where as "Asked for all hints, rapidly", "Asked for some hints" and "Referred to example" mean that the student asked Andes for help. Table 4 shows such a coarser classification. The integers in the cells are the sum of the cells in the corresponding heavy box in Table 3. For instance, the 13 in the cell [Learn, Never ask] is the sum of the four cells in the upper left of Table 3 (0, 2, 5, 6).

Table 4: Coarse classification of behaviors (columns) and outcomes (rows)								
	Never ask		Ask Andes		Ask expert		Total	
Learn	13	17%	42	55%	21	28%	76	31%
	14%		33%		84%			
Apply knowledge	20	26%	58	74%	0	0%	78	32%
	22%		46%		0%			
Lost opportunity	60	66%	27	30%	4	4%	91	37%
	65%		21%		16%			
Total	93	38%	127	52%	25	10%	245	

Table 1.4: Course classification of behaviors (columns) and outcomes (rows)

The bottom row of Table 4 indicates that students often asked Andes for help (52%) and seldom asked the expert for help (10%), which is reasonable, as that is what they were asked to do. However, even though they were trying to correct a red entry, students often did not ask for help at all (38%). This usually turned out badly for them. Of the episodes where they never asked for help, 65% resulted in them either deleting the red entry, guessing a correction without understanding it, or in some other way losing the opportunity to learn. Only in a few cases (14%) did they manage to learn from their guessing.

Now let's consider the 127 episodes where students referred to Andes for help, either by asking for a hint or by referring to its walk-through example. When students faced a red entry, they often (52%) asked Andes for help. Of these episodes, most were beneficial in that the student learned something from the hints (33%) or at least were reminded of something they already knew (46%). Only in a few cases (21%) did students fail to profit from asking Andes for help. Often (11 of 27 episodes) the failure was partly the students' fault because they abused the hints by skipping though them quickly until they got to the bottom out hint, then simply copying it without thinking hard. In short, the failure rate when student ask Andes for help is 21% whereas the failure rate when they ask for no help is three times larger: 65%. This suggests that students who are inclined not to ask for help should be somehow nudged into asking for more help.

Now for the 25 episodes where the students asked the expert for help, the failure rate is 16%, which is not far off from the failure rate for Andes' help (21%). This is consistent with meta-analytic results suggesting the human tutoring is no more effective than step based tutoring (VanLehn, in press).

Another way to look at the data is to find out how various outcomes occurred. Let's first examine how the learning occurred. Reading across the top row of Table 4 we see that of the 76 episodes where learning occurred, most (55%) occurred when Andes gave a hint. Of the 78 episodes where students first got a red entry then recalled the correct knowledge and applied it, they asked for a hint on most of them (74%). Once again, this shows that successful learning and practice after receiving a red entry comes mostly when students refer to Andes for help.

The other classifications did not reveal much of interest. Deletion episodes were classified into those caused by using the wrong tool (14 episodes) and those caused by entering the wrong content with the right tool (50 episodes). There was no apparent difference in the distribution of outcome codes across these two categories.

Correction episodes where divided into ones where the students struggled (47) and ones where they did not struggle (172). Table 5 shows the classifications in terms of behaviors and outcomes of the Correction episodes classified as struggling. The bottom row confirms what one would expect intuitively, that struggling episodes are characterized by not asking for help at all or asking for the expert's help. If the student sought help from Andes and this sufficed for correcting the red entry, then the episode was not coded as struggling.

Table 5: Classification strugging correction episodes								
	No hints, much guessing	No hints, little guessing	Asked for all hints, rapidly	Asked for some hints	Referred to example	Asked expert for help	Other	total
Learning new physics	0	1	0	0	0	2	0	3
Learning Andes conventions	5	4	0	3	0	11	3	26
Applying existing knowledge	1	1	0	0	0	0	0	2
Guessed the correction	3	3	0	0	0	1	0	7
Guessing or applying knowledge	0	2	0	0	0	0	0	2
Copying a bottom-out hint	1	0	4	0	0	1	0	6
Miscelaneous correction	1	0	0	0	0	0	0	1
Total	11	11	4	3	0	15	3	47

Table 1.5: Classification of struggling correction episodes

Discussion

The main conclusion seems to be that asking Andes for help is a good idea, and students should do it more often. If students asked for help (127 episodes) then they often learned something new (33%), or were reminded of something they knew already (46%), and the knowledge allowed them to correct the red entry that initiated the episode. On the other hand, if the students tried to work out the correction on their own without asking either Andes or the expert for help (93 episodes), then they often failed to learn or apply knowledge (65%) and often ended up just deleting the red entry.

This is not to say that Andes' help is perfect. It was sometimes abused (11 of 127 episodes), and sometimes resulted in failures to learn or apply knowledge (16 of 127 episodes). Moreover, students sometimes asked the expert for help (25 episodes), and this could be seen as a failure of Andes to provide adequate help.

There are some limitations to the generality of these results. First, these students had already taken physics, so they seldom learned new physics (7 episodes of 283). What they did learn (72 episodes) was the Andes notational conventions, such always using units for dimensional numbers. Second, these data come from the first two hours of Andes usage, so they are not representative of routine usage of the system.

There are several clear goals for future work. One is to provide proactive hinting to students who are refusing to ask for help even though they keep re-entering an entry incorrectly. This should increase their learning rate during this critical initial encounter with Andes. A version of Andes that provides such proactive hints has been developed and evaluated. The results, which were positive, are presented in a companion paper to this one.

A second goal would be to recode the data using the categories from a recent study of human physics tutoring (Chi, Roy, & Hausmann, 2008), which found that the measure of tutorial dialogues that most strongly correlated with learning gains was how many times the student responded to the tutor with a substantive comment (r=.605) versus an non-substantive comment (r=-.899), where a non-substantive comment was a dialogue continuer ("Uh-huh"), merely agreeing with the tutor ("OK."), repeating the tutor's remark, or making an off-task comment. In the context of Andes, after each tutor turn (e.g., coloring an entry red or giving a hint), we would code the student's verbal protocol as substantive vs. non-substantive. It seems likely that we would again see that this distinction correlates strongly with learning. This would extend Chi's hypothesis that tutoring succeeds just to the extent that it gets students thinking hard and making substantive responses. If this correlation was strong, then the distinction could be used as a timely indicator of whether a tutor turn "worked." For instance, if the tutor colors an entry red in a specific context, and the student makes a substantive comment about it, then we have indirect evidence that minimal feedback worked well in that specific context. This kind of indicator could be used to drive a supervised machine learning algorithm that would infer the contexts in which minimal feedback was effective.

A third goal would be to recode the data to answer more specific questions about students' use of hints, such as (1) when and why do students ask for hints? (2) How do they react to hints? For instance, do they self-explain bottom out-hints?

Chapter 2

PROPOSED HINTING MECHANISM

Introduction

A phenomenon that has recently captured the attention of researchers in the ITS community is “gaming the system”. Baker et al. define gaming as a behavior in which the student attempts to succeed in an educational environment by exploiting properties of the system’s help and feedback rather than attempting to learn the material (Baker et al. 2004, Baker et al. 2005, Beck 2006). In other words, students exploit the hints given by the tutor to solve problems rather than learn and work on their own. Gaming is often associated with poorer learning.

With respect to Andes (VanLehn, et al., 2005), the system used in our study, gaming consists of the following two behaviors:

Hint Abuse: Continuously seeking help, even on simplest tasks till the tutor gives the correct entry (Wood & Wood, 1999).

