



WebCT and Online Assessment: The best thing since SOAP?

Den Pain and Judy Le Heron

Institute of Information and Mathematical Sciences

Auckland Campus

Massey University

New Zealand

D.E.Pain@Massey.ac.nz

J.L.LeHeron@Massey.ac.nz

ABSTRACT

Cheating and increasing class sizes in Information Systems courses have forced us to reconsider our approach to assessment. Online assessment was introduced in our department in a pilot form in 1996 through an in-house developed package, SOAP (Student Online Assessment Program) and has evolved over subsequent years. This paper examines our explorations into WebCT and compares its Quiz tool with the facilities (potentially) offered by developing an in-house assessment package. We examine the differences between the two examples of learning and teaching technology mainly from a teacher's perspective but include some illuminating examples of feedback from our students. We conclude with some suggestions of factors that govern the successful use of online assessment in Information Systems courses.

Keywords: Online assessment, Cheating, Course design, Technology enhanced learning and teaching, Customised or package learning software

Introduction

Online assessment in our Information Systems courses, which started with an exploratory attempt in our first year course in 1996, is now incorporated as part of the assessment strategy in five courses from first to third year. The period 1996-2000 saw the evolution of the Student Online Assessment Program (SOAP) that was developed to automate tutorial support and assessment. SOAP was developed by Tony Richardson with the work on computer-based assessment partly supported by a grant to Tony Richardson and Judy Le Heron from the Fund for Innovation and Excellence in Teaching, Massey University. SOAP provided the ability to automate the testing of student programming skills using Structured Query Language (SQL) and analytical skills by providing a replica of Dataflow Diagrams (DFDs) and Entity Relationship Diagrams (ERDs) which could be labelled according to student analysis of a small case study (scenario). The use of SOAP dramatically curtailed student cheating and the automation of marking and recording of marks lifted a burden from staff. However, during 2001-2002 inevitably our use of SOAP was impacted by the winds of change. Firstly, the staff member who developed the SOAP software became involved in other courses and had limited time to adapt the software to reflect new assessment material (Boisot, 1998, Castells, 1996, Stewart, 1997). Secondly, a new Information Systems programme was introduced focusing on object-oriented concepts, which changed what particular courses required of the software. Thirdly, our university adopted WebCT as the preferred platform for offering courses online and staff were encouraged, although not compelled, to make use of the facilities provided by WebCT. This paper outlines our motivation

for using computer-based testing, our experience of moving from software developed in-house specifically for our courses to generic commercial assessment software. The benefits and limitations of each are summarised, the effectiveness of WebCT in combating cheating is outlined and the impacts of WebCT on staff are discussed. To trace the evolution of our assessment methods see Table 1.

Why computer-based assessment?

Firstly, there should be a harmony between the technological nature of the subject matter and some of the learning and assessment practices (Dowsing, 1999). The students are expected to be adept at using technology as part of their learning mechanisms, for example our courses have provided web access to course material, lecture notes, assignment specifications and administrative information for quite a few years. In addition, our students are required to gain some proficiency in industry software (Computer Aided Software Engineering (CASE) tools) through practice with a minimum of tutorial guidance and written material.

Secondly, technologically based objective assessment has enabled us to reduce the occurrence of cheating (Le Heron, 2001). Teaching staff and the vast majority of honest students find it very demotivating to see students achieve 'success' by cheating. Our experience has shown that if students submit work as a group or have the opportunity to develop work with others before handing in their 'own' work, there will be a small percentage who put their energy into 'beating the system' by cheating. Technology, through monitored (proctored) online tests can provide a significant barrier to the dishonest student. Computer-based testing enables students to be individually tested and features such as randomising of question displays, discourage students from looking at the next screen without the need for heavy handed policing. With SOAP we achieved this by using separate scenarios for different tests so adjacent screens displayed different questions. The use of technology does not of course eliminate the problem of cheating, it just changes some of the mechanisms. Some cheating methods like crib sheets can be used whether the test is computer-based or paper-based and tests have to be monitored in the same way. However, one of our students reported over-hearing, while waiting for a test session, that if you viewed the web page source-code you could see the test password. Although untrue, it does demonstrate that there are always students interested in knowing the answers without knowing their subject. Honest students, such as our informant, want to be assured that cheating is impossible.

