

EINO The Answer

James Hollister, University of Central Florida, USA
Sam Richie, University of Central Florida, USA
Arthur Weeks, University of Central Florida, USA

ABSTRACT

This study investigated the various methods involved in creating an intelligent tutor for the University of Central Florida Web Applets (UCF Web Applets), an online environment where student can perform and/or practice experiments. After conducting research into various methods, two major models emerged. These models include: 1) solving the problem for the student 2) helping the student when they become stymied and unable to solve the problem. A storyboard was created to show the interactions between the student and system along with a list of features that were desired to be included in the tutoring system. From the storyboard and list of features, an architecture was created to handle all of the interactions and features. After the initial architecture was designed, the development of the actual system was started. The architecture underwent a several iterations to conclude with a working system, EINO. EINO is an intelligent tutoring system integrated into the UCF Web Applets. The final architecture of EINO incorporated a case-based reasoning system to perform pattern recognition on the student's input into the UCF Web Applets. The interface that the student interacts with was created using Flash™. EINO was implemented in three of the experiments from the UCF Web Applets. A series of tests were performed on the EINO tutoring system to determine that the system could actually perform each and every one of the features listed initially. The final test was a simulation of how the EINO would perform in "real life." Test subjects with the same educational level as the target group were chosen to spend an unlimited time using each of the three experiments. A single experiment is designed to reinforce a topic currently being covered by the book. Each of the test subjects filled out a survey on every lab to determine if the EINO system produced a helpful output.

Keywords: EINO, Intelligent Tutoring System, ITS, Cased Based Reasoning, CBR

INTRODUCTION

With education budgets shrinking every year and class sizes increasing, teachers can use help to provide a one-on-one experience to every student[1]. Some students require more attention than others. This can leave some student perplexed when it comes to understanding the subject matter. Larger class sizes make it harder on the teacher to perform interactive experiments during class because many of the students may have problems with the equipment or have questions about the assignment. These questions and problems increase the amount of time required to complete the experiment for all the students. The ideal solution to these problems would be to have one teacher for every student in the class. This is the concept behind EINO (pronounced I-know). EINO is an intelligent tutoring system (ITS) intergraded into set of web applets containing each experiment.

The first section of this paper contains information about some other ITSs and what can be learned from these systems. The second section of this paper contains brief information about the web applets or experiments. An overview of the functionality of EINO is covered in the third section. The fourth section contains the testing and results of this system followed by the conclusion and future work.

OTHER TUTORING SYSTEMS

There have been many intelligent tutoring systems (ITSs) over the years. The ANDES system[2] had the most influence in the design of the EINO system. This tutoring system has been continuously revised since its development and individuals have used it for a number of years with great success. The first version of the ANDES system used a Bayesian network to create a model of the student[3]. The information within the student model consists of the student's general knowledge, the student's specific knowledge, the student's problem-solving approach to the current problem, what help the student has received in the past, and what questions the student has already completed of the assignment. The "solution graph" and the assessor are two of the most important components in the student model. Without these two mechanisms, the student model and the hint system would not function properly.

In the ANDES system, a key factor in providing hints for the students is the "solution graph." A "solution graph", which is generated by the problem solver, contains all of the rules, facts, strategies, and goals needed to solve the presented problem. The problem solver is a rule-based system that when it is finished running, results in the creation of the "solution graph." The problem solver is only run once with the given problem, as the "solution graph" for a given problem will not change.

The assessor is recreated from the "solution graph" every time a student selects a problem. This implies the basic structure of the assessor, which is a Bayesian network, is the same as the "solution graph." The difference is that the assessor has the probabilities that the student knows a given rule, fact, strategy, or goal in the current problem. There are five kinds of nodes used in the assessor. The five nodes are: Rule, Fact, Goal, Rule Application, and Strategy. The Rule nodes are used to determine if the student has the ability to apply the corresponding rule. The Fact nodes are used to determine the probability that the student has the knowledge of the corresponding fact. The Goal nodes are used to determine the probability that the student has been working toward the corresponding goal. The Rule Application nodes are used to determine the probability that the student can apply current knowledge and form a new fact, rule, or goal. The Strategy nodes are used to determine the probability that the student could choose a different strategy to solve the given problem. Once a step has been detected by the action interpreter, the corresponding node is set to "true" as the system is now sure that the student has the knowledge necessary to solve the problem. When the user requests "help" from the ANDES system, the probabilities in the assessor are used to give the user what the system deems the most helpful to solve the problem.