Hint Refusal: Repeatedly trying to guess the correct entry, and not clicking on the hint button.

Most of the work done on gaming has focused on hint abuse. Little has been done on hint refusal. According to the help-seeking model developed by Nelson-LeGall (1981), before seeking help, the learner must first realize that the given task is difficult and they may need some help. This ability to perform self-assessment and assessing the difficulty of the task at hand is a meta-cognitive skill that must be developed by students.

In a review of help-seeking literature, Aleven et al. (2003) concluded that although effective help-seeking behavior is related to better learning outcomes, learners are not using help effectively, mainly due to inadequate help use.

One plausible way to reduce help refusal and increase learning is to create systems that can detect when the student needs help and give them help without the students' asking for it (unsolicited help). This is difficult to do this because the system may not know about the nature of the student's problems in order to provide help (Anderson 1993) although expert human tutors may do better (Wood & Wood 1999) because they have access to the visual and verbal cues from the learners. Nonetheless, even a computer tutor can detect some obvious instances of hint underuse, such as making multiple errors on the same step (Wood 2001). Some systems offer both solicited and unsolicited help, but it is important to figure out the circumstances in which each kind of help would be more suited, which may be different.

Research question

The general purpose of this research is to find the effects of unsolicited hints (including meta-hints) on the learning gains of the students. The main hypothesis is that a tutor having unsolicited hints should be better than the one that does not have them, since the unsolicited hints provide a timely help to the students in situations where the students would normally waste lots of time. Also, the meta-hints teach help-seeking skills to the students, which will help the students in identifying scenarios where they need help and ask for it.

Meta-hints are unsolicited and merely suggest that the student click on a hint button. No meta-hints were present in the old version of Andes. Although the CMU Meta-tutor used a much more sophisticated meta-tutoring functionality (Roll, Aleven, McLaren, & Koedinger, 2007, 2010), this is the first time a simple meta-hinting facility such as the one described here has been used.

Groundwork

This research was conducted in two phases. In the first phase of the research (which was a separate study in itself), we looked at student reactions to a step-based tutor, Andes, a physics based intelligent tutor. Ten paid volunteers who had taken physics participated in the two hour study. These volunteers were undergraduate students from a physics course having a working knowledge of solving problems in physics. For the experiment, we used Andes 3, which is similar to Andes2 (VanLehn, et al., 2005) but has an improved user interface. During the process of solving problems in Andes, students were asked to provide information on their cognitive and affective state by talking continuously about what they were thinking and how they were approaching the problem. The computer screen and the verbal protocol given by the students were recorded. These screen recordings comprised the main data for analysis.

Segmenting the data into episodes took advantage of a feature of Andes. Andes provides immediate feedback to the students by turning the student entries green or red depending on whether the entry is correct or incorrect. This policy was used to segment the student data. An episode was defined as a series of student actions starting from a red entry and finishing in a green one. Thus, an episode is a series of steps taken by the student to correct a mistake.

Coding found that in 37% of the episodes, students failed to learn anything or to apply and reinforce knowledge they knew already. That is, the episode was a failure pedagogically. In a staggering 65 % of these episodes, students struggled only because they would not ask for help from the tutor. Instead, they would repeatedly keep editing their red entry, hoping it would turn green. These were clear cases of hint refusal.

A new hinting policy for Andes

On the basis of the protocol analysis done in the earlier study, a new hinting policy was developed for Andes. The policy is shown in Table 1. Each row is explained below.

	Event	Experiment Condition	Control Condition
1	Missing or incorrect units	Unsolicited hint to add or correct units	Some hint available upon request
2	Draw object without label and hasn't mastered labeling objects.	Unsolicited hint to label object	Same hint available upon request
3	20 sec. or 3 turns of red turns, with no help received (frustration)	Meta hint to click hint button	None
4	Entry colored red and hasn't mastered using the hint button	Meta hint to click hint button	None
5	Hint contains an "Explain more" or other link and student hasn't mastered using such links.	Unsolicited hint pointing out link	None
6	Entry colored red and video has not been watched	Meta hint to watch intro video	None

Table 2.1: New hinting policies, where "mastered X" means that student has done X at least 3 times.

Row 1: The first row of Table 1 concerns missing or mistaken units on dimensional numbers. For example, the student whose work is shown in Figure 1 was trying to assign a value of 5 meters to d, which is the magnitude of a displacement that the student defined earlier. The correct way of doing it is d=5 m, where m is the unit of displacement. But the student forgot to put units to the displacement. This was not accepted by Andes and hence the entry was colored red by Andes to indicate that it was incorrect. This is a scenario where an unsolicited hint would help. The experimental version of Andes gave an unsolicited hint "Forgot to put units on a number", which is shown on the right side of Figure 1. Now, at this point the student can either correct his mistake by adding the units to the displacement or click on the "Explain More" button below the hint which would give a more detailed hint. In the control version of Andes, this same hint would only be given if the student clicked on the Hint button to ask for a hint.

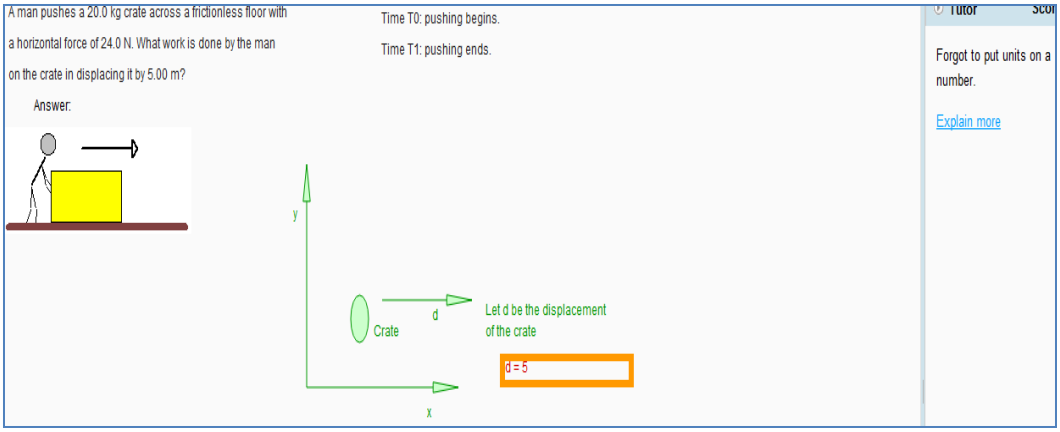

Figure 2.1: Unsolicited hint

Row 2: In the process of solving problems, whenever an object (like a vector or a body) is drawn, it has to be labeled; otherwise others (a human tutor or Andes) will not know what the student was trying to draw. Consider a scenario shown in Figure 2. Here the student was trying to draw a vector for the displacement of the crate, but he forgets to give it a label which would specify that it is the displacement. The right side of the Figure shows the hint that was given in the experimental version, "Generally objects should be labeled." In the control version the same hint was presented on clicking the hint button. In the experimental version, the hint was given to the students until they had thrice labeled an object without getting the unsolicited hint. After the student had achieved such "mastery, the unsolicited hint was no longer given and the experimental version behaved just like the control version, giving the hint only when the student asked for it.

