ITALICS

Innovations in Teaching And Learning in Information and Computer Sciences

The e-Journal of the LTSN-ICS

Electronic Support for Computing Students at a Distance

Pete Thomas
The Open University
Milton Keynes, MK7 6AA

Email:p.g.thomas@open.ac.uk

ABSTRACT

This paper discusses the range of computer mediated support provided to students on a third-level Computing course studied at a distance. An attempt has been made to support all tutorial activities electronically, including the submission and marking of assignments, tutorials, self-help and web support, with some success. The paper reports on experiences in the first year of presentation to over 2300 students and discusses the successful components and the changes that have been made following feedback from students and their tutors.

1. INTRODUCTION

M301, Software Systems and their Development, is a new, 60-point, third-level course in Computer Science. It provides students with the skills and knowledge relating to a broad spectrum of the computing curriculum. The course is primarily about the development of large software systems for modern applications. Since many current applications are distributed, we have concentrated on the production of concurrent systems. As well as teaching the usual topics in software development of analysis, design, implementation and testing, the course also discusses ethical and professional issues.There are three main topic areas in the course: Java, Concurrency and Software Development, and each one is supported by a text book [1], [2] and [3].We assume that students who take this course will have previously studied object-oriented programming ‘in the small’ and will have a basic understanding of the fundamentals of hardware and operating systems. To help students assess whether or not they have the pre-requisite knowledge we have provided a ‘pre-test’ – a web-based self test linked to a glossary – that asks questions about basic ideas of hardware and object-oriented programming and provides answers based on references to the glossary and published texts. It is left to the student to determine whether or not they feel confident to tackle the new course.Whilst the bulk of the subject matter is contained in the set texts, the course teaches primarily through examples, self-assessment questions, exercises and practical activities, which are provided as part of the course materials. However, some additional material is provided via the course web-site [9]. This is either teaching text supplied by the course team or links to other web sites. However, the web site plays a much more important role in the course than simply providing additional source materials as we discuss below. It is not an objective of the course to produce highly skilled Java programmers, however, we do expect students to gain a reading knowledge of the language and to be able to amend and extend existing code. Indeed, it is our belief that since the majority of software development occurs as part of the maintenance of existing systems, students should experience this environment. Therefore, much of the practical work asks students to interact with code provided by the course team.In this paper we shall describe the teaching strategy adopted on the course, concentrating on the support given to students whilst studying the course materials. We shall also describe where the strategy succeeded and where we had to make modifications in the light of feedback from students and tutors. The course was presented for the first time in 2000 to over 2300 students who were supported by a group of 110 part-time associate lecturers (more commonly referred to as tutors). At regular intervals throughout the course we obtained feedback from both tutors and students via questionnaires developed by the Open University (OU)'s institutional survey body. Less formal feedback was obtained via the course's computer conferences.

2. ASSESSMENT

In the OU teaching system, learning occurs primarily as the result of attempting a number of tutor-marked assignments

(TMAs) spaced throughout the course. Students are given feedback on their work from a personal tutor. In the first presentation of M301, there were 6 assignments, one for each block of the course. The assignments are designed to test all the major concepts in the course and are therefore comprehensive and very demanding on student time. There is a final examination that is used in conjunction with the tutor-marked assignments to determine a student's final grade.The Open University has engaged in several pilot projects that have developed an electronic assignment submission and marking (e-TMA) system [4] and [5]. The system, illustrated in Figure 1, is now fully operational and during the year 2000 over 60,000 assignments from students on 46 courses were processed [7]. This figure is expected to rise sharply during 2001 as more courses use the system.Students prepare their assignment as a set of one or more electronic documents that they compress and submit to the OU via the e-TMA web-site. In return, they are sent a receipt - an email message confirming that the assignment has been successfully received and containing a unique identifier that can be used in the event of any subsequent query. Students can also use the e_TMA web site to track the progress of the marking process.

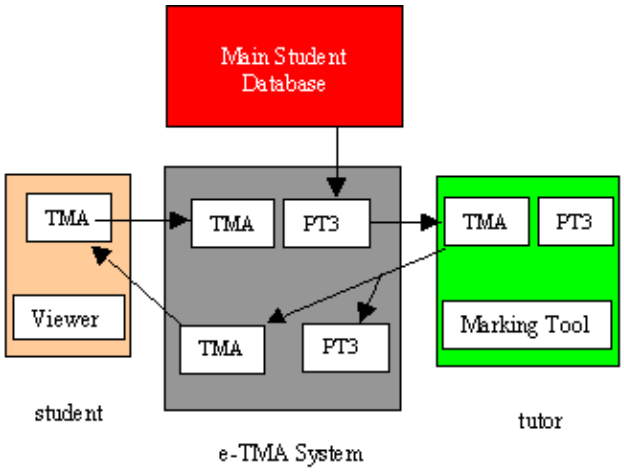

Figure 1 The e-TMA system.