There are two types of suggestions that could possibly be given to a student[3]. The first type of suggestion would include conceptual help, which involves helping the student with the theory behind why to perform the particular operation at that specific time. The second type of suggestion has to do with procedural help. This is the most common type of help needed. Procedural help is the answer to the question of "what do I do next?" Since the ANDES system allows the student to solve the presented problem in any manner. ANDES needs to determine which strategy the student is using to solve the problem. This is to make sure that the hint given to the student is helpful. The way that ANDES achieves this task is by performing a depth-first search of the "solution graph." This will provide all possible ways to solve the presented problem. The next task done by the ANDES system is to determine which solution path the student appears to be attempting. Once done, the ANDES system can present a procedural hint that is helpful to the student. This is how help is given in a proven system.

EINO is a Case-Based Reasoning (CBR) system. An example of a case-based reasoning tutor was presented by Regan and Slator[4]. This team of professors has created a variety of virtual environments for educational purposes. One of these virtual environments, the DollarBay, was created to assist students in gaining some experience in retailing merchandise. A decision was made by the design team to add a tutor into the system to help students who were having trouble understanding the concepts. CBR was soon added to provide assistance to the student. A new tutor is created each time a student enters the virtual environment. In any CBR system, the strength of the system is derived from the design of the case library and the ability to find similar cases.

The cases of the tutoring system for the DollarBay virtual environment are divided up into one of three different groups of cases. The first group of cases includes the historical cases. This group contains all cases of actions that a good player has made when the simulation is finished. The second group is the case library; this group

has a copy of every distinctive case. The third group of cases is the prototypical cases. This case group is made up of possible players designed by the creators of the system. These possible players will be matched up to how the player is currently interacting with the system, over-cautious, over-spender, etc. The tutor uses the last group of cases and all the active cases to determine how to interact with the student. The tutor does not appear to the student until after the student has been in the system for twelve hours[4]. This allows the student to become familiar with the system. The tutor then presents different messages based upon how the student is doing in the virtual environment; an example of a message is “Try to keep more cash on hand” if the student has a low cash reserves. After the system was evaluated, it was determined that the tutor functions properly and provides helpful information to a struggling student and thus is able to help the student perform better.

When designing a system to handle all the interactions between the tutor and the student, it is helpful to use a technique that is able to represent the interactions. Storyboarding is a design tool that lets one imagine how the end product could appear and is used mainly for designing movies and television programs. This same concept has been adapted for the e-learning environment, but the current storyboarding concepts are just replacing software design documents. In order for a student to learn, an interaction must occur with a teacher over some form of medium. This medium could be a simple conversation, a book, a video, or any way a professor may choose to utilize a learning environment. Current storyboarding concepts are inadequate to handle the use of all of these mediums[5]. Consequently, a re-creation of storyboarding must take place. The first item needed in the re-creation is a node. Nodes can represent an entire episode or just a scene in the episode[5]. To make this simpler, an episode will require a subgraph explaining the episode scene by scene. A scene could include a video that is playing, a picture or pictures that are being displayed, or a document opening up for the user. By having nodes represent scenes and episode, this allows storyboarding to be scalable. In most videos, whether it is on television or in the movies, there are multiple scenes and between each scene a transition is needed. This is also true in storyboarding, where transitions are represented by edges in storyboarding. To connect a node and the edges, comments are used. This allows the creator to express knowledge about the interaction between the teacher and the user. Every scene could be broken down into even more details. This technique is used so the creator of the intelligent tutor does not become lost in the various interactions of the system and student. A storyboard has been created for the EINO system.

