

Improving Student-Tutor Dialogues in E-Learning

Stuart Garner
Edith Cowan University, Australia

s.garner@ecu.edu.au

Abstract

In flexible, asynchronous e-learning environments, student-tutor interactions are usually by telephone, email or fax. In courses that utilise software in their teaching, many such interactions involve students requesting assistance from a tutor in connection with the use of that software to solve problems. Not only is it difficult for a student to explain a problem which they are having, but it is also difficult to respond to such a query without being able to "walk through" examples by using a white board or via the software which is being used within the course.

There are now tools available that can improve such interactions by recording computer screen activity in some manner. Most tools also allow voice-over narratives to be recorded. This paper gives an overview of these tools, discusses how they can be used to provide better student-tutor dialogues and describes our experiences with a summer school course in which Lotus ScreenCam was used for such dialogues.

Keywords: e-learning, dialogues, flexible Learning

Introduction

Educational institutions today have an ever-increasing number of distance learning students enrolled in a wide variety of units. The rate of increase will probably become larger as institutions put more courses on the Web, thereby encouraging yet more students to study at a distance in an asynchronous, flexible e-learning environment.

Communication between distance learning students and tutors is usually by telephone, fax machine or email, the latter probably being most popular nowadays. Email is also frequently used as a communication medium between tutors and students who are not studying at a distance (campus based students), saving the students from having to make appointments with their tutors.

Problems arise when students wish to articulate the difficulty they are having in a particular area of the curriculum or with a particular problem solving exercise. This is particularly the case in subject areas that involve problem solving such as mathematics and computing. Tutors have often commented how difficult it is to conduct

technical sessions using these impoverished media (Domingue and Mulholland 1997, 1998). A typical student - tutor dialogue concerning a programming problem is as follows (Domingue and Mulholland 1997):

*student: "I'm having problems."
tutor: "What sort of problems?"
student: "I don't understand."
tutor: "What don't you understand?"
student: "I don't understand anything about this latest assignment."
tutor: "Is there anything in particular about the assignment that you don't understand?"
student: "It doesn't make sense"
tutor: "Can you outline any concepts which are causing your problems?"
student: "Recursion I think"
tutor: "Anything in particular about it?"
student: "I just can't get it to work"
tutor: "What have you typed in?"
student: "Well the first line is...."*

An example of an email message which I received recently from an International student concerning her difficulty with a computer programming concept was:

It's me again, Eileen. I would like you to help me up by clearing some of my doubts, as there is no one that i could turn to, at this moment. I'm very confused with the two-dimensional arrays. From Schneider ex 7.5.5

Material published as part of this proceedings, either on-line or in print, is copyrighted by the author with permission granted to the publisher of Informing Science for this printing. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the author.

Improving Student-Tutor Dialogues in E-Learning

onwards to 7.5.8, I don't understand how to get the output. I'm very confused with the j and k variables. I hope that you could get back to me as soon as possible before software paper. thanks a lot, Stuart.

It can be seen from the above examples how difficult it is for tutors to help students when students cannot adequately explain their difficulties via telephone or email.

In computer programming courses, students often have difficulty in continuing with the development of a program because of a simple bug. Because of the difficulties encountered describing the problem to their tutor, they will usually send the program to their tutor as an email attachment. The tutor then has to check out the program and respond with hints as to where to "look" for the source of the programming error. This often proves to be unsatisfactory and the tutor may well end up fixing the bug so that the student can carry on.

In this paper I will discuss some of the tools that are available to help improve such student - tutor dialogues and describe some of the experiences that I have had when using the Lotus ScreenCam tool in such dialogues.

Types of Tool to Improve Student - Tutor Dialogues

There are now tools available that can improve such interactions by recording computer screen activity in some manner. Many tools also allow voice-over narratives to be recorded. The tools fall into two categories:

- Tools that are specific to a computer application.
- Tools that are generic and can be used with any computer application.

Tools That are Specific to an Application

There are several applications available which have in-built facilities enabling some form of screen capturing to take place together with the ability to annotate the screen captures with either text or an audio commentary. Some applications allow a series of screen snapshots to be taken and then played back; others dynamically record all screen activity during a computer session. Three such examples follow.