A parameter file (known as a PT3) is added to each incoming set of TMA documents so that, when tutors access the e-TMA web site to download scripts, they obtain both the student's work and a variety of administrative information taken from the OU's main student records database. It is important to recognise that tutors, like students, work remotely and the e-TMA web site is the interface for both groups to the assignment handling system.Tutors use a specially designed marking tool that helps them mark on-screen, details of which can be found in [7]. Having completed the marking of a student's TMA, the tutor uploads the script and it's associated PT3 (which now contains a record of the marks awarded and the comments made) to the e-TMA web site. The marked TMA is then available for download by the student. The marking tool converts the student's script into a Microsoft Word document which can be read by the student using Word Viewer (provided by the OU) if the student does not posses Microsoft Word.The University keeps a copy of each submitted (unmarked) TMA and a copy of each marked TMA for auditing purposes. Once a marked TMA is in the e-TMA system an authorized monitor will examine the tutor's work for its accuracy of marking and appropriateness of comments which may become the basis of tutor development.At the end of the course, students sit a conventional three-hour invigilated examination and their final grade for the course is a weighted average of their continuous assessment and examination marks.The e-TMA system has been in use for over five years and on other courses has improved both the student and tutor experience. Students prefer electronic submission because it is more convenient, robust, accurate, and faster than conventional paper via the postal service system. Students and tutors like the system because they can each read the others' writings! Tutors' reactions to the system tend to be polarised either positively liking or definitely disliking the system, with the vast majority being in favour. Generally, tutors come to the system with reservations about on-screen marking, but once they appreciate that a different approach is required, soon develop their own methods for doing the job. Our evidence suggests that on-screen marking for experienced practitioners takes no longer than paper-based marking and indeed, the feedback given to students has improved [4,5]. The success of eTMAs has encouraged other course teams to adopt the system, either for the course as a whole or as an option for students.

3. TUITION

The role of the tutor is not solely to mark and provide feedback on the assignments. Tutors also provide direct support to individuals and groups. Each tutor looks after about 25 students and is responsible for (a) answering individual queries and (b) providing tutorials to the group. In M301, individual tuition occurs via email or telephone, and group tuition via computer mediated conferences (CMC) and face-to-face tutorials. Experience on other Computing courses has shown that students appreciate a variety of support mechanisms and that taste dictates which form(s) of support an individual will use. Therefore, all forms of support are offered to all students but there is no compulsion to use any. There is historical evidence that, where support is offered on a voluntary basis, up to about a third of all students will actively participate no matter what medium is used. Interestingly, with CMC, up to about a third will post messages, about a third just read the messages on the conference ('lurk') and about a third do not make use of the provision at all [5]. In early experiments into the use of CMC in tuition [4,10] it was found that group size is significant for successful tutorials. That is,

the number of students assigned to a group had to be over a certain 'critical mass' if successful student interaction was to be achieved. Certainly, the critical mass is greater than 25 so it was decided to encourage tutors to combine their student groups to make larger cohorts. In the event, as we report later, attempts to run electronic versions of traditional face-to-face tutorials were poorly attended.

4. THE WEB SITE

The web site has several functions:

- to provide a pacing mechanism by showing the study scheduled for that week;
- to provide access to additional study materials including links to other sites;
- to give access to tutor-marked assignments (TMAs);
- to keep the course up to date by issuing news, clarifications and errata.

Figure 2, below, shows the opening page of the M301 web site which is always related to the current week's work. It shows the information for week 13 (the course is studied over 32 elapsed weeks).