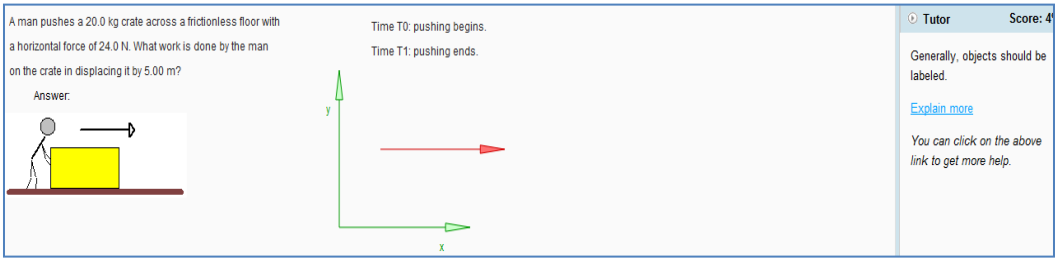

Figure 2.2: Unsolicited hint for labeling objects

Row 3: In *the* experimental version, if a student made three consecutive red entries without taking any help or if the time duration for which an entry remained red was

greater than 20 seconds, the experimental version gave them an unsolicited prompt to click on the hint button (shown in fig 2 below). This was a meta-hint since it does not actually tell students something about their mistake or the corresponding correct step, as an unsolicited, non-meta hint would do. After a student had achieved *'mastery'* in the use of the hint button (i.e., it was used three times after red entries), the experimental version no longer presented the meta-hint. The control version did not present this meta-hint.

Row 4: Similarly, when students had just had an entry turn red and they had no yet used the hint button three times, then the experimental version gave them a meta-hint suggesting that they click on the hint button.

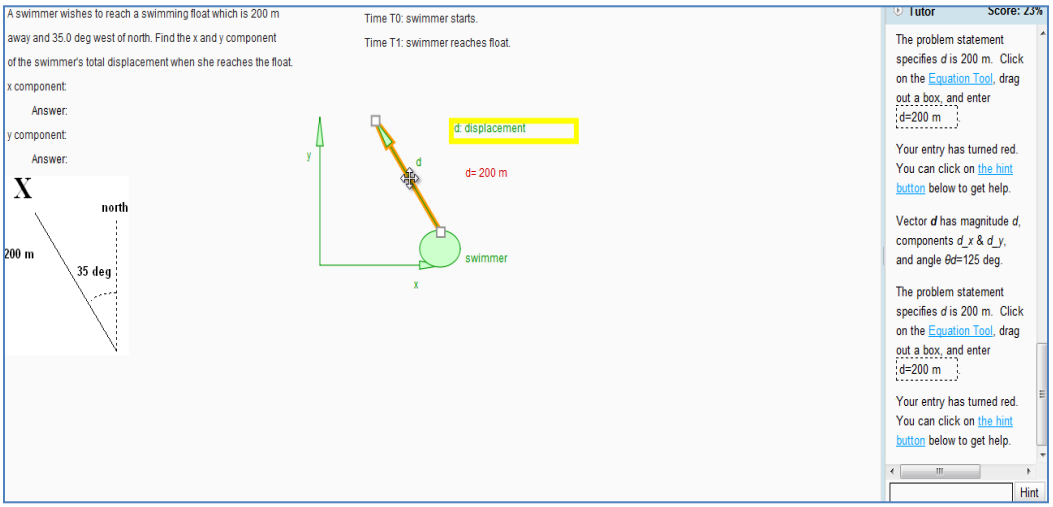

Figure 2.3: Meta Hint urging the student to take help in case he is unable to correct the mistake on his own

Row 5: Andes normally gives hints in the form of a hint sequence starting from the most general hint and going to more and more specific ones, finally giving the bottom-out hint which would give the correct step to be entered. For the first hint, the student has to click on the hint button and thereafter they have to click on the 'Explain more' button. From the first phase of the study, we observed that even students who clicked on the hint button and took the first hint struggled a lot because they did not click on the 'Explain more' button after looking at the first hint. In order to teach this skill, we created a meta-hint that

would urge the student to click on the 'Explain more' button to get more help. This hint would be given to the student till the time he achieves *'Mastery'* in its usage. This is shown in Figure 4. Similarly there are other scenarios in which there would be a link in a hint and in order to get more information on the hint the student should click on the link. For those scenarios, we made meta-hints telling them to click on the links to get more information on the hint that is given to them.

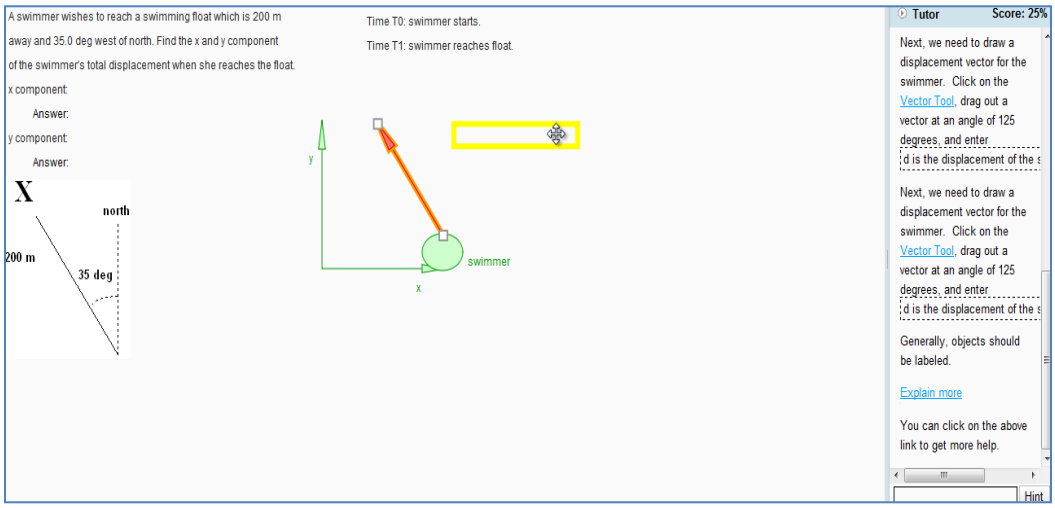

Figure 2.4: Meta Hint telling the student to click on the 'Explain more' button for more hints.

Row 6: When students log into Andes for the first time, they get a prompt telling them to view an introduction video before proceeding to solve problems, since the video would essentially tell them all the conventions to be followed by Andes. In some cases, students simply dismissed the prompt and started with the problems. Due to this they were not clear on how to solve problems and hence ended up making mistakes. In order to prevent this, whenever students using the experimental version made some mistakes and they had not seen the training video, Andes gave them a meta-hint telling them to watch the video before attempting to solve the problems.

The old version of Andes, which was used in the study described earlier, was identical to the control version used here, with one exception. The old version of Andes had an

unsolicited hint for missing or incorrect units, which correspond to the experimental version here rather than the control version (see description of Row 1 of Table 1). All the other unsolicited hints in the old version of Andes concerned user interface conventions, such as only entering syntactically correct equations into the equation tool, whereas the units convention is more of a physics notational convention. By making the units hint unsolicited in the experimental version, the control version now has unsolicited hints only for violation of user interface conventions, which is typical of most software. This made the distinction between versions cleaner.

Methods

Our experiment simply tested the comparative effectiveness of the control version of Andes with the experimental version that gave more unsolicited hints and gave meta-hints.