The Internet Software Visualization Laboratory

This tool helps students learn Prolog programming in a graphical environment and supports individual exploration and both synchronous and asynchronous communication (Domingue and Mulholland 1997, 1998). It allows students and tutors to communicate with each other via "software visualizations", which are representations of programs and their execution using techniques such as typography, graphic design, animation and cinematography. Movies are sequences of screen captures that can be replayed using a VCR type interface control.

Sam Multimedia Demo

Sam, as in "Play it again..", allows the recording and replaying of multimedia demos together with synchronised audio commentaries within the StatPlay product (Les, Cumming et al. 1997; Cumming, Finch et al 1999). StatPlay is a sophisticated tool to help students learn key concepts in statistics. Several educational possibilities are suggested for Sam in the paper (Les, Cumming et al. 1997) including: learners making their own demos and then submitting them as queries for tutor comment; and tutors recording explanations on student errors.

ASCENT CASE tool version 2

ASCENT is an educational computer assisted software engineering (CASE) tool (<http://www-scm.tees.ac.uk/research/modsim/softeng/welcome.html>) that includes a new facility to allow students and tutors to record presentations of graphical models that they have produced (Hoggarth and Lockyer 1996). These so-called "automated walkthroughs" comprise a series of snapshots of screen images taken as visual models are "built up" by a user, together with text annotations. Figure 1 shows an example of a screen taken from such a walkthrough.

So far such walkthroughs have only been created by tutors, for student use, as part of specific curriculum sequences. In an evaluation, the majority of students commented that the incremental view of development (of visual models) displayed by such walkthroughs aided their understanding and that they had only grasped the systems development methodology that they were learning when they had witnessed the walkthrough development of examples.

Of course, the walkthrough facility can be used in student-tutor dialogues enabling students to clearly explain problems that they may be having with a particular visual model and tutors being able to give direction to students.

Tools That are Generic

There are several dynamic screen capturing tools currently available that allow the capture of all screen activity, including mouse movements, plus a voice-over narrative. Most are Windows based with the exception of CameraMan which is also available on the Macintosh. They are useful in creating "show and tell" type movies or WYSYWD (what you see is what you do). Royalty-free players are available for such movies to be played back.

The tools include:

- Lotus ScreenCam (<http://www.lotus.com>)
- CameraMan (http://www.mwg.com/tools/cameraman_page_general.htm)
- HyperCam (<http://www.hyperionics.com>)
- Microsoft CamCorder (<http://support.microsoft.com/support/kb/ARTICLES/Q159/5/61.asp>)

Lotus ScreenCam

This is the best known of this category of tool and is used in both Education and Industry. An example of its use in Education is in the creation of movies for the CD ROM multimedia course in software development (Garner 1997, 2000). It has been used by many industrial organisations including PeopleSoft who recorded approximately 200 movies about their computer applications for use by their customers (Lotus 1997).

The ScreenCam interface is shown in figure 2. It is very straightforward to use with its VCR type controls and movies with or without sound can be made. The latter can be annotated with text captions and this is very useful if a requirement is to keep file sizes to a minimum, for example when sending movie files as email attachments.

Movies created by ScreenCam can be saved in a special ScreenCam format or as Video for Windows ("AVI") files. The advantage of the former is that the files are

