Course presentation by computer-mediated communication systems

Ian Sommerville,
Computing Dept., Lancaster University,
Lancaster LA1 4YR, UK.

Phone: +44-524 593795
E-mail: is@comp.lancs.ac.uk

For many years, the traditional method of instructor-student communication has been the formal lecture where a lecturer presented material to a group of students. This approach was developed last century as a teaching system based solely on tutorial teaching became impractical due to increasing demands for higher education.

We are now facing a comparable situation where our current resources are being more and more stretched by demands to educate more and more students at a lower and lower cost per student. Although higher education systems vary radically from country to country, I believe that this demand of more for less is fairly general, in Europe and North America, at least. The system based on lectures has probably never been very effective educationally; it is now less and less cost-effective as class sizes increase and their is a need to present the same lecture to different groups at different times.

Rather than present conventional lectures, I decided to adopt a different approach to the teaching of a final-year course on software engineering. This course was titled 'Reliable Systems Development' and the major topics covered were Computer-based Systems Engineering, Requirements Engineering, Software Reliability and Safety-critical Software Systems. The emthod adopted was based on communications with the student through electronic systems (E-mail and a computer conferencing system). The different work packages were disseminated through the conferencing system. The course was based on a textbook plus further notes which were distributed to the students.

In the past, I had set a limited number of assessed exercises in my course but I always felt that the coverage offered was inadequate. In the UK, it is not usual to have teaching assistants associated with a course so it was impractical to significantly increase the amount of assessed work. Accordingly, apart from a group requirements project, all coursework was self-assessed.

In principle, it would have been possible to run the course using this approach without any personal contact with the students. However, I did not feel that this was the best approach. Rather, I wanted to use some of the time allocated to lectures to present background material, other items of interest and so on. The class therefore had a weekly meeting for the 10 weeks of the course. The topics covered at this meetng included:

1. Guest lectures from software engineering professionals in industry.
2. Lectures from researchers in the Department about their research work.
3. Background material and discussions. For example, I showed a 1-hour video which covered air crashes and possible software faults and used this to stimulate discussion on professional responsibility.

This experiment in software engineering education was partially motivated by economic factors. Although we have not yet reached the stage where multiple lecture presentation is necessary, it is possible to foresee that situation occurring. It was also prompted by a feeling that students did not get much out of lectures. Finally, I felt that there was a significant lack in some student's education in that some choose to do almost all of their course work using personal computers and did not, on an everyday basis, make use of electronic mail and other forms of computer-mediated communication.

In the remainder of this paper, I present some general information about how I ran the course and how it was evaluated. I then highlight the problems which arose and summarise my conclusions on the experiment.

Conference structure and usage

The first problem which I faced was which conferencing system to use and how to structure the conferences. We had available a local bulletin board system plus a conferencing system called Caucus which had been used in other departments for computer-mediated teaching. Neither system was particularly good, both were Unix-based with poor, command-line interfaces. I chose the Caucus system because it had better security control and because it had reporting capabilities. None of the material I submitted was confidential but I wanted to reassure students that no-one apart from their classmates could read their submissions. I also wanted regular reports of who used the system so that I might be able to correlate this with preformance in later assessments.

Having chosen the conferencing system, the next problem was to decide on the conference structure. The conferencing system distinguished between items and responses and allowed separate control over these. I wanted to make a corpus of course material available in one place so I decided I needed a conference where students could not set items but could only respond to them. However, I also wanted to encourage students to stimulate discussion so there was a need for another conference where that was possible. I therefore established 2 conferences to run the course:

1. SE3_info was a conference where course material was published weekly. This is discussed later in the paper. In this conference, students could respond to course material but could not add their own items. I suggested that their responses should be points of clarification rather than more general discussions.
2. SE3_forum was a more general conference which acted as a forum for discussion. Anyone could post items or respond to items. I encouraged students to post items on any topics they wished although I suggested that they should try and make them relevant to the course in some way.

The conferences were used as I suggested but I discovered that the conferencing system did not allow for simple navigation from one conference to another. For example, it was not possible to link, in any way, entries in SE3_forum with items in SE3_info. I certainly found this to be a problem although I am not sure whether it caused students any real difficulties.