Design

This experiment was designed as a two-group training study. There are three common measures in such study: (1) the number of training problems solved, (2) the amount of time spent solving training problems and (3) the gain in scores between a pre-test and a post-test. A training experiment can equate one of these three measures across conditions, but not two. A typical design equates number of problems solved (measure 1) and uses training time (measure 2) and gain scores (measure 3) as dependent variables. We used a less typical design, which equates the amount of training time (measure 2) and treats number of problems solved (measure 1) as the main dependent variable. This design requires a second dependent measure, such as gain scores (measure 3), which determine how students' competence changes. Without such a measure, we would not detect when a group is completing more problems because they are sacrificing their learning. We used this design because the instruction focused more on teaching the conventions of Andes, so it was not clear what pre-testing and post-

testing would work besides simply using Andes. Thus, we used log data measures such as errors and latencies as embedded assessments of competence. Discussion of the individual measures is delayed until the results section.

Materials

Two versions of Andes3 were developed. The experimental version had unsolicited hints on physics conventions and meta-hints on using the system help appropriately. The control version used no meta-hints and used unsolicited hints only on user interface conventions. The most serious bugs uncovered during the early study were removed from both versions of Andes3.

Participants

Participants were recruited from an introductory physics class. These 35 paid volunteers had all covered Work and Energy in class, and had taken the relevant midterm exam on it. Thus, they should all have been at least somewhat familiar with the concepts and skills taught during the experiment. One student declined to participate after starting the experiment, so that student's data was destroyed. Thus, 34 students completed the experiment.

Procedure

Students were randomly divided into control (N = 17) and experimental (N=17) groups. The experimental group used the experimental version of Andes and the control group used the control version of Andes. Students were run individually in a lab setting for a maximum of 2 hours. After a brief introduction to the experiment and to Andes, students in both groups solved problems in Andes. The first problem they solved was a "walk through" where Andes told them each step to do. This problem was not counted in any of the analyses discussed below.

Students wore a headset microphone and were instructed to talk aloud as they worked, explaining their reasoning. If they fell silent, the experimenter, who was within earshot but not next to the students, would remind them to keep talking. Their voices were recorded along with the state of their screen using screen recording software. The resulting video files have not yet been analyzed. The results reported here were derived from analysis of the log data output by Andes.

Results

First dependent variable: Problems solved per unit time

As mentioned earlier, the design of the experiment equates time spent on solving problems (measure 1 in the list above) and use number of problems solved (measure 2) as the first main dependent variable. Unfortunately, the actual time spent solving problems varied due to logistics, difficulties with the computer systems, difficulties with the screen recording system, etc. Although most participants worked for the full 2 hours, the participant who worked for the shortest time worked for only 1 hour and 34 minutes. Thus, in order to determine which group completed the most problems on average in the same period of time, we only counted the number of problems completed by the 1 hour and 34 minute mark. This is the maximal duration for which all students were working, so this time period is referred to as the "common work time."

For the experimental group, the mean number of problems completed during the common work time (1 hour, 34 minutes) was 7.47 problems, whereas the control group completed only 5.98 problems on average. This difference was reliable ($p=0.034$), and the effect size was large ($d = 0.727$). Figure 4 shows these results graphically, with error bars indicating the standard error of the mean. Figure 5 shows the number of students completing each problem. Problems appear on the x-axis in the order that they were given to students. All students completed the initial walk-through (vec1ay) and the first 4

problems. Of the 17 experimental students, one student completed only 5 problems; whereas 4 of the 17 control completed only 5 problems.

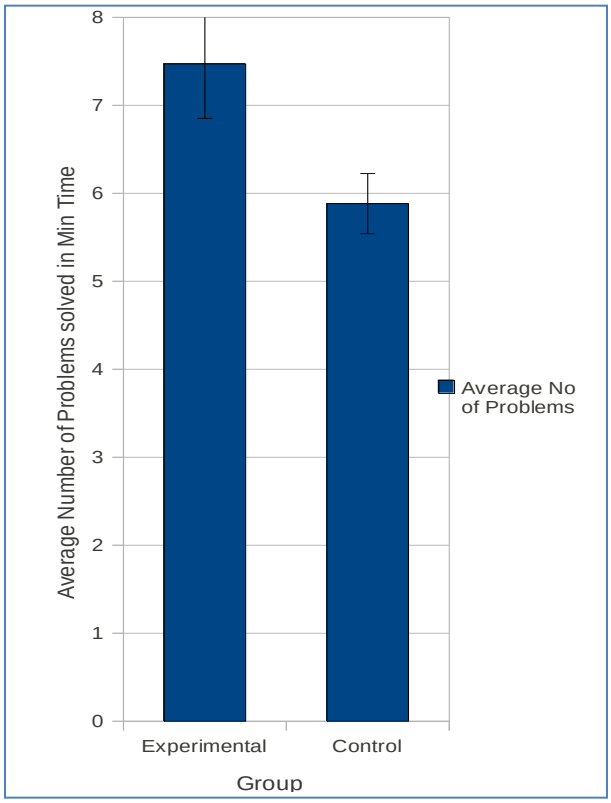

Figure 2.5: Average number of problems solved in the common work time

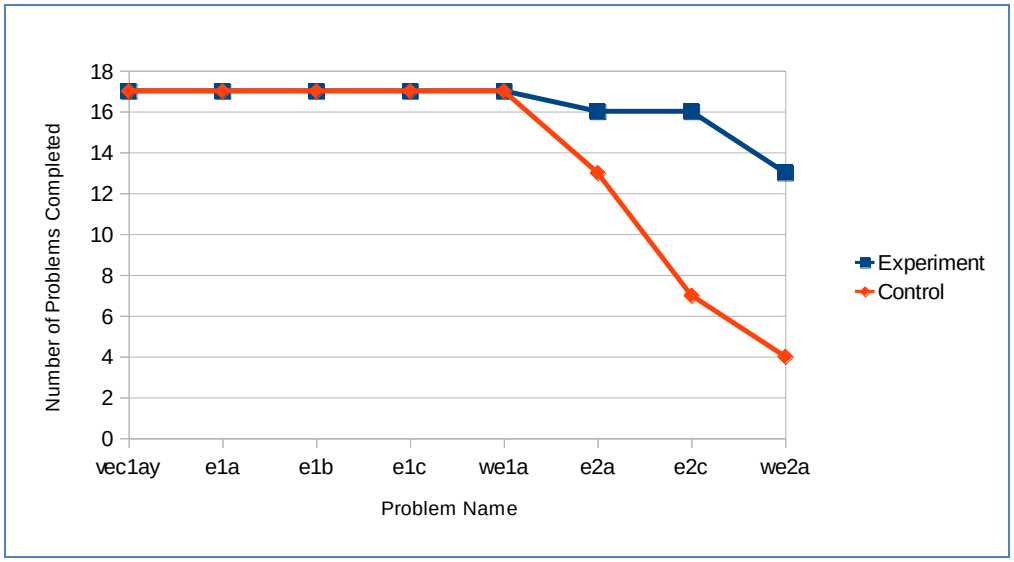

Figure 2.6: Number of problems completed by students of both conditions

If the experimental students always learned more than the controls, then their advantage over the controls should increase monotonically as the experiment progresses. In order to see if this trend occurred, Figure 6 shows the average time to complete each problem, and Figure 7 shows the cumulative time. When interpreting these graphs, as well as others that have problems along the horizontal axis, it is important to know that only the first 4 problems were completed by all 34 students in the experiment. Figure 5 shows the number of students who completed each problem. Although a few students completed more than 8 problems, only 8 problems are shown on the horizontal axes for these Figures and subsequent ones.