Thirdly, educational technology is harnessed in an effort to maintain the quality of learning when student demand outstrips the supply of staff resources (Ward & Jenkins, 1992). The reduction in funding per student over recent years is acknowledged both in New Zealand and Britain (Friedlander & Kerns, 1998, Richardson, 2002). In disciplines like Information Systems where student demand is high but recruiting and retaining academic staff is difficult, this results in very high student-staff ratios. Although it is clear that automatic marking relieves a lot of the work associated with traditional report marking it still requires substantial effort and organisation. Basically, there is a time shift in the work required i.e. more effort is required early in a course to ensure the online questions are formulated, loaded, correctly devised and proof read. An experienced team running courses for the second time can accomplish this smoothly but with constant changes in material and technology (including versions of the learning/teaching package) even this situation can provide tense high-pressure periods early on in a course. Some of this burden may be relieved by collaboration between institutions, the swapping of question banks, the sharing of best practice but this again is rare because it requires not only good planning and continuity of staff but assumes an environment in which institutions collaborate rather than compete.

1996	1997	1998	1999	2000	2002
Paper-based assignment of Microsoft Office Suite skills marked in student's absence plus computer-based skills consistency test	Computer-based test of Microsoft Office Suite skills marked in student's presence with marks recorded manually	Computer-based test of Microsoft Office Suite skills marked in student's presence using SOAP's marking panel to record marks directly to the student database	Computer-based test of Microsoft Office Suite skills marked in student's presence using SOAP's marking panel to record marks directly to the student database <i>and</i> to test and automatically mark student's SQL programming skills	Computer-based test of Microsoft Office Suite skills marked in student's presence using SOAP's marking panel to record marks directly to the student database <i>and</i> to test and automatically mark student's SQL programming skills <i>and</i> conceptual, analytical and DFD & ERD modelling skills	Computer-based test of student's conceptual, analytical and UML modelling skills using WebCT to automatically mark and record marks directly to the student database <i>and</i> provide statistics on question, test and student performance

Table 1. Paper to SOAP to WebCT: An evolutionary timeline

From 'our' software to 'their' software

SOAP was our answer to a number of the problems we faced – rapidly increasing class sizes, slowly increasing staff numbers and an increase in student cheating. It allowed us to provide tutorial support, practice tests and supervised tests with automated marking and recording of results. Marking consistency improved, marking was completed quickly, the drudgery of entering results was a thing of the past and cheating was almost impossible under supervision. SQL testing worked smoothly presenting students with a random selection of questions from the question database and we were able to lay a marking panel on the screen over Microsoft Office applications to enable student test marks to be recorded directly in the marks database. While the software developer was on the teaching team using SOAP there were minimal problems. However, once the developer moved on to teach other courses the staff continuing to use SOAP were reliant on his time and goodwill to make changes to the software to reflect new approaches and to fix errors. And, because the software was not intended for commercial use, it was idiosyncratic. Entering new test questions and matching answers in the test database was painstaking, it was easy to make mistakes and hard to check. Entering a new DFD or ERD was difficult because the diagrams were graphics upon which text boxes and drop-down lists were placed. We were stuck with the original diagram structures as only the developer could make changes to the source code. All we could do was change the content of the drop-down lists that provided our answer options.

Therefore when WebCT was licensed by our institution we were keen to find whether its Quiz tool would give us the independence to introduce new material, particularly diagrams, while maintaining all the advantages to which we had become accustomed using SOAP. We went to training sessions, we heard the accolades, and we were told about the wonderful tools that we could make available to our students. The ability to run tests were alluded to, but the practicalities of actually creating a test were glossed over. And no wonder, it is not an intuitive package for the newcomer. Although WebCT provides a lot of online documentation it is not easy to navigate through or interpret. It wasn't until a departmental visitor with WebCT experience walked and talked us through the basics of test construction that we were

able to make any headway understanding the WebCT ‘modus operandi’. It was a steep learning curve. However, our previous success with SOAP gave us the motivation to persevere and to push the WebCT boundaries to reflect the complexities of assessing student understanding of analytical and modelling concepts in relation to business scenarios, Unified Modelling Language (UML) and object-oriented CASE tools. But as Figure 1 shows, WebCT’s Quiz tool is for our purposes focused primarily on the automated marking of analysis skills and concepts and cannot be adapted to the other situations where we used SOAP, namely running SQL queries or recording assessment of Microsoft Office skills. Nevertheless, WebCT offers website and student record management facilities not provided by SOAP, but it is not those features but where they overlap that are the focus of this paper.