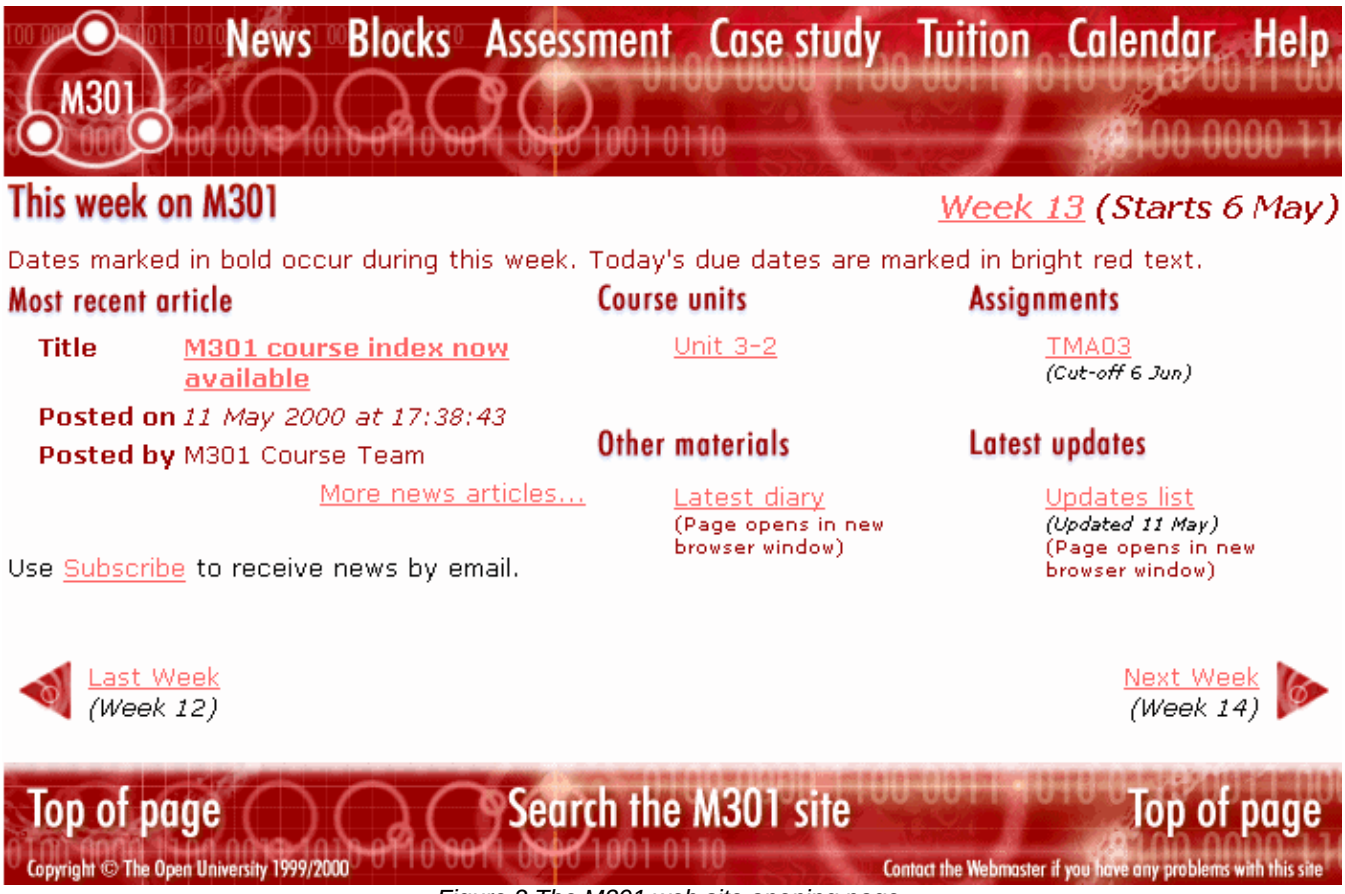

Figure 2 The M301 web site opening page.

The top banner is a menu that contains hot-links to other significant pages of the site. The *Blocks* link gives access to information about the course materials associated with each block (there are six blocks in all); *Assessment* leads to a page from which assignments can be downloaded; *Case Study* gives access to the materials describing a significant case study; and *Help* gives contact information about where additional support can be obtained (the OU provides a range of student support services including a Helpdesk that deals with queries relating to hardware and software problems and operates seven days per week at peak periods). The *Calendar* hot-link accesses a year-long diary that specifies the study and associated materials for each week (unit) of the course. It is the information in the calendar that drives the web site and dictates what appears on the opening page. Much of the information on the opening page relates to new materials added to the course during presentation, a facility not normally available in paper-based courses. An extract of the *Calendar* is shown in Figure 3.

Block 1 - Introduction to Java

Starts 03 Feb 2001 (Week 1), ends 16 Mar 2001 (Week 6)

Course units		Assignments	Cut-off date
Unit 1.1	*Introduction*	TMA 1	*11 Feb 2001*
Unit 1.2	*Basic constructs in Java*	TMA 2	*25 Mar 2001*
Unit 1.3	*Using inheritance*		
Unit 1.4	*Inheritance, composition and polymorphism*		
Unit 1.5	*Swing, AWT and applets*		

Figure 3 An extract of the course calendar.

The *News* page lists all news articles, (the most recent of which is also referenced on the opening page: Figure 2 indicates that a new questionnaire is available). The aim has been to ensure that relevant information can be accessed quickly, with the number of hyper-links to be followed minimized. Figure 2 also shows an important feature of our approach. To avoid students having to log into the site at regular intervals just to see whether or not there have been additions to the site (an action that could be costly in terms of telephone use and time, especially when using slow modems), a subscription mechanism has been added. Whenever the site changes, an email message is sent to all students who have opted for the service. The email message not only indicates that a change has taken place but also indicates the nature of the change and gives a hyper-link to the relevant page(s).