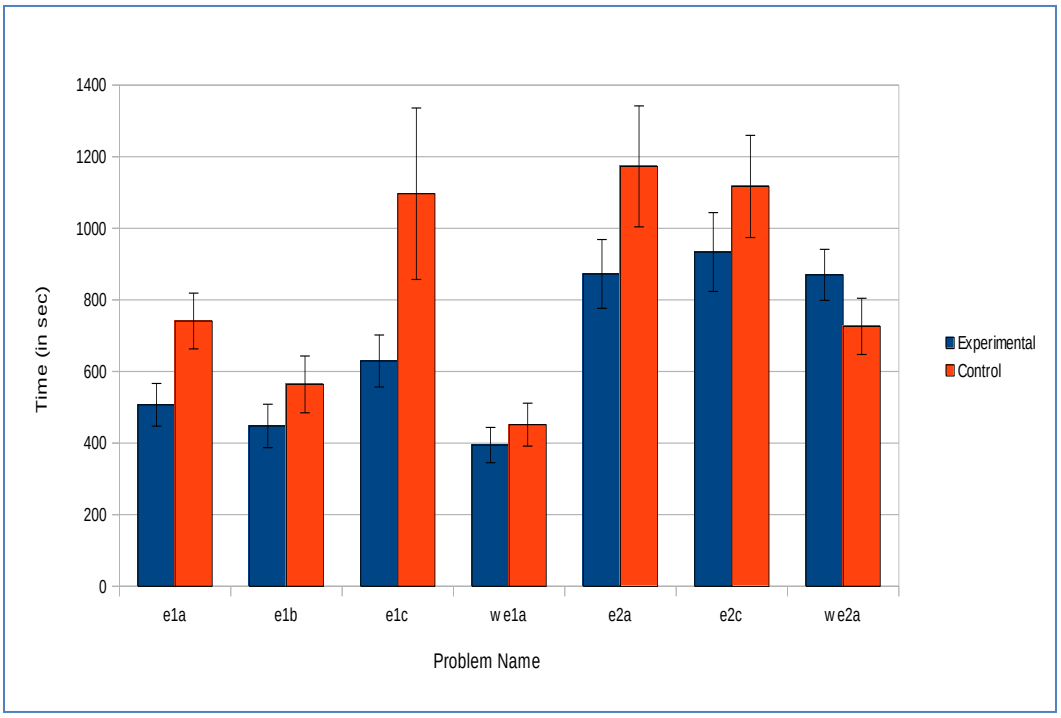

Figure 2.7: Average time taken by students to solve problems.

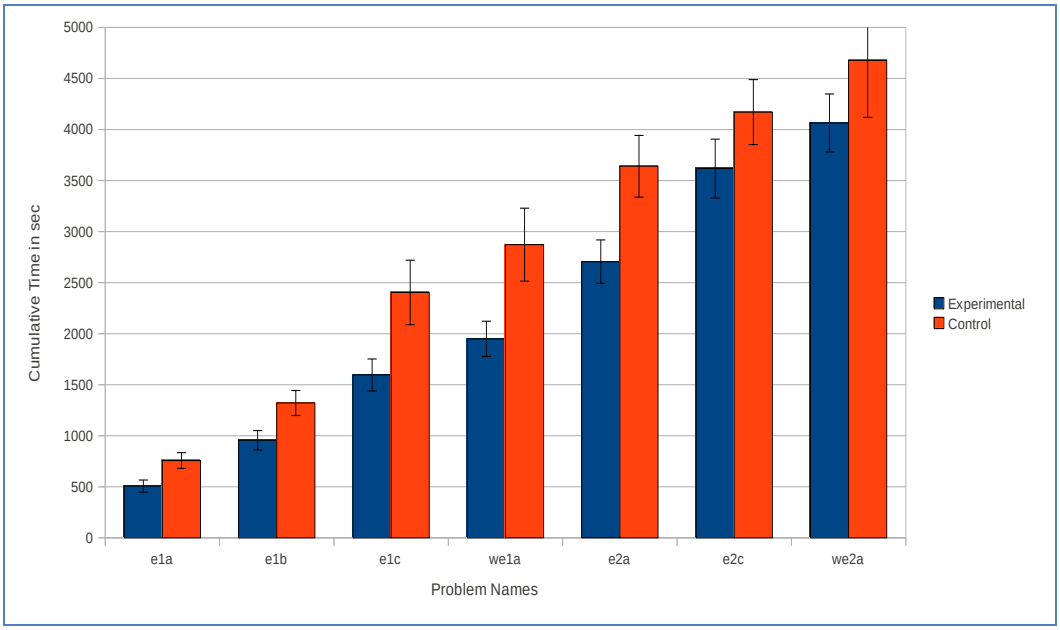

Figure 2.8: Average of the cumulative time taken by students to solve problems.

Returning now to the interpretation of Figure 6 and 7, it appears that the predicted increasing advantage does occur during the first 3 problems, but the trend's existence is obscured by the third problem (e1c) which seems to be considerably harder than the others. In fact, the first, second and fourth problems (e1a, e2b and we1a) use similar concepts and notations, while the second, fifth and sixth problems (e1c, e2a and e2c) also use a similar set of concepts and notations, but not the same set as used by e1a, e2b and we1a. This could explain why the time for solving e1c is higher than it should be given the expected trends.

After the fifth or sixth problem, the expected trend disappears in part because the students completing those problems are no longer a random sample of the population. For instance, on the last problem shown, we2a, the average time was taken over 4 control students who were the only ones who hadn't gone home yet; they were the ones who worked fast enough that they completed we2a before the two hours elapsed. Thus, this 4-student group is probably more inherently competent than the 10 student group of experimental students.

In short, Figures 6 and 7 show some signs of the expected trend, which is that experimental groups time should start being much lower than the control group, but as the control group finally learns the requisite knowledge, their time per problem comes closer to matching the experimental group's time per problem.

Second dependent variable: Competence

The preceding section showed that the experimental group, who received more unsolicited help than the control group, also solved problems faster. However, this increase in speed could be occurring if the unsolicited help simply reduced the time to repair errors and did not result in better learning. Thus, we need some measure of competence in order to test that the unsolicited help is not doing more harm than good.

Given that the instructional objectives mostly concerned the use of specific physics notations and the error rate over time, if students were learning the conventions early (experimental condition), then their errors later in the experiment should decrease. Moreover, their overall errors should be lower by the end of the experiment because their error rate drops faster than the control group's error rate drops.

However, we cannot just count errors. If a student is uncertain about what to do, then some students prefer to ask for help rather than risk making an error. Other students have the opposite preference. Thus, the student controls whether their weakness shows up as errors or help requests. Thus, a better measure than errors alone is the "assistance score," which is defined to be the sum of the solicited hints and the number of errors divided by the total number of solution steps. This measure has been used in earlier studies with Andes (Hausmann, Nokes, VanLehn, & van de Sande, 2009; Hausmann, van de Sande, & VanLehn, 2008; Hausmann & VanLehn, 2007a, 2007b) and other steps-based tutoring system that have allowed students to ask for help. Note that if Andes chose to give an unsolicited hint or meta-hint, it did so only *after* the student

made an error. Thus, the manipulation did not directly prevent errors, but only reduced errors by teaching students to avoid making them.