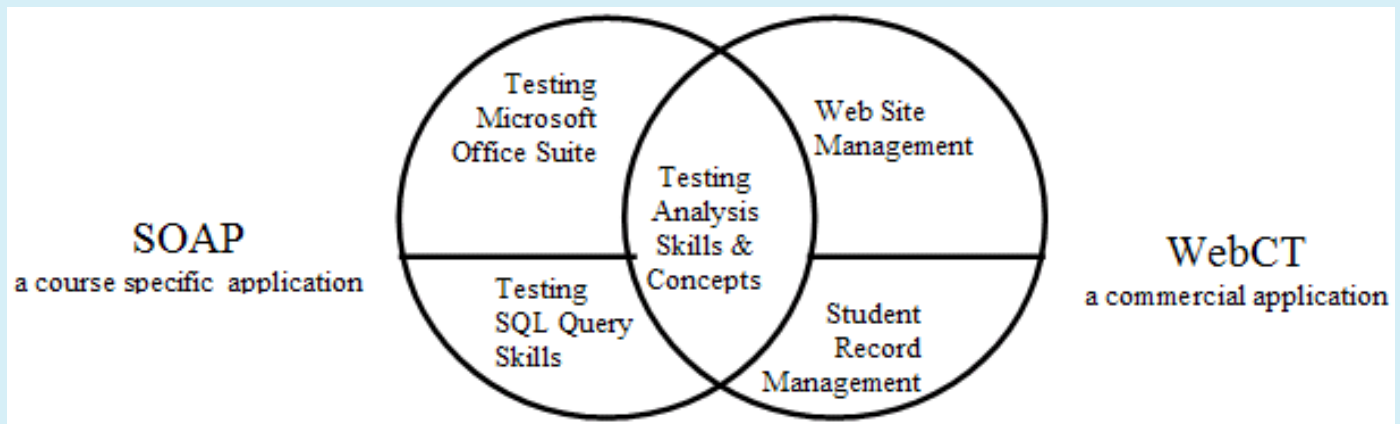


Figure 1. Comparison of WebCT and SOAP tools

Incorporating a test involves five main tasks: creating a question database; constructing a test from questions in the database; setting the specifications for test availability, security, delivery and release of marks; running a test and reviewing a test. The first of these is the most time consuming but all require attention to detail as explained below.

Creating the question database

While creating a set of test questions takes significant time it is only the first step. The second step, which takes as much time, is to convert them into one of WebCT’s question presentation formats. Although WebCT provides five types of question formats, only three are relevant to our subject area. ‘Short answer’ questions require students to enter their own answers while ‘multiple choice’ and ‘matching answer’ questions present the student with multiple answers (option buttons/check boxes and drop-down lists respectively) from which to choose. All three of these question formats have the option of providing general feedback to the student once the test is marked. In addition, the multiple-choice format can provide feedback specific to each of the options provided both correct and incorrect. However, each of the three also constrain the way the answers can be provided.

Constructing a test

Bureaucratically, this is the easiest of the five tasks and is carried out by selecting questions, or sets of questions, from the Question Database. However, it provides significant pedagogical challenges when several tests of equal difficulty have to

be constructed. Selection is easier if questions have been stored by category. Once the questions for the test have been chosen each is allocated a mark. When questions are created marking is specified as a percentage of the allocated mark for each sub-question or partially correct answer. This enables the decision about the value of each question to be determined when a test is created. If a set of questions is used each question in the set must be worth the same mark. From this set the designer can specify that some or all of the questions be randomly selected for each student who sits the test. Random selection, even if all the questions in the set are used, means that the questions are presented in a different order to each student who takes the test.

Setting test specifications and test security

Two important aspects of security are ensuring the integrity of the tests and maintaining the integrity of the recorded results. WebCT has two modes of access – designer access to create material for student use, and student access to material or quizzes made available for the class. If a designer wants to check the WebCT material from the student perspective then the designer is also allocated a ‘student’ usercode and password. Students are not aware of the existence of the question database and tests are not visible to them until the designer makes them visible. Even when a test is made visible to students their access can be restricted. Likewise, student access to their results can also be restricted until a specific date and time of release. Students only see their own results, which they cannot change, and the designer can determine whether they can also review their own test answers with or without the correct answers and feedback comments.