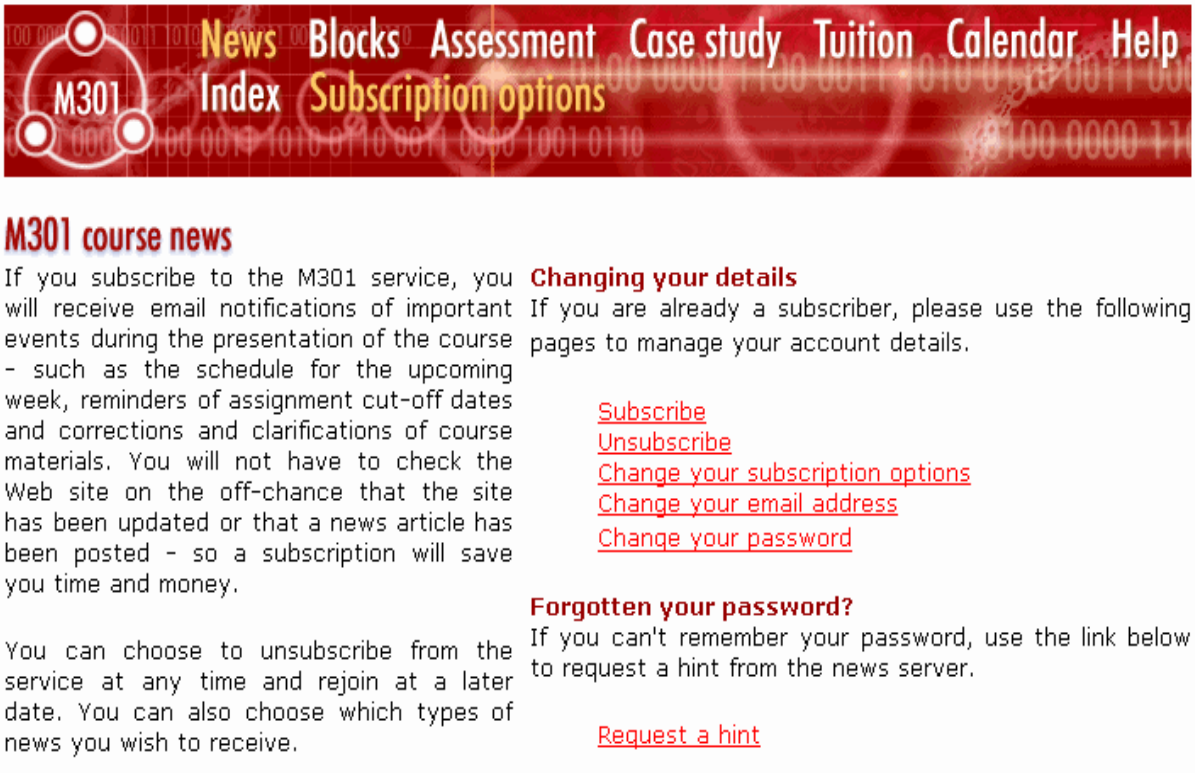

Figure 4 The course News page advising of the subscription facility.

The *Block* link gives access to additional materials. Figure 5 illustrates part of the contents of a typical page related to a course unit.The course is divided into 30 study units, each one of which is studied over a one-week period. The study of the unit typically involves reading part of a set book and examining other materials. Students are also asked to visit different web sites. In Unit 1.1, for example, they read part of the Killer Robot Case, a fictional account about the problems that arose when a computer controlled robot malfunctioned (an exercise primarily designed for the study of ethical issues). The page for Unit 1.1 also instructs the students to read the first of 17 parts of the Case Study describing the development of a major software project.In addition to new materials, a unit page also gives the aims and objectives of the unit and any errata and clarifications that might have arisen. The pages in the web site are organised hierarchically in a small number of levels and the banner menu expands as the user traverses down the hierarchy as illustrated in Figure 6.

Unit 1.1

This unit is studied in Week 1 Week 2

Introduction and the IDE

On this Page

Review
Aims and objectives

Related Pages

Killer Robot case
(Page opens in new browser window)
Case study *(Part 1)*

Set book readings

Budd, Chapters 2 and 4

Aims and objectives

The aims of this unit are to:

give an outline of the whole course and describe the purpose of each block;

introduce some of the important issues in software development;

introduce you to the facilities of the Java Integrated Development Environment (hereafter referred to as the IDE);

examine some of the basic constructs of the Java programming language.