WEB APPLETS

The content that was chosen for the web applets is based upon The Infinity Project©[6]. The Infinity Project© was created as an attempt to get high school students more interested in engineering studies, and more specifically Computer and Electrical Engineering. Several professors at Southern Methodist University (SMU) noticed a drop in the number of engineering students and created a year-long class that high schools could offer to their students. The class, the Infinity Project©, was comprised of a textbook and an experiment manual that the professors developed and produced. In order to complete all of the experiments in the Infinity Lab Manual, the school must also purchase a lab kit, that includes a circuit board with the different components for each lab, along with speakers, and computer software that was used to control the circuit board[6]. In 2005, the University of Central Florida (UCF) decided to support the project and potential university students and hired graduate students to create some of the experiment experiments in a web format. This was done so that the students had the option of spending more time engaged in the activities that went beyond what the teacher had done in class with the lab kits on the experiments.

These web applets, UCF Web Applets, became the content for the ENIO system. Each web applet contains three tabs of content. The first tab provides the student with background theory about the concept they are about to learn. The second tab shows the student the overall layout/schematic of the components used in the experiment. The third tab is where the student performs the experiment. An example of an each tab can be seen in Figure 1.

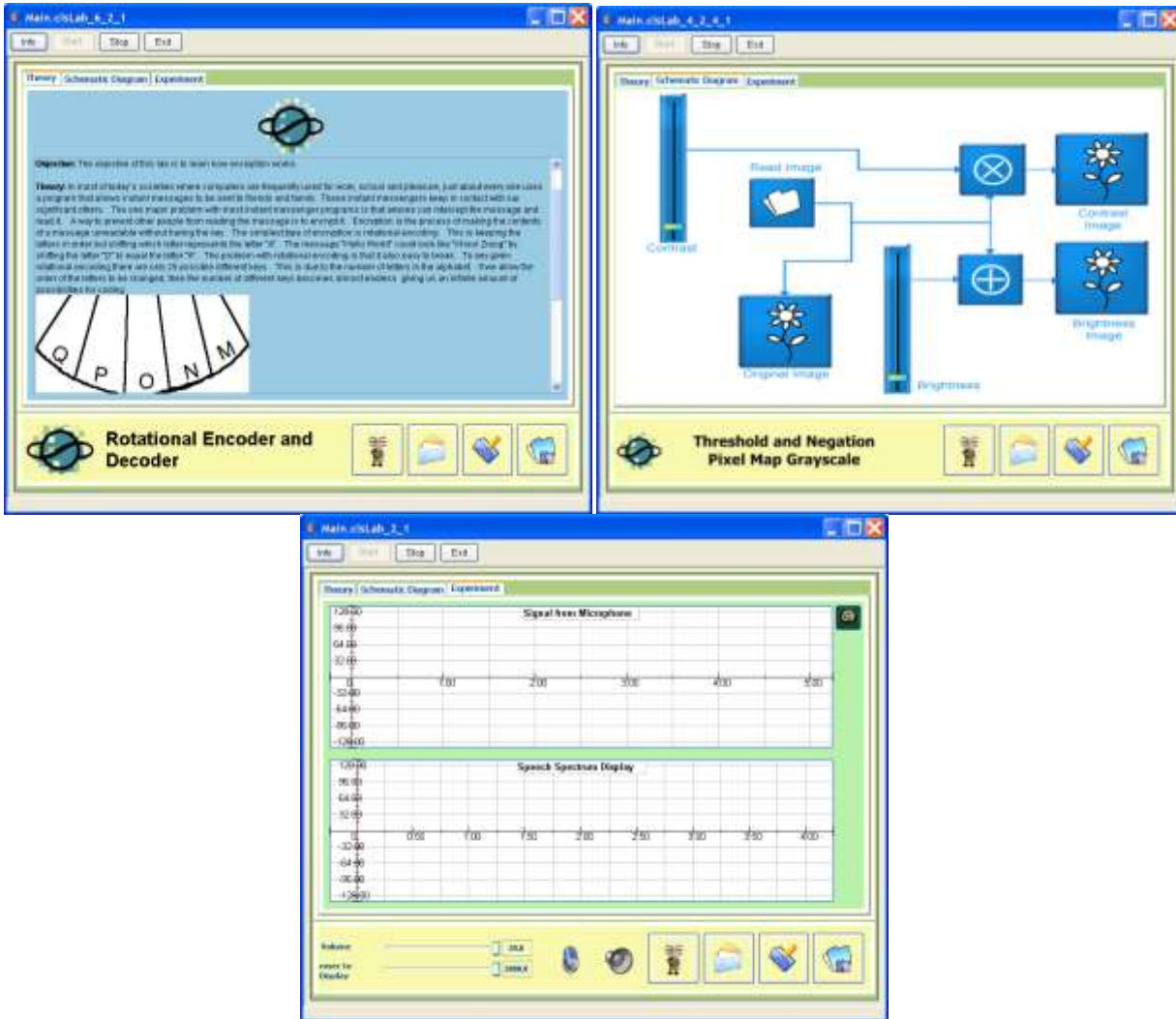


Figure 1: UCF Web Applets

EINO

One of the most challenging aspects of designing the EINO system is designing the tutoring system to function in conjunction with the UCF Web Applets. EINO is built to stay in the background until needed by the student so that the system does not aggravate the student. One of the simplest ways to look at a new project and understand how it works is to look at a block diagram. A block diagram for the standard operation of EINO, while running, is shown below, in Figure 2. When an experiment is started, EINO is running in the background ready to help the student. The student will not see EINO until the student requests help or EINO has advice to offer. The first and simplest block diagram to look at is the one for help requests made by the student. In this situation, help was requested by the student. EINO responds by opening a webpage with the EINO general help menu. The page displays a list of components used in the given experiment. The student is able to click on the component in which they are having problems in the experiment or about which he/she would like more information. EINO presents the needed information to the student. If the topic is considered a “big” topic, EINO follows up by asking a few questions about the topic to ensure the student understands the information. With the “small” topics, EINO simply presents the information requested. The “big” topics are key concepts that are necessary for the student to understand what that experiment is teaching, while the rest of the topics are considered to be “small.” The general help menu list included an option for simple experiment problems. EINO can present a step-by-step guide of how to

fix certain issues that might cause the experiment not to function, for example, the sound malfunctioning. The webpage with the EINO general help menu is only activated when the student requests help.