Test security is controlled by ‘Quiz Settings’ where test access is prescribed in terms of release date/time, availability to specific students, availability on specific computers and test password. WebCT also provides the ability to specify the style of question delivery, test duration, the form results are displayed in and the timing of the release of marks. It is necessary to specify the number of times the student can take the test. We limit this to once to reduce opportunities for cheating. The ‘selective release’ section allows staff to hide a test from the view of all students not specified on the ‘release to’ list. As part of our security measures, prior to the test, we limited release to a student ID used only by the teaching team. The ‘security’ section allows both an ‘IP address mask’ which restricts test availability to specific machines and a ‘proctor password’ which must be entered by/for each student before s/he can start the test. We used both these options firstly, to restrict test access to machines available in the university computer laboratories, and secondly, to restrict test access to test supervisors so that students could not gain access to the test at home or on university computers in unsupervised laboratories. The randomising feature of WebCT means that the questions are presented in a different order for each person who takes the test, which discourages those who try to copy from the person at the next computer. Finally, as an additional security measure we use different, but comparable, tests at each session to prevent students at later sessions finding out the questions from those who took it earlier.

Running a test

To minimise cheating we run our tests under supervision in university computer laboratories. As part of our test security we list tests for release to only one student user ID (known only by the teaching team) thereby hiding it from actual students. Just before the test is due to start it is made visible by removing the selective release criteria. Because we run multiple test sessions with different test versions over two evenings we only release the test versions we will be using immediately. We have found it is a good idea to check that the dates and times specified for the test to start and finish are accurate, and that we have remembered the proctor passwords for each test.

WebCT will give access to user IDs on the class list; it does not identify who enters that user ID so identification of each

student who takes the test is important. We carry out multiple checks. We register students on arrival for their test session, check that the student name on the WebCT screen matches the name on the student ID card during the test, and require each student to sign a class register at the conclusion of the test so we can compare signature and appearance with the student ID card. These measures appear to deter ‘stand-ins’.

Reviewing a test

Pedagogical problems such as incorrectly penalising unexpected but correct answers requires a diligent review of students results because even the most imaginative teacher will not come up with the same responses that the students are capable of. We included the statement that “Incorrect answers will be penalised” to discourage students from guessing True/False questions. However, we discovered later that a student for whom English is a second language thought this meant each choice of ‘False’ would be penalised, while a student for whom English is a first language thought it implied that an unanswered question would be penalised. If we discover a mistake in a test question after the test has been run we can change the question, the answer and/or the student marks to correct the situation. WebCT allows the changes to retroactively affect the completed tests and therefore student test results. Student grades can be adjusted by changing the mark allocation for a particular question for everyone or a specific student’s score for a single question or the test overall. As a result students, by and large, have accepted that the assessment has been fair, even when they have been disappointed in their own performance (Quirk, 1995). Statistics on individual student tests and class performance by question mean that not only can student performance be monitored but also that tests and their component questions can also be analysed and fine-tuned to ensure they are appropriate assessment instruments (Snow, 1989). Our in-house package, SOAP, had little in the way of features for reviewing marks. We could scale a test’s marks if there were problems during a particular test session but there were no records of an individual’s answers or statistics on particular questions.

Our first live test - held hostage by hyphens

Although we had ‘tested the test’ in the student environment beforehand we still encountered some unexpected problems which made our test unavailable for our first test session. Firstly, WebCT’s instructions suggested that to release the test to all students, all that was required was to ensure that nothing was entered in the ‘release to’ field. Unfortunately, this instruction is insufficient. In addition, the ‘release based on’ box should contain hyphens, which are inconspicuously located at the top of the drop-down list of identifiers. If the User ID remains selected WebCT attempts to find a User ID for the empty ‘release to’ list. Selecting the hyphens from the ‘release based on’ list made the test visible to the students.