Previous Unit

Next Unit (1.2)
Basic Constructs in Java

Figure 5 The web page for Unit 1.1 (part).

Figure 6 The web site menu bar.

The assignments must be downloaded from the web-site (see Figure 7). Since TMAs are the backbone of the course and are only available from the web site, we take particular care to ensure that the downloading process is as trouble free as possible, and that materials are provided in as usable form as possible. Therefore, we provide the questions in pdf format (Adobe Acrobat is part of the course software) and Microsoft Word. There is also a template (provided in rich text format) which students can use to enter their solutions. This scheme worked well: none of 2300 students failed to download their assignments due to a software problem with the site. In addition, the template helped to ensure that essential information was not omitted from a student's script and ensured that tutors received similarly formatted assignments which made their marking task easier.

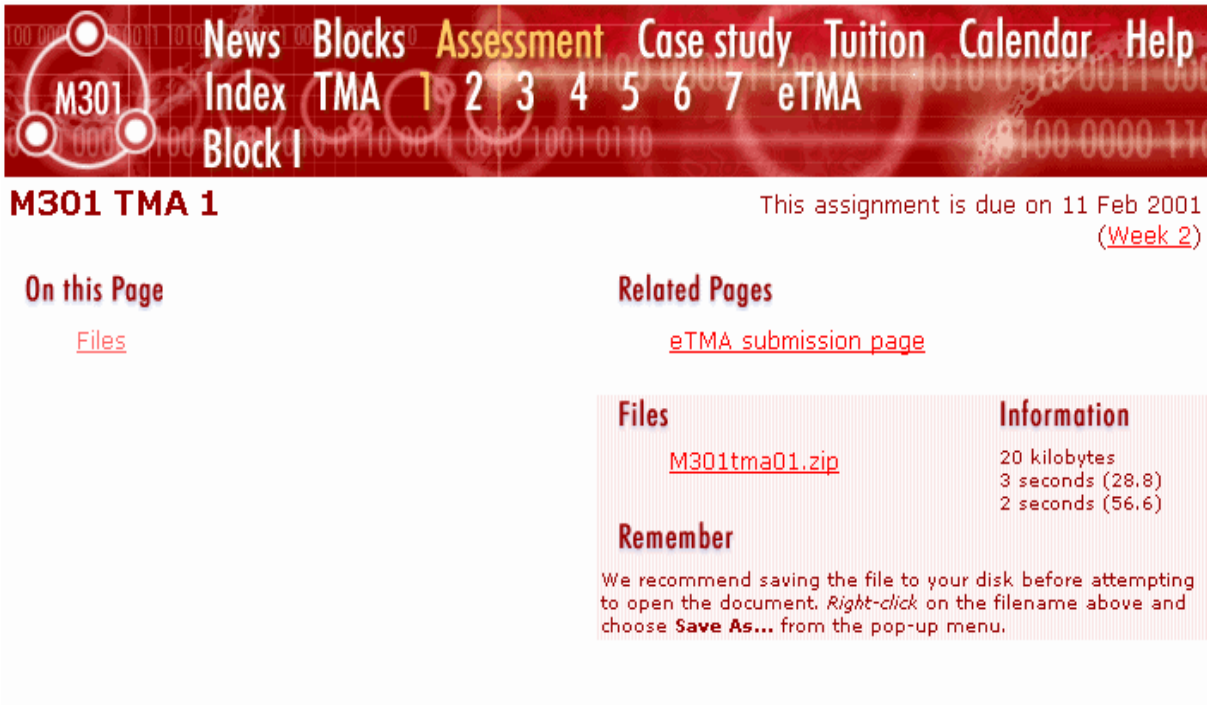

Figure7 The Assessment page.