For the steps done during the common work time, the mean assistance score for the experiment as was 0.173, whereas the mean assistance score for the controls was 0.222. This difference is reliable ($p = 0.045$) and the effect size was large ($d = 0.682$). Figure 8 shows these results graphically, with error bars indicating the standard error of the mean. Figure 9 shows the mean assistance score per problem. The experimental students have better (lower) scores on almost every problem. Although we would expect their advantage to gradually decrease as the control group finally learns what the experimental group learned earlier, there is no sign of such a trend.

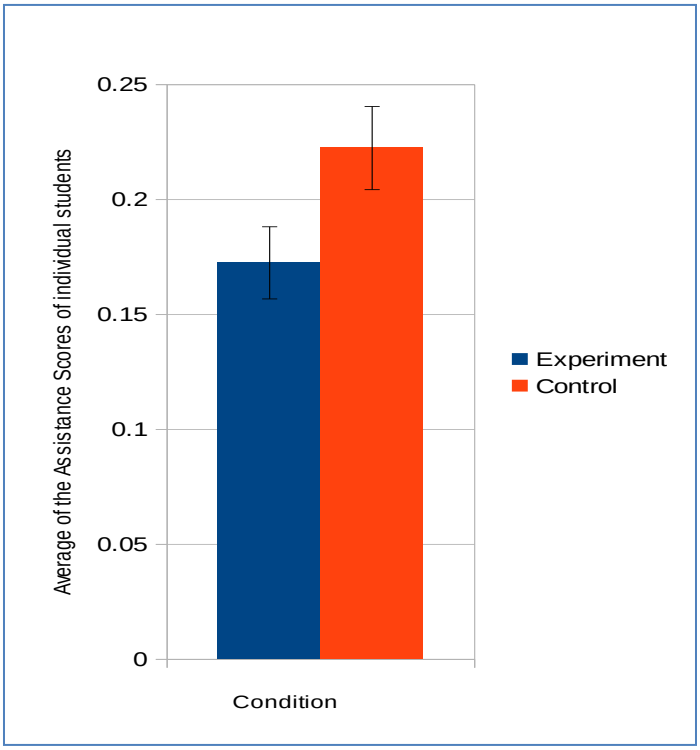

Figure 2.9: Average of the Assistance Scores

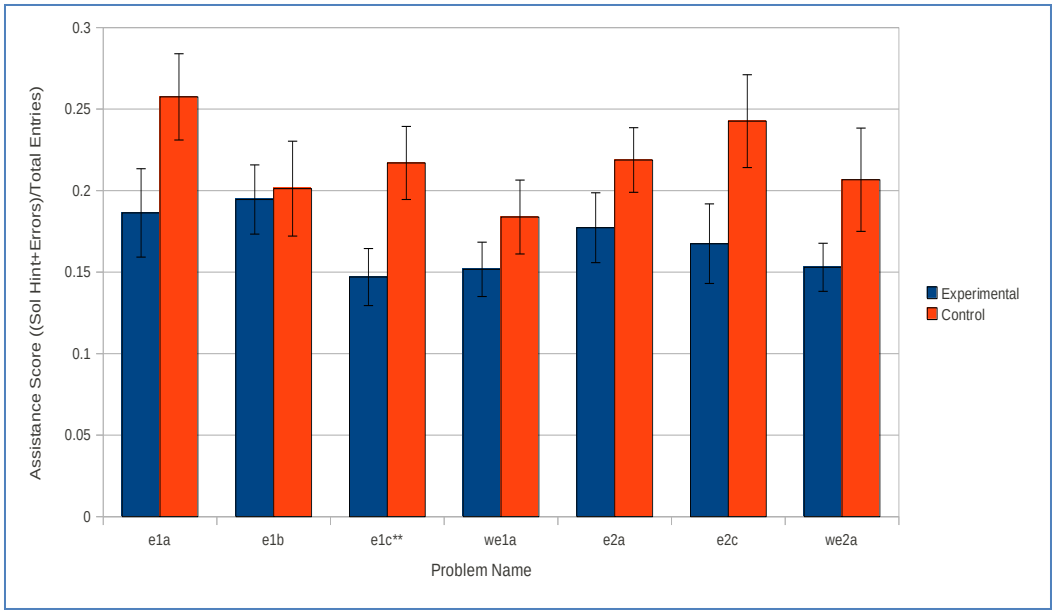

Figure 2.10: Assistance Scores

In general, when students learn, they not only make fewer mistakes but they also recover from the mistakes more rapidly. Thus, we defined the error correction time as the length of an error-correction episode. The episode starts during when a student's entry turns red and ends when the student re-enters the entry correctly. For each student, we calculated the average time for that student's error-correction episodes. Then taking the mean over the students in each group, we found that the mean error correction time for the experimental group was 73.21 seconds versus 116.76 seconds for the control group. This was statistically reliable ($p = 0.0004$) and a large effect size ($d = 1.13$). This is represented graphically in Fig 10.

As with the other measure of competence, errors, we would expect the experimental group to start well ahead of the control group, then to see the control group catch up. This trend is partly visible in Figure 11, which compares the two group's average error correction time per problem. The experimental group recovered from errors much faster initially, and the control group started to catch up. As discussed earlier, the trend is more visible with problems e1a, e1b and we1a, which all tap the same knowledge. Problem e1c taps different knowledge, so the error correction time jumps upward on it.

Moreover, after the fourth problem (we1a), the trends become difficult to interpret as the samples become non-random as discussed earlier (see Figure 5).