Secondly, WebCT performance differed unpredictably by Web Browser. Despite the student computer laboratories all having identical software, WebCT was found to be more reliable using Netscape Navigator in some laboratories and more reliable using Internet Explorer in others. One problem was that although the IP address mask was set to allow access from any machine in any of computer laboratories, students using Internet Explorer on some of the machines were restricted from accessing the test. This was resolved by removing the IP address mask. A second problem was that the ‘proctor password’ was rejected as incorrect by some computers when using Internet Explorer but accepted by Netscape Navigator. The third problem was that Netscape Navigator shut out some students during the test requiring them to log on to WebCT again. This problem was exacerbated by the continuation of WebCT’s timer during the log off/log on period. Designers are alerted to this problem in WebCT’s documentation. However, there is no way to extend the length of the test for individual students or allow individuals to attempt the test more than once because test settings affect every computer. So we allocated them each a ‘dummy’ user log on and password to enable them to resit the test at another session and transferred their marks to their records manually. These can be regarded as ‘teething’ problems that have not recurred at subsequent test sessions but this type of difficulty will re-occur when there are changes to the assessment software and the

testing environment.

Some limitations of WebCT's Quiz tool

Although the amount of time required to become comfortable with the way the WebCT Quiz tool operates is significant, the main limitations of WebCT, summarised in Table 2 and explained below, relate to creating the question database and delivering tests.

Creating Questions	• Course assessment must be adapted to fit the question templates and marking criteria • Question types are primarily multi-choice • Cannot enter answers on diagrams • Creating a question database is time-consuming • Inadvertently penalising unexpected but correct answers
Running a Test	• Test delivery may differ by Browser • Student ID is not continuously displayed • Test timer is updated only on 'Save' • Test timer continues while student is logged off • WebCT can be minimised to get access to private files
Evaluating Tests	• Time required to analyse the statistical information

Table 2. Limitations of WebCT's Quiz tool

When you are creating questions the first limitation you become aware of is that course assessment must be adapted to fit the WebCT question templates and there is no opportunity to create new question types or test formats to suit the unique requirements of a course. While there are three main question formats with different presentation styles and marking criteria, test questions must be formulated very simply to facilitate automated marking. Complex questions must be broken down and presented as a series of more simple sub-questions. Although diagrams can be incorporated into the presentation of a question it is not possible for students to label elements of a diagram directly (to better replicate a CASE tool). Instead parts of a diagram must be labelled and identified by the student in the labelled answer section below the diagram. While WebCT provides for essay questions their marking cannot be automated. Multi-choice questions are the primary delivery mechanism. Presentation of answer choices can be either by a series of option buttons/check boxes (for a single question) or by one or more drop-down lists, for one or more related questions. The marking options, which differ for each question type, might also influence which of the question formats is chosen. How a question will be marked requires extra consideration in the 'short answer' format because students may type in a misspelling, an alternative spelling, a synonym, a different tense, a different part of speech, in a different case or with more spaces than the answer you envisaged. There are pedagogical problems that can be encountered in parallel, such as having too narrow a view of acceptable responses and therefore incorrectly penalising unexpected but correct answers.

All these decisions for each question being created means that a significant investment of time is required to build up the question database (Brown, Race & Bull, 1999). Although this investment of time pays dividends in the future the initial

impact is not inconsequential. Also, the nature of these decisions means that creating the database cannot just be handed over to a data entry person. It is possible to import questions into WebCT provided they have been formatted using WebCT's codes to indicate specific question display formats, answer options, marking alternatives and student feedback. Familiarity with the codes and layout required may lessen the time to create questions in the database but it also makes it more difficult to detect errors. However, checking is still necessary to ensure the imported questions are appropriate for your course.

The test delivery problems with different browsers, mentioned previously should not be an ongoing problem provided you could choose your browser. However, WebCT test delivery does leave a couple of loopholes for those determined to cheat. Firstly, the name of the student (which reflects the user ID entered) is only displayed at the top of each question and disappears as the student scrolls down through the question. Once the test is finished there is no way of checking whom the test was completed for. So test supervisors must site the name on the screen, to compare with the ID of the student taking the test, while the test is in progress. Secondly, it is possible to minimise the WebCT screen to get access to private files that may contain study notes. While vigilant supervision discourages this sort of activity, we make it more difficult by logging all computers onto a password secured network drive that gives no access to student files. So far we have had no incidents of students trying to view material on a floppy disk.

Some advantages of WebCT's Quiz tool

Although WebCT only provides one aspect of the functionality previously offered by SOAP, its Quiz tool has a number of advantages, as summarised in Table 3 and outlined in more detail below.