An interesting feature of electronic assignments, is knowing whether or not a student has deliberately failed to make an attempt at a question or has simply forgotten to include (part of) a solution. In a paper-based system, it is often possible to deduce, for example, that a page has been omitted by mistake; there are fewer clues in an electronic document. The template can help in this respect, but we also ask students to state explicitly when they decide not to attempt (part of) a question. In general, a TMA will ask the student to submit several files as their solution to a TMA, one of which must be a word processed document containing the textual and graphical elements of their solutions and is referred to as the *Solution Document*. Other files will typically include Java program code, screen dumps from running programs, and UML model diagrams. However, we soon discovered that it is very easy to ask for a multitude of files in solutions, and that a complete answer to a TMA could grow alarmingly large. This caused a great number of problems resulting from the length of time to upload and download files, particularly when slow speed modems were in use. Therefore, we have begun to be more selective in what we ask to be submitted, only asking for additional files when we want tutors to execute programs or examine UML diagrams with the course's modelling tool as part of their marking task.There was an interesting effect of the use of the web site for keeping students up to date with changes. We thought that it would be beneficial to provide information about errors and clarifications as soon as they were notified to us. This was not the case. That we did not distinguish carefully enough between errors and clarifications, and issued reports on an almost daily basis at the start of the course was not appreciated. Students gained the impression that the course was riddled with errors when in fact the number was comparable to other courses in their first year of presentation. They also felt that they had to visit the web site on a too-frequent a basis to keep abreast of changes. We responded to the complaints by (a) issuing news items on a weekly basis and (b) redesigning part of the web site to separate the errata from clarifications in such a way that access to the information was clearer. The issue of releasing new information via the web site has been quite contentious. Each OU course has a calendar that sets out precisely when students should be studying particular materials and when they should submit their assignments. Whilst we know that for the majority of students the cut-off dates for assignments is the major pacing mechanism and we can target the issue of new material to match, we can never be sure that we are meeting the needs of all students. At one time or another, every student will wish to ‘get ahead’ with their studies (perhaps because they have other commitments, such as work or holidays, that they need to attend to at the time an assignment is due). Consequently, issuing material after a student has completed an assignment causes a great deal of frustration and anger. Our reaction has been to reduce the amount of new material issued during the course and, whenever new information is issued, to do it as far in advance as possible. However, errors normally only come to light when students are studying the materials which means that we can only report the errors after a number of students have studied the material: a *Catch-22* situation. Of course, we aim to minimize errors, but we are reducing our use of the web site in precisely the area for which the technology is ideal!There are a number of novel features of the M301 web site design and these are described in [12].

5. COMPUTER CONFERENCES

5.1 Tutor conferences Prior to the start of the course a computer conference was set up for a small group of tutors (see Section 6) to enable them to develop support material collaboratively for the cohort of tutors on the course. Subsequently, as tutors were appointed they were added to the conference and the structure of the conference was expanded to cover different aspects of the tutors' role. By the start of the course all 110 tutors were on the conference, and after about a month of operation, over 80% of tutors had contributed and all had accessed the messages.Following consultation with tutors we set up a comprehensive hierarchical structure of sub-conferences. In the event, once the course started and