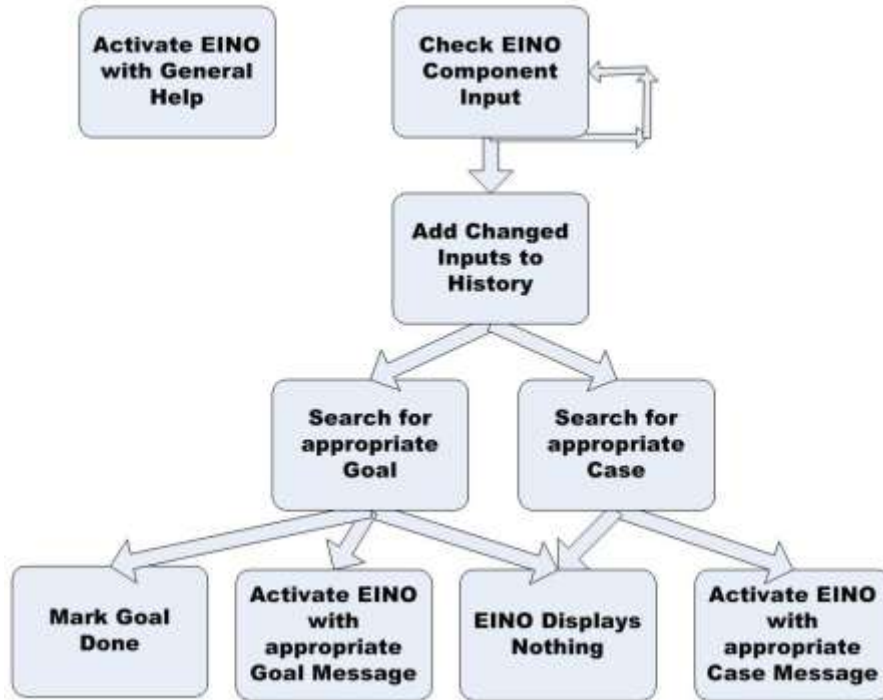


Figure 2: EINO Block Diagram

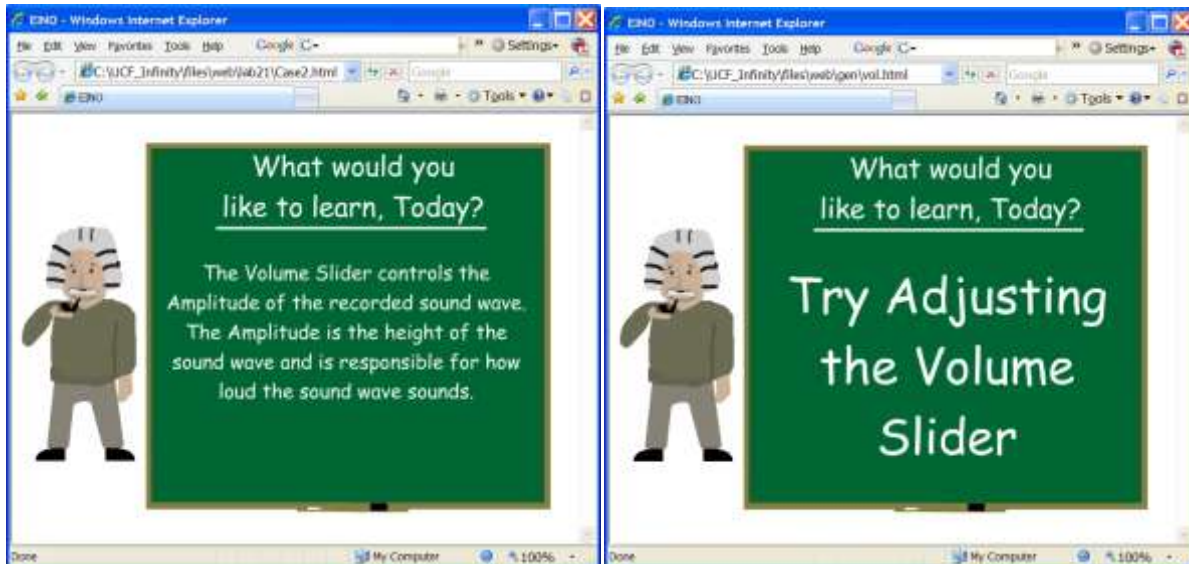


Figure 3: Output from EINO

The second way the EINO interface can be activated is shown in the second part of the block diagram of Figure 2. A sample activation of EINO can be seen in Figure 3. While an experiment is open, EINO checks roughly every second to see whether any of the inputs for that experiment have been modified by the student. A single lab is

an experiment designed to reinforce a topic currently being covered by the book. There are several different components used in the UCF Web Applets. A slider can change values very quickly while a microphone button has no numeric values. The only value that a button has is whether the button is pressed or un-pressed. Because a slider can change values rapidly, EINO requires the slider to maintain the same value for a count of three or roughly three seconds before the change is registered. This way, EINO knows that the student has stopped moving the slider and the value is not an intermediate value while the student was changing the values. If any experiment input has changed since the last check, the changed component's label and value is added into a history list. In the case of a button which has no numeric value, it is given a default value of one. This is so every input would have a value. The history list contains the last ten input changes made during the experiment. A list of the last ten input changes is enough to determine what the student is trying to accomplish in the experiment. After the item is added into the history list, EINO begins the first of two different searches.