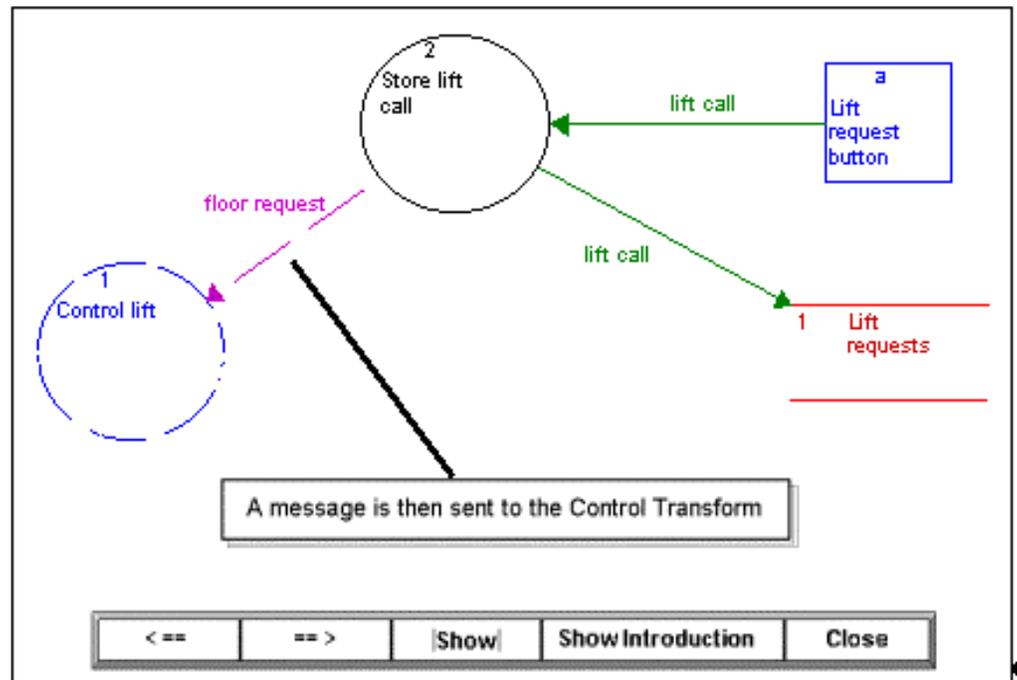


Figure 1



Figure 2

approximately one third the size of "AVI" files. The latest version of ScreenCam also permits the user to pause when making a movie and allows various sound qualities to be selected, ranging from 8bit, 11KHz to 16 bit, 44KHz. There is of course a trade off between the quality of sound and the movie file size.

In creating movies for student-tutor dialogue, I have found in practice that it is necessary to use text captions without audio narrative in order to keep the file size manageable. ScreenCam has an excellent caption editor allowing the user to quickly produce a set of captions with a choice of backgrounds. The user can also specify the screen position of each caption. During the recording of a movie, the captions can be shown and hidden in the sequence in which

Improving Student-Tutor Dialogues in E-Learning

they were originally saved by using two predefined function keys. A movie which is approximately 1 minute in length with 4 captions recorded with 256 colours at a screen resolution of 800 x 600 is around 125KB in length. This can be reduced to just 55KB if compressed into "Zip" format, and can be comfortably sent as an attachment to email.

CameraMan, HyperCam and CamCorder

These three tools all capture screen activity and permit audio narratives in a similar way to ScreenCam, however files can only be saved in "AVI" format which are, as mentioned above, much larger than the ScreenCam specific files. The advantage of "AVI" movies is that they can be integrated into any Video for Windows applications such as Macromedia Director, Adobe Premiere and they can be converted to QuickTime format. Such quality may well be required if a sophisticated multimedia title is being produced, however in the context of producing movies to improve student-tutor dialogues, such sophistication is not needed and file size reduction is probably the most important feature needed in the movie recorder. Also, these three tools do not support text captioning, a very important requirement when files sizes have to be kept to a minimum for use with email.

Use of Lotus ScreenCam in Practice for Student-Tutor Dialogues

During a recent summer school at Edith Cowan University, I made use of ScreenCam for student-tutor dialogues within the Software Development 2 unit. Between lectures and laboratory sessions, students had no contact with me as I was off campus, however I did have access to email at home enabling students to send me ScreenCam movies of any programming problems that they were having. In addition to movies, students would also send the programming code enabling me to use this when making a "reply" movie. An example of a screenshot taken from a movie, which was sent to me by a student, is shown in figure 3.