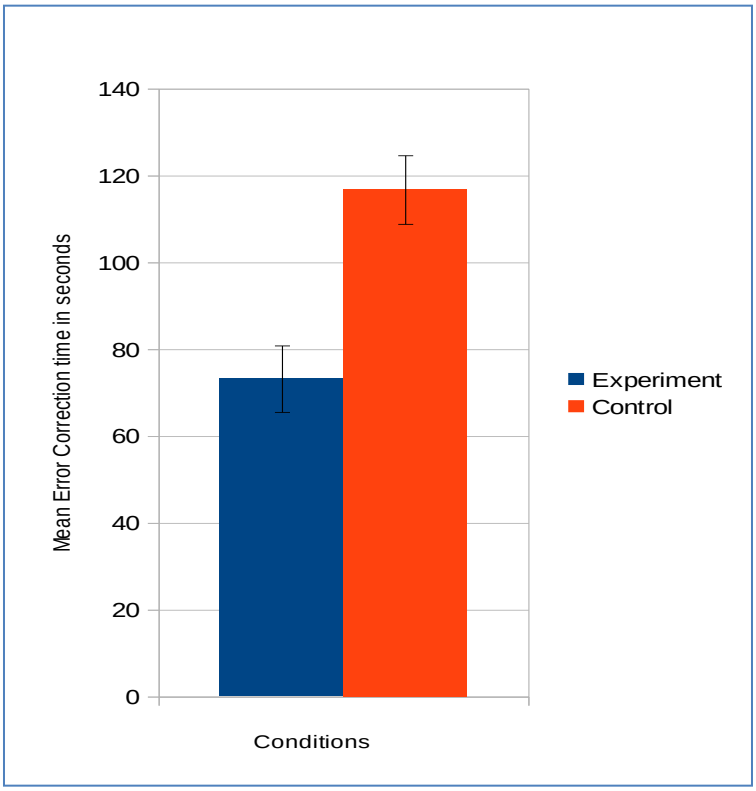

Figure 2.11: Mean Error Correction time

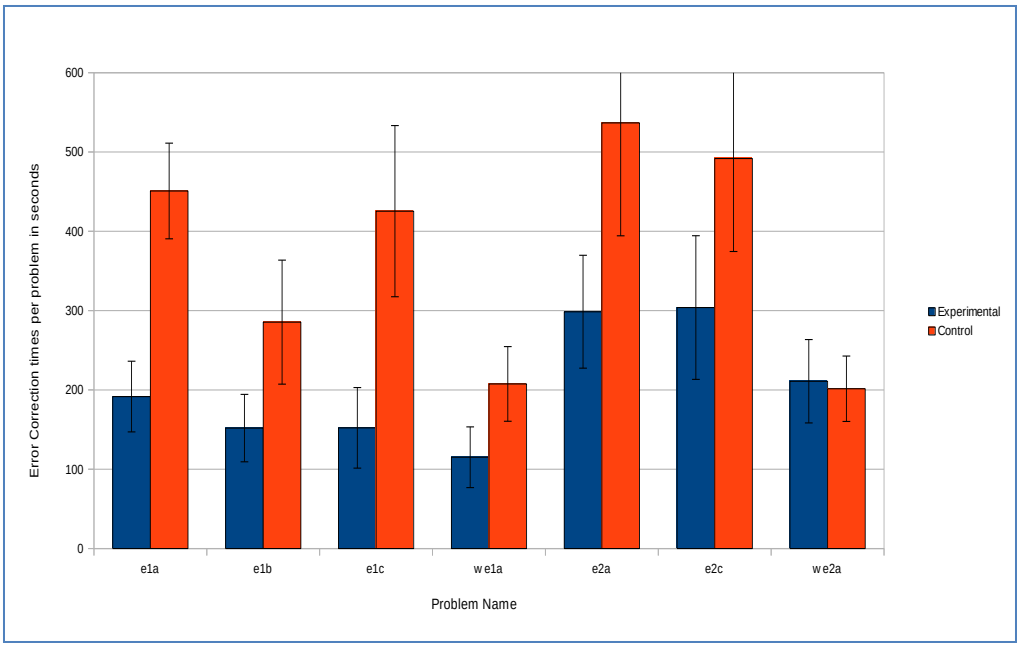

Figure 2.12: Error correction times per problem.

Discussion

All the results suggest that the experimental group performed better than the control group. The experimental students completed more problems (7.47 vs. 5.98) in the same amount of time (1 hour 34 minutes). Their average time per problem was lower (Fig 6). Their assistance scores were lower (0.173 vs. 0.222), where the assistance score is the sum of the number of errors and the number of help requests divided by the number of steps taken. And finally, the experimental students took less time to correct their errors than the control students (73.21 seconds vs. 116.76 seconds). These results are consistent with the hypothesis that giving unsolicited hints and meta-hints after the student has made a mistake is clearly a worthwhile practice for Andes.

However, there are some limitations to this study that should be mentioned. First, it only covered the first 2 hours of user experience. After 2 hours, the unsolicited hints and meta-hints may be less effective, and may even become irritating. Second, there were trends in the data suggesting that the control group slowly learned what the experimental group rapidly learned. If so, then the advantage of unsolicited hints and meta-hints may disappear over a few hours. After 10 hours of usage, it would be hard to tell from students' behavior which condition they had been in.

Most importantly, the unsolicited hints on physics conventions, such as appending units to numbers, and the meta-hints focused on using the help system appropriately. These are not conceptually difficult, so it would not take a student long to comprehend the hint and get back to what they were doing before they were interrupted. It is not clear how students would react to more conceptually difficult hints. In particular, it is not clear whether applying this same policy would be successful for hints about the more difficult aspects of physics that dominate students floundering time after their initial two hours with Andes.

Although these limitations caution us about over generalizing the results, this is still a positive and highly welcome result. For many years, we have been mystified about the large number of anonymous users who try Andes on the web for a few problems then never come back. We have even called it “the user interface hump” and talked extensively about how to get users over it. Now we may know why, and seem to have a solution. It is likely that other intelligent tutoring systems have similar problems, and for similar reasons. Instructors and computers often like students to communicate in a more precise fashion than students are wont to do naturally. Even when students have been told and shown how to communicate precisely, they seem to require unsolicited hints to get them to actually change their habits

REFERENCES

Aleven, V., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In G. Gauthier, C. Frasson & K. VanLehn (Eds.), *Intelligent Tutoring Systems: 5th International Conference, ITS 2000* (pp. 292-303). Berlin: Springer.

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

Baker, R. S., Corbett, A., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A. Z., et al. (2006). Adapting to when students game an intelligent tutoring system *Intelligent Tutoring Systems* (pp. 392-401). Berlin: Springer.

Baker, R. S. J. d., Corbett, A., Koedinger, K. R., & Wagner, A. Z. (2004). *Off-task behavior in the cognitive tutor classroom: When students "Game the System".* Paper presented at the Proceedings of ACM CHI 2004: Computer Human Interaction.

Baker, R. S. J. d., de Carvalho, A. M. J. A., Raspat, J., Corbett, A., & Koedinger, K. R. (2009). Educational software features that encourage or discourage "gaming the system" *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (pp. 475-482). Amsterdam: IOS Press.

Baker, R. S. J. d., Walonski, J., Heffernan, N. T., Roll, I., Corbett, A., & Koedinger, K. R. (2008). Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of interactive Learning Research, 19*(2), 185-224.

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

Cade, W. L., Copeland, J. L., Person, N., & D'Mello, S. K. (2008). Dialogue modes in expert tutoring. In B. P. Woolf, E. Aimeur, R. Nkambou & S. P. Lajoie (Eds.), *Intelligent Tutoring Systems: 9th international Conference, ITS2008* (pp. 470-479). Berlin, Germany: Springer.

Chi, M. T. H. (2009). Active-Constructive-Interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science, 1*, 73-105.

Chi, M. T. H., Roy, M., & Hausmann, R. G. M. (2008). Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *Cognitive Science, 32*(2), 301-342.

Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science, 25*, 471-533.

Cho, B.-I., Michael, J. A., Rovick, A. A., & Evens, M. W. (2000). An analysis of multiple tutoring protocols. In G. Gauthier, C. Frasson & K. VanLehn (Eds.), *Intelligent Tutoring Systems: 5th International Conference* (pp. 212-221). Berlin: Springer.

Corbett, A. (2001). Cognitive computer tutors: Solving the two-sigma problem *User Modeling: Proceedings of the Eighth International Conference* (pp. 137-147).

Core, M. G., Moore, J. D., & Zinn, C. (2003). The role of initiative in tutorial dialogue *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (Vol. Morristown, NJ, pp. 67-74): Association of Computational Linguistics.

Evens, M., & Michael, J. (2006). *One-on-one Tutoring By Humans and Machines*. Mahwah, NJ: Erlbaum.

Fox, B. A. (1991). Cognitive and interactional aspects of correction in tutoring. In P. Goodyear (Ed.), *Teaching Knowledge and Intelligent Tutoring* (pp. 149-172).

Fox, B. A. (1993). *The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Frederiksen, N., Donin, J., & Roy, M. (2000). *Human tutoring as a model for computer tutors: Studying human tutoring from a cognitive perspective.* Paper presented at the Modeling human teaching tactics and strategies: Workshop W1 of ITS2000, Montreal, Canada.

Graesser, A. C., Person, N., & Magliano, J. (1995). Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology, 9*, 359-387.

Graesser, A. C., VanLehn, K., Rose, C. P., Jordan, P., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine, 22*(4), 39-41.