This first search involves looking through the goals for the experiment. Every experiment is designed to work as if the student was performing the experiment in class. This implies that the student is expected to be following an experiment manual in order to complete the experiment. How could a system, like EINO, know if the experiment has been completed by the student? The first possible solution is to require the student to complete the experiment step-by-step according to the lab manual. This would make the experiments inflexible and discourage the students from practicing the lab on their own time as it would limit their ability to experiment with the lab. The other solution is to divide the step-by-step lab instructions into a set of goals. If the goals are arranged properly, the student will accomplish the same tasks but this will allow them the flexibility of completing the step-by-step lab instructions in any order that the student may deem appropriate. A single goal could be anything that deals with the input into the lab, like setting the volume to a value of 58. EINO is configured to use the second approach. The goal search looks through history list to see if the last action meets any unfinished goals. A goal may require that another goal be met before it is able to be considered. In other words, certain goals will need to be done in a set sequence. This makes the goal search more difficult to perform. EINO gives a higher priority to the goals that must be done in order than to a single goal. If the student is in the middle of a set of goals and makes a mistake, EINO leaps to the aid of the student with a "Just in Time" hint of what the student should do next. When a goal or a goal set has been met, EINO marks those goals as done and they are removed from the search list until all of the goals are met. The other possible action for EINO is that after the goal search is done is to stop and do nothing. If help is not required by the student, EINO hides in the background.