issues began to be raised, it soon became clear that the conference structure was too complex. Tutors complained that they could not find appropriate information and it was never entirely clear where a particular issue should be discussed. The result was that we re-designed the sub-conferences into a flat structure with just 5 sub-conferences covering: notices from the course team, assignments and tutorials, technical issues, feedback and chat. By the end of the course, the use of the notices and feedback sub-conference had diminished significantly indicating that five sub-conferences was too many.Our experience has been that during the early part of the course the use of the tutor conference was extremely high with tutors discussing a wide range of topics. Of particular interest to the course team were the discussions of the course materials and assignment questions, which provided valuable feedback either in terms of errata or the need for clarifications of the course materials. A surprising feature of the tutor conferences was the extent to which tutors were willing to ask questions relating to their understanding of the course subject matter, and the willingness of other tutors to answer the questions. The tutor conference has been used as a self-help mechanism. The value of the conference has, therefore, been very high in terms of tutor development, feedback to the course team and the reduction in the load on the course team in answering individual tutor queries. At our end-of-course tutor debriefing conference (face-to-face!) tutors reported how useful the tutor conferences had been in helping them to support each other. **5.2 Student conferences**Computer mediated conferences for students are an integral part of the course tuition strategy. This strategy was based on the normal tuition mechanisms provided by the University. A single tutor normally looks after a single group of approximately 25 students. On a typical OU course, students in each tutor group look to their tutor for marking assignments, answering individual queries and conducting a small number of face-to-face tutorials. On M301 we added to this structure by expecting tutors to provide tuition to their group via computer conferences. Therefore, we set up a student conference structure in which each tutor had a small number of sub-conferences for dealing with general queries, tutorials and chat. Research in this area [4] has shown that a student population of 25 is insufficient to generate much interest and that for useful student interaction to occur a much higher ‘critical mass’ is required. Therefore, we encouraged tutors to combine their groups, thereby increasing the student population and giving students access to more than one tutor. This approach effectively provides students with greater access to tutors. However, there is a limit to the number of participants in a conference that can be properly supervised. Moderating and answering questions on a large conference can become overwhelming. Therefore, we encouraged the setting up of regional conferences (of up to 200 students each) for chat purposes and reserved the smaller tutor-group conferences for academic and course related discussion.In the event, these structures have only been partially successful. It is generally the case that only a small proportion of students will actively participate in a conference (up to about a third) although a larger proportion (up to two-thirds) will lurk [4]. However, to gain these levels of activity requires a great deal of encouragement from tutors: if students feel that the conference is not directly serving their purpose, they will not participate. Conversely, there is a large minority of students who look to computer conferences as a major source of help: the ability to discuss a range of issues with other students is seen as a vital support mechanism in a distributed environment. The fact that many of our students had studied together in the previous year on a course that provided a nation-wide conference meant that students expected the same facility to be available on M301. That M301 provided a different structure was not seen to be beneficial by these students. The students decided to set up their own national conference that has somewhat detracted from the course team’s structures. This is a good example of raising expectations in one course that are not met in subsequent courses. It is interesting to note that the idea of a national student conference on the earlier course has now been dropped in favour of tutor and regionally based conferences, which should ensure that our computing courses provide a uniform structure to students.The course team developed a set of asynchronous electronic tutorials for tutors to use in their small group conferences. Typically, an e-tutorial consisted of a small number of stages spread over a two-week period. In each stage, students were given a problem to solve (normally related to ideas discussed in a previous stage) and were asked to solve the problem through discussion. The tutor's role was to moderate the conference, providing help where necessary, and to ensure that the discussion kept to the point. To ensure that each stage could progress satisfactorily, tutors were asked to post a summary of the discussion at the end of each stage. Each e-tutorial was related to a specific aspect of the forthcoming assignment in order to encourage student participation. To minimize tutor effort, the course team provided a set of customisable messages to post to the conference relating to the structure of the e-tutorial and the individual stages. The design of the e-tutorials was based on earlier research [see 4 and 5, for example], which indicated that, for success, asynchronous tutorials had to run for at least 10 days in order to allow a sufficient number of students to participate and to allow the discussions to develop. We also knew that students would not, in general, participate unless they perceived the utility of the exercise. Despite the fact that each e-tutorial would help students answer their assignments, the number of students who participated was disappointingly low. Students were, in general, prepared to participate in the early e-tutorials but interest soon waned unless the tutor spent a great deal of time encouraging them. In questionnaires, students said that they often did not see the relevance of the e-tutorials, that the e-tutorials were not time-tabled appropriately and took up too much time. An analysis of these points showed that a discussion that lasts for more than a week is not helpful because each week the student moves on to new topics in the course and does not want to continue a discussion on previous work. It is also the case that students work at different speeds and some students felt that they did not have enough knowledge to contribute to or understand the topics under discussion all of which contributed to the feeling of irrelevance. Whilst we continue to believe that electronic tutorials are valuable for a minority of students, particularly those who cannot attend face-to-face tutorials, we have redesigned the exercises to be answered in a much shorter period and time-tabled them to fit the course calendar more closely. This will inevitably mean that fewer students will be able to participate, but we hope for a greater participation rate than before.

6. TUTOR TRAINING

Given the broad range of topics taught in the course and the significant reliance on electronic support mechanisms it was

recognized that it would be difficult to recruit tutors who were totally conversant with all aspects of the course. Therefore, we would have to engage in significant tutor training. It is usual, at the start of any course, to provide tutor briefing sessions. However, we felt that traditional approaches would not be sufficient.The first step was to appoint three 'super tutors' – experienced associate lecturers who reviewed our materials and provided specific training materials for all tutors. Secondly, we set up a tutor computer conference in advance of the course start date so that tutors could develop their own self-help group. Our experience has been that both activities have helped enormously: an examination of the extent to which the conference was used confirms this.For the majority of tutors, the idea of marking on-screen was quite novel and, to an extent, daunting. To overcome this problem a tutor training pack was produced. Tutors were given access to a training web-site where they could work through a set of prepared exercises designed to take them through the download-mark-upload cycle. Tutors could also act as students to see precisely what students were required to do so that they were better able to help students who got into difficulty.Feedback from tutors via questionnaires showed that tutors had little problem with the downloading and uploading process. However, the on-screen marking exercises were not found to be beneficial because the training assignment was not sufficiently similar to the assignments received from students. The result has been that we now provide example assignments for each course that uses the system. Typically, we use suitably edited assignments from previous presentations of the course.