The conferencing system had a poor user interface and limited documentation. I deliberately did not make any attempt to extend the documentation and made the point to the students that learning to use poorly designed interfaces was a skill they should develop. I assumed that they would have no real difficulty in this respect. I was wrong! I discuss this problem in the conclusions to the paper.

Introductory information

At the start of the course, I handed out a number of documents discussing the approach which was planned and providing various information about the course and its organisation. The key documents were:

1. A general introduction to course presentation. This emphasised the importance of active, self-managed learning and told the students what was expected of them. This document is reproduced (in a slightly edited form) in Appendix 1.
2. Information about how to get started with the conferencing system and how to use a Postscript viewing system to look at OHP slides associated with each part of the course.
3. A timetable of class meetings and what was planned for each meeting. A meeting was organised each week with topics decided in advance as discussed above.

   This material was also, of course, made available on the conferencing system.

Weekly submissions

As the course instructor, I posted information weekly which could be accessed and commented on by students. Normally, I added 2 items to the SE3_info conference each Friday:

1. Set work for the following week. This was made up of required reading, further suggested reading, a number of self-assessment problems and the filename where the OHP slides for the course were stored. Midway through the course, students requested that key points in the material should be indicated and this was added to all entries. An example of weekly set work is shown in Appendix 1.
2. Sample solutions for the self-assessment problems which had been set two weeks previously. I advised students to use the problems to drive their reading of the material and expected them to complete these within two weeks (this was grossly over-optimistic as I discuss later). The solutions were complete answers and, where appropriate, a commentary was included.

In the first year of presenting the course, there was a heavy preparation overhead in making this material available.

Project work

As well as self-assessment exercises, the students were also expected to complete an assessed group project which involved developing a requirements specification and a formal specification in Z for a computer-system to manage the submission, distribution and assessment of student assignments. Groups were expected to use the conferencing system to communicate with the course instructors to derive the requirements for this system and to discuss individual projects with the instructors involved (the project spanned 2 courses). Our hope was that this would reduce the overhad involved in talking to individual students about their work and that groups would interchange information about the requirements.

This system did not work at all. I guess because the work was to be assessed, students were very protective of ‘their’ information and did not wish to make it public. Only one group used the conferencing system to ask questions about the requirements and they gave up fairly quickly when it became clear that no other group was providing shared information.

Method evaluation

The objective evaluation of any different approach to teaching at this level is almost impossible. There are simply too many variables involved and evaluation must therefore be subjective and intuitive. I planned the evaluation of this approach using 2 approaches:

1. A comprehensive questionnaire (36 questions) which students filled in during the last weekly class meeting. This was intended to collect student's opinions about the course and their suggestions for improvement. Students were generally very constructive (but not uncritical) in their response to this. This questionnaire was in addition to our normal course questionnaire which is completed by students at the end of every course.
2. A comparitive evaluation of student performance. A sub-set of the students who took the software engineering course also took another course which I presented on software management. This course was deliberately presented as it had been in previous years (conventional lectures; conference access was denied; self-assessment was limited). The performance of students taking both courses would be compared. The performance of the group taking the software engineering courses would also be compared with the previous year group's performance.

At the time of writing, the course evaluation questionnaires have been completed and anlysed. Students have not yet had any final assessments so it is not possible to make examination performance comparisons at this stage. The questionnaire highlighted some of the problems discussed in the following section.

Identified problems

I anticipated that students would have difficulties in adpating to this approach but I thought that their general interest in computing and their obvious ability to use computer systems would mean that these problems were not serious. In fact, the problems which were encountered were more significant than anticipated. The key problems which arose were:

1. *Process problems* I did not give students any information about the process of using a conferencing system for learning. This was partly because the system was new to me but also because I anticipated that they would have some personal learning model which they would adapt to the system. From the questionnaire, I found that students mostly did not have any structured approach to learning but were opportunistic and event-driven e.g. the need to finish an assignment meant they were forced to learn. They found it very difficult to adapt to the more structured approach which was required and, in essence, they were unwilling to spend the time required to develop a useable learning process.
2. *Access problems* I was unhappy that many students did not make everyday use of computer-mediated communication systems as they preferred to work on their personal computers. Our student rooms do not have network access so these were invariably self-contained. I had hoped that adopting this method of learning would make them more familiar with network-based communications. In fact, the students with personal machines were very unwilling indeed to work in a Unix rather than a PC or Macintosh environment. Their previous experiences of Unix were not positive (they had only used it fo assignments) so they resented being

forced to use an unfriendly system. The class divided roughly 50:50 in this respect.