The second search the EINO performs involves looking through the cases library for the lab. One way to describe CBR is to think of a filing cabinet that is filled with multiple folders[7]. The first step in a case-based system is to find the closest case, or problem, in the filing cabinet to the one being presented to the system. In the next step, the system reuses and/or revises the output from the closest case and applies it directly to the one presented by the student. The last step would be to save the new case and the subsequent output for reuse later. The whole process can be described in three words and those words are: retrieve, reuse, and retain[7]. The concept behind the CBR system in EINO is to recognize patterns in the student's actions and produce help to the student before they ask for it. A single case contains a history list that has the pattern that EINO is seeking for and a number of the EINO webpage to display for the student. The student presents a list of actions through use of the lab and EINO looks through the previous cases to find one close enough to the students' work. Then EINO presents the help that was needed in the previous attempt. If no match is made, then EINO will wait until the student's actions match something in EINO's case library. The output of EINO when a case is matched is typically an explanation of a component in the lab. A third way to generate an output from EINO is to change an input while the lab is off. Every lab has an on and off button. If the lab is in the off state, then the On button is displayed. Likewise, if the lab is in the on state, then the Off button is displayed. While in the Off state, some of the inputs for the experiments will not function as the student would anticipate it to operate, if at all. This is simply due to the fact that some of the connections of the lab components are not made until the lab is turned on. These "malfunctions" may cause the student to question what is happening and ask for help. Therefore, EINO leaps to action and will give a gentle reminder that the lab is not turned on and should be turned on to function appropriately.

RESULTS

The EINO system underwent a series of tests of two main factors. These two factors are:

1. The functionality of the EINO intelligent tutoring system
2. The helpfulness of the EINO intelligent tutoring system

The first sets of tests were designed to test the functionality of EINO. The functionality of the EINO intelligent tutoring system is tested to determine if EINO can perform as it is expected to perform based on a set of requirements. For example, would EINO offer assistance to the student when a match was made in the case library. The system passed all of the functionality tests as expected and will not be discussed further.

The helpfulness of the EINO intelligent tutoring system is tested to determine if EINO is truly helping the student understand the basic engineering principles. EINO may be providing comments to the student, yet if the comments do not make a difference in the students’ understanding of the material, then EINO may be of no benefit. The helpfulness of the EINO is tested by performing a real life simulation and should provide information on the helpfulness of EINO along with how well EINO functions in the actual world of the classroom or lab. This test was verified by five test subjects who consented to test the functionality of three different experiments. Three of the test subjects were within the first two years of their college education. The reason these students were chosen is that these are the type of students that are the targeted audience of the EINO intelligent tutoring system. The procedure to perform this test was standardized as following: every test subject was given a copy of the step-by-step instructions for each lab from the lab manual and unlimited time to “play” with each lab.

After the test subjects finished each lab, a brief survey about the helpfulness of EINO was administered. The survey consisted of eight (8) questions about the student’s experience with the system. From these completed EINO surveys, it was discovered that the most common help message presented to the student is “the lab is Not On” message. This means that most students assumed the lab was ready to go and “On” when they started the applet. There were two significant questions on the survey that dealt with the helpfulness of EINO. The first question dealt with whether EINO was easy to understand. Fourteen out of fifteen surveys responded that EINO was easy to understand. The last survey was not filled in because the test subject did not trigger any response from EINO during the lab. The second question dealt strictly with the helpfulness of EINO. The question asked the test subject to rank the helpfulness of EINO on a scale of one to ten with ten as the best. The overall results of this question can be seen in Table 1.