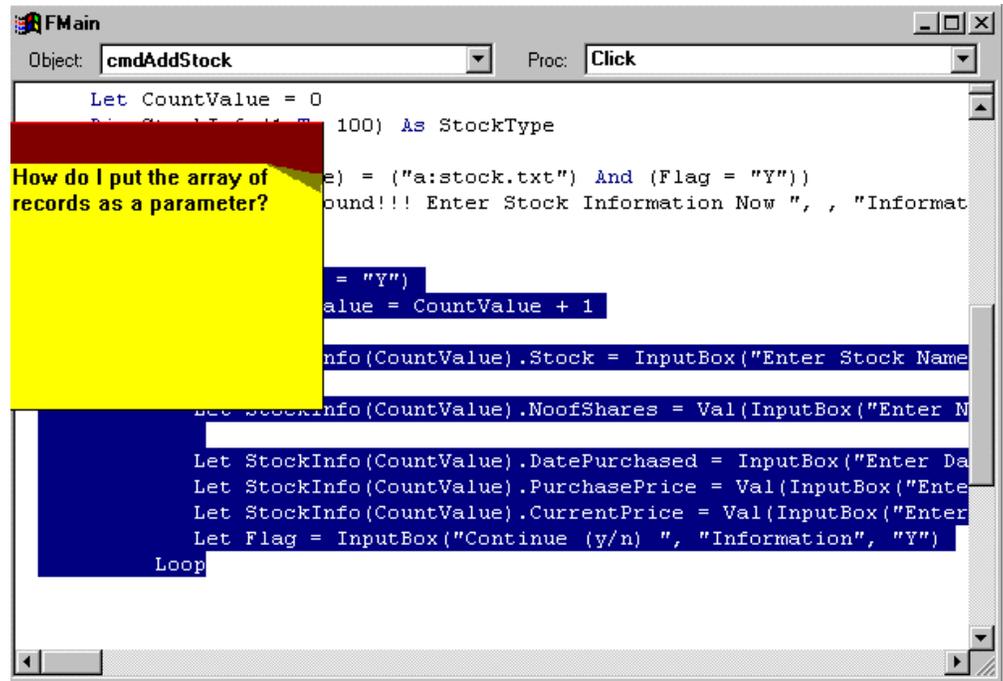


Figure 3

The movie had several captions and concerned a problem that this student was having with passing arrays to subprograms in Visual BASIC. A screenshot taken from the movie, which I made and subsequently sent back to the student, is shown in figure 4.

The screenshot in figure 4 includes a text caption that has nothing to do with the original student problem. It is the sort of comment that I would make if I were looking at the code that a student had produced in a laboratory session. In the rest of the movie, I was able to make suggestions on how to overcome the original problem and I also included a captioned comment about the lack of comments within the student's programming code. By using ScreenCam, I had been able to engage in a richer asynchronous dialogue with the student than I would otherwise have done by conventional means.

Discussion and Conclusion

The tools to improve student-tutor dialogues which are specific to computer applications are obviously only useful if those applications are used within our teaching. However, the techniques used could be incorporated within other educational software and perhaps developers should give some thought to this idea. The advantage of having the dialogue facility built-in to the software is that there

One further advantage that I have found when requiring students to make movies of their problems is that they are forced to think more deeply about those problems during the structuring of movies. This in some cases has helped them solve the problems themselves.

An interesting idea is that a database of previous student-tutor dialogues should be created which would then be used as a resource to help students in their learning (Mayes and Neilson 1996). In my view, this is something worthy of further consideration.

Another possible use for such movies would be in the production of rich feedback to students on their

assignments. Tape-recorded commentaries have been used to provide feedback on essays (Anson 1997) however this is probably not appropriate when commenting on the execution of a computer program or the layout and formulae in a spreadsheet. Such movies would be much longer than those produced for short student-tutor dialogues and would need to include voice over narratives for speed of production. The file sizes would be large and movies would need to be returned on a high capacity removable storage medium such as a Zip disk that can hold 100 to 250 MB. The feedback movies could also be made available for use by other students who will study the same subject in future semesters. By watching movies of the feedback given to other students, students are able to learn in a vicarious manner in a similar way to "listening in" to a tutor discussing a problem with a fellow student (McKendree, 1997).

Finally, I would say that as more courses are delivered flexibly using technologies such as the Web, it becomes more important that we have richer ways of communicating with students. Some of the tools described in this paper may well form the basis for such rich communication.

References

- Anson, C. M. (1997). In Our Own Voices: Using Recorded Commentary to Respond to Writing. *New Directions for Teaching and Learning*, Spring 1997(69), 105-113.
- Cumming, G., Finch, S., & Thomason, N. (1999, 4-7 Nov 1999). *Educational design for effective learning: Building and using a computer-based learning environment for statistical concepts.*