3. *Coursework management* The course ran concurrently with a number of other courses which set assessed coursework. The students managed their time to give priority to this assessed work and neglected the self-assessment exercises which had been set. By the end of the course, out of 70 students, only 2 or 3 had attempted some exercises from every assignment. None had tried all exercises (this was expected) and about 30% had chosen not to do any self-assessment exercises at all.

Conclusions

In the questionnaire students were evenly divided in their opinions about the course. Some students liked the approach and much preferred the flexibility that it offered. Others found it difficult to adapt and to manage their time and said they preferred conventional lectures as these 'forced' them to work on the subject.

I intend to run the course in the same way next year but I will make a number of changes to address some of the students' problems.

1. I will probably change the conferencing system to the local bulletin board system. The advantage of this is that students already have experience of the system so there may be less user hostility to the unfamiliar interface. Although the structuring and reporting facilities are not so good, these problems can possibly be overcome.
2. I will pay much more attention to teaching students about the process of using the system. I think that this is perhaps the most important lesson learned. Working out the most effective way to use an unfamiliar piece of software is difficult and it was unrealistic of me to expect students with little experience to derive an effective process. I am convinced that an effective process is problem driven and I will suggest this to them.
3. Rather than simply set 4 or 5 self-assessment problems, I will identify the key problems for the students that can drive their learning process. In essence, I will place less emphasis on self-assessment problems which implies that you use the problem to find out how much you know. Rather, the problems will be set to facilitate learning.

Appendix 1 - Introductory material

Introduction

This course is a continuation of the 2nd year course on software engineering. This year, for the first time, it is being presented in a radically different way. Rather than the conventional 'chalk and talk' lectures, the course will be based on electronic communications between lecturer and students.

There are several reasons why this approach is being adopted:

1. Conventional lectures are based on passive rather than active learning. They are not particularly effective as a means of instruction. My intention is to move from

a passive to an active model of learning where students take more responsibility for their own learning experience.

2. Conventional lectures are centred around the notion that the most effective communication between student and teacher is face-to-face communication which happens at regular fixed intervals. Before the advent of electronic communications, there was clearly no alternative to this but we now have computer-mediated communication systems which allow us to explore other alternatives which are asynchronous rather than synchronous.
3. Part of your general educational experience is learning to use computer-mediated communication systems. This will give you practical experience in the use of these systems.

Apart from these worthy educational reasons, I also recognise that conventional lectures are generally pretty boring for both the teacher and the student. Whilst an individual lecture can be inspiring or entertaining, all lectures can't be stimulating particularly when factual material is being presented.

Passive and active learning

The conventional 'chalk and talk' approach to lecturing developed when universities expanded to such an extent that individual tutorial teaching became impossible. At that time, the availability of educational materials was very limited (there was no notion of an undergraduate textbook, no photocopiers, limited publications) and the role of the lecturer was to distil 'wisdom' from his books and papers and them impart that wisdom to students. Students took notes in lectures and used these notes as a basis for their revision and to guide what reading on a topic which was possible.

By and large, the system of lecturing remains much the same as it was in the 19th century. We now often use pre-prepared overhead project slides but apart from that, the conventional model of a lecture hasn't really changed.

However, student's expectations have changed. They expect lectures to be backed up with additional material such as a textbook and or handouts. They expect copies of slides. They don't want to make notes (quite rightly in my opinion) as they recognise the difficulties of revising from notes.

Consequently, student participation in lectures tends to be completely passive. They come along, half-listen to a lecture but don't participate in it. Lectures are seen as a way of defining what must be learned rather than a learning experience. The time spent in lectures is largely wasted.