Table 1: Overall Helpfulness of EINO

EINO Helpfulness Score	Number Found Helpful
10	8
9	2
8	2
7	1
0	2

This signifies that twelve of the test subjects gave the helpfulness of EINO an eight or higher. There were two zeros that were given by the respondents. The first zero was given because the test subject never triggered a response from EINO on one of the experiments. The other zero was given because one student thought that EINO kept offering help a little too often and found it annoying rather than helpful. There is no single way to prove a system is helpful as the term “helpful” has different meaning to every individual.

Another part of the helpfulness of the EINO system is the comprehension level of the system. There could be a chance that the EINO system is helpful yet difficult to understand. The survey also attempted to ascertain how easy EINO is to understand to students involved in the use of the program. Fourteen out of fifteen surveys of the test subjects responded that EINO was easy to understand. The tests have proved that that a good percentage of students find EINO “helpful.” These findings validate the usefulness of the EINO intelligent Tutoring system for use in the classroom or lab situation.

CONCLUSION

The concept of an intelligent tutoring system or CBR or even a combination of using the two concepts together is not a new idea. In fact, the ANDES system has been in continual development for over ten years and is still undergoing changes. With each new development and application of an intelligent tutor, the design and process of these systems must also change.

As discussed above, there are two key types of help or instruction. The first type of help that is offered is conceptual assistance. Conceptual help involves helping the student understand the theory behind what is occurring in the particular experiment. The EINO intelligent tutoring system solves the problem of providing conceptual help to the student in several of ways. The first way is the general help button. At a click of a button the student has access to a list of the major components in the particular lab. Then by clicking on the subject matter with which the student is having problems solving, the student has a resource of conceptual help in any experiment. The first resource the student has access to is the theory tab, which is displayed by default on start up of the lab. The second resource is the general help button and the third resource is the Infinity textbook, which has all the theory for all of the experiments in a hard copy format, in the order which they are displayed. Each of these three ways presents the information to the student a different way. A second way EINO provides conceptual help is with the CBR system. The CBR system in EINO is looking for patterns in the students' input into the system. The student may keep changing the certain component in order to see the effects of what the component does. After a predefined number of changes, EINO displays a message describing the component and what the component does. This is the second way EINO provides conceptual help to the student.

The second type of assistance is procedural help. Procedural help involves helping the student answer the question "What do I do next?" Procedural help within the UCF Web Applets is a tricky matter. The student will be using a lab manual. The lab manual provides for the student a step-by-step instruction on how to complete the lab. Even with the lab manuals, a student could have problems. This is why EINO still provides some procedural help in the form of a "Just in Time" hint. A "Just in Time" hint is produced when EINO knows what group of goals the student is working through and the student makes a mistake by changing the wrong input. The "Just in Time" hint will suggest to the student to change a certain input component.

When both of these help systems are combined to function as one, the student is provided with all of the assistance they could require to successfully completing the course. This is the overall goal of an intelligent tutoring system.

FUTURE WORK

The biggest challenge that has been discovered in EINO involves the manner in which EINO delivers messages to the student. Each message is displayed in an Internet Explorer (IE) window. Every time EINO delivers a message, a new IE window is opened. This means the old message is still left opened in the background. This flaw could possibly confuse the student. In the student's mind, EINO could be suggesting to try two items at the same time, since there are two items opened. In the real life simulation none of the test subjects had a problem with the leaving of EINO's messages open.

What can be done in the future to make a more efficient version of EINO? The most obvious answer would be to create a better interface for EINO. The Flash™ interface works well but this design involves combining two very different programming languages. The whole process can become overwhelming and confusing to the creator. The solution for this can be accomplished by incorporating the interface and web applet into one program instead of the two window approach used by EINO. Although the two window approach works, it can become frustrating switching back and forth between windows. When both windows are combined, the tutoring messages would seem more fluent from the lab and could respond to the student's actions more rapidly. The lab's interface would have to be expanded to allow room for a dialog area in which EINO could communicate effortlessly with the student. This would also allow for a redesign of the layout of each of the lab's interface which would allow the lab's interfaces to become more uniform. A more concise design with greater potential can be created if the tutoring system and the web applets are built at the same time. However, the EINO system has proved to function well in its current form and design.