Hausmann, R. G. M., Nokes, T. J., VanLehn, K., & van de Sande, B. (2009). Collaborative dialogue while studying worked-out examples. In V. Dimitrova, R. Mizoguchi, B. Du Boulay & A. C. Graesser (Eds.), *Artificial Intelligence in Education*. Amsterdam: IOS Press.

Hausmann, R. G. M., van de Sande, B., & VanLehn, K. (2008). Are self-explaining and coached problem solving more effective when done by pairs of students than alone? In B. C. Love, K. McRae & V. M. Sloutsky (Eds.), *Proceedings of the 30th meeting of the Cognitive Science Society* (pp. 2369-2374). Mahway, NJ: Erlbaum.

Hausmann, R. G. M., & VanLehn, K. (2007a). Explaining self-explaining: A contrast between content and generation. In R. Luckin, K. R. Koedinger & J. Greer (Eds.), *Artificial Intelligence in Education* (pp. 417-424). Berlin: Springer.

Hausmann, R. G. M., & VanLehn, K. (2007b). Self-explaining in the classroom: Learning curve evidence. In D. S. McNamara & G. Trafton (Eds.), *Proceedings of the 29th Annual Conference of the Cognitive Science Society* (pp. 417-424). Mahwah, NJ: Erlbaum.

Hastings, P., Arnott-Hill, E., & Allbritton, D. (2010). Squeezing out gaming behavior in a dialog-based ITS. In V. Aleven, H. Kay & J. Mostow (Eds.), *Intelligent Tutoring Systems 2010* (pp. 204-213). Berlin: Springer-Verlag.

Hume, G., Michael, J., Rovick, A., & Evens, M. (1996). Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences, 5*(1), 23-49.

Katz, S., Allbritton, D., & Connelly, J. (2003). Going beyond the problem given: How human tutors use post-solution discussions to support transfer. *International Journal of Artificial Intelligence in Education, 13*, 79-116.

McArthur, D., Stasz, C., & Zmuidzinas, M. (1990). Tutoring techniques in algebra. *Cognition and Instruction, 7*(3), 197-244.

Merrill, D. C., Reiser, B. J., Merrill, S. K., & Landes, S. (1995). Tutoring: Guided learning by doing. *Cognition and Instruction, 13*(3), 315-372.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences, 2*(3), 277-306.

Muldner, K., Burleson, W., van de Sande, B., & VanLehn, K. (2010). An analysis of gaming behaviors in an intelligent tutoring system. In V. Aleven, J. Kay & J. Mostow (Eds.), *Intelligent Tutoring Systems: 10th International Conference, ITS 2010* (pp. 184-193). Heidelberg, Germany: Springer.

Muldner, K., Burleson, W., Van de Sande, B., & VanLehn, K. (2011). An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impacts. *User Modeling and User-Adapted Interaction, 21*(1-2), 99-135.

Murray, R. C., & VanLehn, K. (2005). Effects of dissuading unnecessary help requests while providing proactive help. In G. I. McCalla, C.-K. Looi, B. Bredeweg & J. A. Breuker (Eds.), *Proceedings of Artificial Intelligence in Education* (pp. 887-889). Amsterdam: IOS Press.

Ohlsson, S., Di Eugenio, B., Chow, B., Fossati, D., Lu, X., & Kershaw, T. C. (2007). Beyond the code-and-count analysis of tutoring dialogues *Artificial Intelligence in Education*. Amsterdam: IOS.

Ringenberg, M., & VanLehn, K. (2006). Scaffolding problem solving with annotated, worked-out examples to promote deep learning. In K. Ashley & M. Ikeda (Eds.), *Intelligent Tutoring Systems: 8th International Conference, ITS2006.* (pp. 114-123). Amsterdam: IOS Press.

Roll, I., Aleven, V., McLaren, B., & Koedinger, K. R. (2007). Can help seeking be tutored? Searching for the secret sauce of metacognitive tutoring *Proceedings of the International Conference on Artificial Intelligence in Education* (pp. 203-210). Amsterdam: IOS Press.

Roll, I., Aleven, V., McLaren, B., & Koedinger, K. R. (2010). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*.

Shih, B., Koedinger, K. R., & Scheines, R. (2008). *A response time model for bottom-out hints as worked examples*. Paper presented at the Proceedings of the 1st International Conference on Educational Data Mining.

VanLehn, K. (1999). Rule learning events in the acquisition of a complex skill: An evaluation of Cascade. *Journal of the Learning Sciences, 8*(2), 179-221.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence and Education, 16*, 227-265.

VanLehn, K. (in press). The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist*.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science, 31*(1), 3-62.

VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., Taylor, L., et al. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education, 15*(3), 147-204.

VanLehn, K., Siler, S., Murray, C., Yamauchi, T., & Baggett, W. B. (2003). Human tutoring: Why do only some events cause learning? *Cognition and Instruction, 21*(3), 209-249.

Walonoski, Jason; Heffernan, Neil (2006). Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems: Lecture Notes in Computer Science, 4053, 382-391

Woolf, B. P. (2009). *Building Intelligent Interactive Tutors*. Burlington, MA: Morgan Kaufman.

APPENDIX A

LIST OF PROBLEMS USED IN THE EXPERIMENT

Following are the list of problems used in the experiment (students were made to solve as many problems as they can in the same order): Vec1ay, e1a , e1b , e1c , we1a , e2a , e2c , we2a , we3a, we4a, e2b, e5a, e5b, weq2, weq4, weq5, e3a, e6a, e8a, e9a.

APPENDIX B

IRB APPROVAL

CITI Collaborative Institutional Training Initiative (CITI)

Responsible Conduct of Research Curriculum Completion Report
Printed on 5/18/2010

Learner: Rajagopalan Ranganathan (username: rrangan5)
Institution: Arizona State University
Contact Information Dr Kurt VanLehn, Professor CS Dept, ASU
Department: Computer Science
Email: rrangan5@asu.edu

Engineers Responsible Conduct of Research:

Stage 1. RCR Passed on 05/18/10 (Ref # 4426336)

Required Modules	Date Completed	Score
Introduction to RCR for Engineers	05/17/10	no quiz
Completing the RCR for Engineers Course	05/17/10	no quiz
Research Misconduct	05/17/10	5/5 (100%)
Whistle Blowing and the Obligation to Protect the Public	05/17/10	6/7 (86%)
Responsible Authorship in Engineering	05/17/10	4/4 (100%)
Ethical Issues in Peer Review and Publication in Engineering Research	05/18/10	4/4 (100%)
Conflicts of Interest in Engineering Research	05/18/10	4/5 (80%)
Environmental Ethics	05/18/10	3/4 (75%)
The Ethics of Mentoring	05/18/10	7/7 (100%)
Human Subjects Research in Engineering Fields.	05/18/10	5/5 (100%)
The Use of Live Animals in Research	05/18/10	7/8 (88%)
Ethical Issues in the Management of Data in Engineering Research	05/18/10	13/14 (93%)
Collaborative Research in Engineering Fields	05/18/10	5/7 (71%)
The CITI RCR Course Completion Page.	05/18/10	no quiz
Arizona State University	05/18/10	no quiz

For this Completion Report to be valid, the learner listed above must be affiliated with a CITI participating institution. Falsified information and unauthorized use of the CITI course site is unethical, and may be considered scientific misconduct by your institution.

Paul Braunschweiger Ph.D.
Professor, University of Miami
Director Office of Research Education
CITI Course Coordinator