7. FEEDBACK

In any new project, feedback from students and tutors is essential. The computer conferences have provided useful feedback on errors and those parts of the materials that require additional support. However, we also wish to collect opinion of the course as a whole. Therefore, we ask students to fill in a web based questionnaire after studying each block of the course. To ensure that students are made aware of the existence of the questionnaire we signal its existence via the web site and the assignment.The information we have gained from the various forms of feedback have led to updated materials being made available to tutors and students and are influencing our plans for subsequent presentations. For example, students have found the first block of the course to be too rushed. They have not had sufficient time to absorb all the material on Java programming particularly given that they have had to learn to use several new software packages including a Java IDE and the computer conferencing system (the OU uses the software called *FirstClass*). Therefore, we are modifying the structure of the first block of the course and its associated TMA to provide more time for student study. In particular, we are setting an initial assignment that addresses issues related to setting up the software for the course. This means that students will be forced into setting up their software right at the beginning of the course prior to starting their study of the substantive topics of the course. This should ensure that any problems relating to hardware or software systems should come to light early. Tutors will be able to help students overcome such problems before they interfere with normal studies. As mentioned earlier, the course depends heavily on the existence of three set books. Feedback from a substantial number of students and tutors has enabled us to identify problem areas in the teaching strategy adopted by the set books. We have been extremely fortunate in that the authors of the books have been eager to help us improve their teaching strategy. In the case of [2], for example, we have worked closely with the author to revise some of his materials (he has also had feedback from his own students which has also influenced the changes). The publishers of the book have been very supportive in this process, to the extent that a revised edition of the book has been published in time for the second presentation of the course (2001).

8. SUMMARY

We have tried to make M301 almost totally electronically supported. We have had some successes and some criticism that led us to rethink the design of some of the components.The main lessons learned have been:

- to manage student expectations better, by ensuring that students obtain consistent support and approach across a range of courses;
- to provide systems that students perceive to be directly relevant to their needs;
- to reduce the extent to which new materials are provided to students during the course;
- to provide appropriate training to tutors in the new electronic features used in the course.

We have attempted to use a range of electronically supported pedagogic devices with varying success. Electronic submission of assignments is extremely popular with students, and tutors are becoming better able to support students via this medium through appropriate training. The web site enables us to react quickly to student needs such as reporting errors and giving additional support materials. However, we have discovered that this facility has to be used with discretion otherwise it becomes a source of frustration and dissatisfaction in students. Computer conferences used for self-help are extremely popular and effective with both students and tutors. They provide a quick and effective mechanism for solving immediate problems (i.e. problems that their users want solving). Setting problems as exercises in electronic tutorials are much less popular.One measure of the success of this approach is the course retention rate: the proportion of students who complete the course. In a distance education environment in which the students are adults studying part-time, students have to withdraw for a wide variety of reasons. The primary reasons are reported by students to be pressure of work and family commitments. Undoubtedly, some students withdraw because they find that the course is too demanding or does not meet their needs. In the event, in its first year of presentation, M301 has achieved the highest retention rates of all our computing courses.In conclusion, we shall continue to support students electronically in as wide a range of ways as possible. However, we shall reduce the level of our expectations as to the extent to which all students

will use all the mechanisms available. Nevertheless, we have found that each support mechanism is of use to some students and helps to reduce the isolation experienced by many students studying art a distance and contributes to our desire to retain as many students as possible.

9. REFERENCES

Bacon, Jean, *Concurrent Systems,* Second Edition*,* Addison Wesley Longman, ISBN 0-201-17767-6 (1998).

Budd, Timothy, *Understanding Object-Oriented Programming with Java*, Updated edition for the Open University, Addison Wesley Longman, ISBN 0-536-62310-4 (2000).

Stevens, Perdita with Pooley, *Rob Using UML*, Updated Edition, Pearson Education Ltd, ISBN 0-201-64860-1 (2000).
Carswell, L., Thomas, P., Petre, M., Price, B., *Distance education via the Internet: a report on students' experiences*. British Journal of Educational Technology, **31** (1), 29-46 (2000).

Thomas, P., Carswell, L., Price, B., Petre, M., *A holistic approach to supporting distance learning using the Internet: transformation, not translation*. British Journal of Educational Technology, **29** (3), 149-161 (1998).

Thomas, P., Carswell, L., *Computer Science Education over the Internet: The Future?* in: *Computer Science Education in the Twenty-first Century*, A Greening (ed), Springer-Verlag, New York, ISBN 0-387-98881-5, pp215-262 (2000).
Thomas, Pete and Taylor, Dean, *Reducing the distance in Distance Education*, Proceedings of the 4th International Computer Assisted Assessment (CAA) Conference, Loughborough, June, (2000).

M301, *Software Systems and their Development*, A third-level course, The Open University (2001).
http://m301.open.ac.uk/

Mason, R and Bacsich, P, *Embedding computer conferencing into university teaching*, Computers and Education **30** (3-4), 249-258 (1998).

Thomas, Pete, Weedon, Ray, Hall, Pat and Armstrong, Steve, *Supporting Computing students at a distance*, Proceedings of the 8th Annual Conference on the Teaching of Computing, Edinburgh, LTSN Centre for Information and Computer Science, (2000)

Thomas, Pete and Richards, Mike*, Database Integration - Releasing the potential of educational web sites*, Proceedings of the International Conference on Information and Communication Technologies for Education (EDICT), Vienna, (2000).