AUTHOR INFORMATION

James Hollister received the B.S. degree in Computer Engineering from The University of Central Florida (UCF), Orlando in 2005. He received his M.S.Cp.E. degree in Computer Engineering, Intelligent Systems track from UCF, Orlando in 2007. James is currently pursuing his Computer Engineering PhD. He is a member of the Tau Beta Pi, Eta Kappa Nu and Delta Epsilon Iota Academic Honor Societies. James is also an Eagle Scout and a member of Mensa. He worked with the Infinity team at UCF, which created an online lab environment for students to practice labs, from 2005 till 2007. James spent two years teaching math at Valencia Community College, Orlando. He is currently working as a teaching assistance in the college of engineering at UCF and working on the Lifelike Project. The goal of the Lifelike project is to create an avatar that looks and acts and reacts in a lifelike manner. His major research interests are knowledge-based systems, artificial intelligence, and avatars.

Samuel M. Richie received his Ph. D. degree in Electrical Engineering from the University of Central Florida in 1989. He has been a faculty member at UCF for 30 years and is currently the Assistant Director of the School of Electrical Engineering and Computer Science, as well as the undergraduate coordinator for the Electrical Engineering and Computer Engineering programs. He previously served as the Assistant Dean for Distributed Learning for the College of Engineering and Computer Science at UCF. He has been principal or co-principal investigator for external research in the areas of surface acoustic wave transducer modeling and computer aided design, the development of modeling software for bi-directional and three-phase unidirectional surface acoustic wave transducers, and surface acoustic wave reflective array correlators. Other research includes development of algorithms for the synthesis of finite impulse response functions for systems with completely arbitrary magnitude and phase specifications, surface acoustic wave resonator and resonator filter modeling and CAD, surface acoustic wave proximity coupled, waveguide coupled resonator filter modeling and CAD. Additional research areas have included communications systems simulation including the U. S. Navy MARK XV identify friend or foe (IFF) system and advanced concepts in point scanning computer image generation algorithms, architecture, and system development.

Arthur R. Weeks received his Ph.D. degree in Electrical Engineering from the University of Central Florida in Orlando in 1987. After completion of his Ph.D., he spent one year at the Royal Signals and Radar Establishment in Malvern, England studying laser beam propagation. Dr. Weeks left his Vice President of Corporate Technology position at Invivo Research Inc. to join the Electrical Engineering Department at the University of Central as an Associate Professor in Jan. 2003. Dr. Weeks' interest include color image processing techniques, the reduction of noise within images using adaptive nonlinear filters and the use of Artificial Neural Networks in pattern recognition. Dr. Weeks is a senior member of IEEE, and is a member of SPIE and Tau Beta Pi.

REFERENCES

- [1] M. Landsberg, "Budget cuts push some classrooms way over capacity," in *Los Angeles Times* Los Angeles, 2009.
- [2] K. VanLehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill, "The Andes Physics Tutoring System: Lessons Learned," *International Journal of Artificial Intelligence in Education*, vol. 15, 2005.
- [3] A. S. Gertner, C. Conati, and K. VanLehn, "Procedural help in Andes: Generating hints using a Bayesian network student model," in Fifteenth National Conference on Artificial Intelligence, 1998, pp. 106 - 111.
- [4] P. M. Regan and B. M. Slator, "Case-based Tutoring in Virtual Education Environments," in Collaborative Virtual Environments Proceedings of the 4th international conference on Collaborative virtual environments, 2002, pp. 2 - 9.
- [5] K. P. Jantke and R. Knauf, "Didactic design through storyboarding: standard concepts for standard tools," in ACM International Conference Proceeding Series; Vol. 92 Proceedings of the 4th international symposium on Information and communication technologies, 2005, pp. 20 - 25.
- [6] G. Orsak, S. Wood, S. Douglas, D. M. Jr., J. Treichler, R. Athale, and M. Yoder, *The Infinity Project Engineering our Digital Future*. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.
- [7] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AICom - Artificial Intelligence Communications*, vol. 7, pp. 39 - 59, 1994.

NOTES