The key objective of this alternative approach to teaching can simply be stated as:

Students MUST take responsibility for their own learning

This means that each student must decide how to structure their time to ensure that they learn the material which is deemed to be essential for a course. Students will have direct access to the 'wisdom' in educational materials rather than having this 'wisdom' filtered through the lecturer. The role of the lecturer changes from a mouthpiece to an enabler who provides help and guidance plus a means of self-assessment for students.

Computer-mediated communication

Of course, the notion of active learning is not a new one and the tutorial system based around small group teaching is an active learning experience. In most universities, however, the increase in student numbers has meant that small group teaching is a thing of the past and is no longer possible given the limited staff resouces which are available.

Computer-mediated communication systems such as electronic mail and conferencing systems provide us with the opportunity to regain some of the personal communications which are inherent in small group teaching whilst freeing us from synchronous interaction. That is, we don't all have to be in the same place at the same time to communicate.

In this course, you will make use of two forms of computer-mediated communications:

1. *Electronic mail*. You are probably familiar with this. It supports user to user asynchronous communications. The same message can be sent to a group of users. There are various different electronic mail systems available on the Unix systems and it doesn't much matter which one you use. You should normally try and contact me using E-mail (is@comp.lancs)

   The most widely used systems are *elm* and *mail* and you should make sure that you are familiar with one of these systems. If you haven't used an E-mail system before, you can find out about either of these by issuing a 'man elm' or 'man mail' command when logged on to Unix.

2. *Computer conferencing*. This is a development of electronic mail systems. The key difference is that you don't just see your communications with someone, you see everyone else's. This is a great advantage in teaching. If you have a question, it is likely that other people have the same problem. You can see their questions and the responses in the conference.

   The conferencing system which you will use is called caucus which is available on the central university computer (cent1). There is a separate document describing how to get started with caucus. I will log on to caucus at least once per day when I am in Lancaster.

You may also wish to make use of another program called a Postscript pre-viewer which will allow you to view document images on a screen. Therefore, rather than slides being displayed on an overhead projector, you can view these slides directly on your workstation. The name of this Postscript viewing system is ghostview. Using ghostview to look at slides is described on a separate sheet.

Of course, the existence of these systems does not mean that you can't come and talk to me directly or by phone (x3795). However, students in the past have sometimes found it difficult to find me as other research committments means that I am often out of my office. One reason for introducing this new system is to address this communication problem. If you want to see me personally, it is always worth E-mailing me to fix a time.

Appendix 2 - A conference entry

Viewpoint-oriented System Modelling

Core Reading

You should read Chapter 4, 'System Modelling' of 'Software Engineering', 4th edition. You can skip section 4.3 on data modelling as I assume that you have already covered this in a database course. Note, however, that you may be asked in an exam to develop a data model of something.

Further Reading

There is not a great deal of accessible further reading which covers viewpoints and system modelling. A good survey paper which addresses viewpoints in requirements definition is:

'Viewpoints for Requirements Definition'. G. Kotonya and I. Sommerville, IEE/BCS Software Engineering J., 7 (6), November 1992, 375-387.

This is available in the library.

Self-assessment

1. You should do Exercises 4.5 and 4.6 on Pages 83/84 of 'Software Engineering'.

2. (i) Explain why it is important to carry out a viewpoint analysis when deriving the requirements for a complex computer system. (2 marks)

ii) An automated ticket issuing machine allows train travellers to buy their train tickets by selecting their destination and then inputting coins greater than or equal to the value of the ticket. The machine keeps track of coins received and depending on the number of available coins of various denominations, it can either give a customer change or demand that the customer inputs the exact amount to pay for the ticket. If customers have problems, they may request human assistance by pressing a button on the machine.

The machine display always tell the customer what action to carry out after a current action has been completed. A separate display shows whether or not the machine is in 'Exact fare' or 'Change available' mode.

Using CORE viewpoint diagrams, identify viewpoints which must be taken into account in deriving the requirements for such a system. (5 marks)

(iii) From the viewpoint of a customer buying a ticket, construct a tabular collection diagram showing inputs, outputs and actions. (5 marks)

(iv) Using CORE action diagrams, model the processing required of the ticket issuing system. For simplicity, you may ignore the handling of customer problems and routine machine

maintenance involving filling and emptying the machine.
(8 marks)

(This is an example of an exam question)