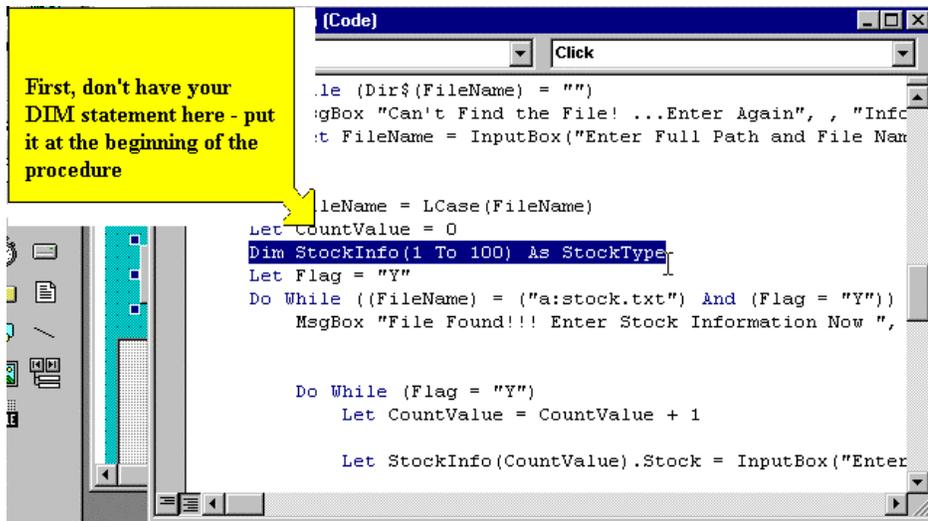


Figure 4

would be no need for generic screen activity recording tools to be purchased. This is important for institutions during this period of shrinking budgets and also of course for students who already have to cope with purchasing expensive textbooks.

Currently, however, the number of such applications with in-built student-tutor dialogue tools is very limited and most of us will need to use a screen activity recorder such as ScreenCam if we wish to improve such dialogues in the manner outlined in this paper. Also, because of current bandwidth limitations when using modems, it is likely that we would make movies with captions rather than with audio narratives in order that the movies can be sent via email attachments.

There would mean that students would have to purchase ScreenCam and this would have to be a requirement for the course. The educational cost of ScreenCam is around AU\$60 and it is also bundled with Lotus SmartSuite which itself only costs around AU\$90 for educational use. Such a requirement can probably be justified to students if ScreenCam were used in several units of the course.

The time required to make such movies also needs to be taken into consideration. To make a two-minute movie with audio narrative takes two minutes! Movies with text captions take longer to make, however with practice the time per movie can be reduced and is not much greater than producing a detailed email description of a problem or a detailed email explanation.

Improving Student-Tutor Dialogues in E-Learning

Paper presented at the International Conference in Computer Education 99, Chiba, Japan.

SAM and Multimedia Demo. Paper presented at the ASCILITE 97, Perth, Western Australia.

Domingue, J., & Mulholland, P. (1997). Teaching Programming at a Distance: The Internet Software Visualization Laboratory. *Journal of Interactive Media in Education*, 1, on-line - 28 pages. Accessed Oct 2, 2000 at <http://www-jime.open.ac.uk/97/1>.

Lotus Corporation. (1997). *White Paper (Technical)*. Accessed 12 Sep, 1999 at <http://www.lotus.com/home.nsf/welcome/screencam>.

Domingue, J., & Mulholland, P. (1998). An effective web-based software visualization learning environment. *Journal of Visual Languages and Computing* (9), 485-508.

Mayes, J. T., & Neilson, I. (1996). *Learning from other People's Dialogues: Questions about computer-based answers*. In B. Collis & G. Davies (Eds.), *Innovating Learning with Innovative Technology*. Amsterdam: North-Holland.

Garner, S. (1997). *Cost Effective Multimedia Production for Frequently Updated Courseware*. Paper presented at ASCILITE 97, Curtin University, Perth.

McKendree, J., & Mayes, J. T. (1997). *The Vicarious Learner: investigating the benefits of observing peer dialogues*. Paper presented at the Computer-Assisted Learning Conference (CAL '97), Exeter, UK.

Garner, S. (2000). *Systems Analysis and Design using ASCENT*. Perth: Knowledge Base.

Hoggarth, G. R., & Lockyer, M. A. (1996). Systems Development Methods Guidance and CASE: An Integration between CASE and CAL. *Software Engineering Journal* (May 1996).

Les, J., Cumming, G., Thomason, N., & Finch, S. (1997). *Explanation and Guidance in a Learning Environment: Recording and Using*

Biography

Stuart Garner is a member of the school of Management Information Systems within the faculty of Business and Public Management at Edith Cowan University in Perth, Western Australia. He teaches in the areas of systems and software development and Web commerce development.