Creating Questions	• WebCT template for entering questions is easy to use once it is familiar • Question banks in WebCT format can be imported • Ability to incorporate an image into any/all questions • Ability to provide question specific feedback statements
Creating Tests	• Ability to control & customise of most aspects of test creation – presentation, content & marking • Ability to control & customise test security • Ability to provide unique tests • Ability to assign a different mark for a question when it is used in different tests
Running a Test	• Can control time taken • Can control number of attempts • Password protection • Control over which machines have access
Evaluating Tests	• Students appear to have greater confidence in a commercial application • A test can be automatically re-graded for the entire class if a mistake is discovered in a question • Ability to guarantee the security of test results • Ability to analyse the answering patterns for each question

Table 3. Advantages of WebCT's Quiz tool

A commercial application such as WebCT provides the user with a lot more control over the presentation, content, marking and security of automated assessment than a teacher adapting an in-house application in response to teaching pressures has time to incorporate. WebCT provides flexibility to introduce reasonable variety into what is essentially a 'multi-choice' approach not only in the way questions can be presented to students but also in the way automated marking can be customised for each question. The facility to incorporate an image, diagram or text passage into a question increases the ability to test the knowledge, comprehension, application and analytical skills of students, as educational objectives (Pritchett, 1999).

The WebCT test designer is able to incorporate a number of security measures to prevent unauthorised access to tests. These range from specifying precisely to which students the test will be made visible, when and for how long, the duration of the test, the number of attempts permitted and the computers on which it will be released. In addition, a 'proctor password' can be specified which an authorised person must enter on each computer before the student can access the test. WebCT's randomising facility not only presents questions in a different order for each new test begun but also alters the order of the answer choices on drop-down lists, which essentially provides a different test for every student. WebCT has the ability to provide feedback statements for each question, which enables the Quiz tool to be used for remedial use (Crooks, 1988). However, we have not yet had the time to invest in that refinement and felt it was important to concentrate on producing high quality tests first before we introduced additional functionality (Whittington, 1999).

Students certainly appear to have more confidence in a commercial software application than an application they know has been created in-house. Our experience of using both, a departmentally focussed, customised assessment tool, SOAP, and a university-wide (20,000 accounts) package, WebCT, would suggest that the wider familiarity and general significance of the latter gives it more acceptance within the student body than the less well known more specific SOAP software. This may be a function of (local) market dominance i.e. our students use WebCT for many of their courses and although they may be critical of its interface/response times etc. they do not perceive it as anything other than part of the university's infrastructure, over which they have little power. But even though some of the problems we encountered with WebCT were very visible to students, no one complained that the software was to blame for their poor performance. The WebCT log on restricts a student's access to their own tests and test results. They cannot access another student's record while they are logged on under their own user code and password so they cannot copy someone else's work or see their results.

SOAP on the other hand can be seen as more or less course specific and therefore more open to criticism and over which the students may (through their comments) have more control. They felt they could talk directly to the software developers and knew we were more amenable than the vendors of some large package that the university has licensed, especially as we invite criticism of any information system with which the students interact. When we were using SOAP there were always a few students who insisted they were disadvantaged by the software not working properly.

Impacts of computer-based testing

There must be institutional support for the students; just because computer-based testing exists doesn't mean it can be used successfully, they need guidance/practice in the particular technology used to assess them. We have found that although our students tend to be skilled computer users we noted considerable improvement in their acceptance of the technology and their scores when we increased the time devoted to guidance and practice directed at particular screens/test types used by SOAP (Race, 1995). This is easy to overlook, particularly when there are pressures on time and resources. Although some students will be able to take to a new format with relative ease and little preparation most will not. Given the stress associated with any assessment situation it is important that it is the content of the test that challenges them not the use of

the software. While the naïve view maybe that online learning is a low maintenance option for teaching staff, considerable work is required to prepare the learner for the format of the assessment or the result will be the under performance of both student and software.

One of the problems introduced by lab-based testing is not being able to test a class of 250 students at the same time. In our institution individual computer laboratories contain 25-45 machines and even four adjacent laboratories are insufficient to cater for the whole class. In addition, competing teaching and learning demands of other courses usually preclude the block booking of these facilities, especially during the day. Scheduling testing over several evenings provides students with some flexibility to work around their other commitments and simplifies the logistics. The major drawback is that some students pass on test information to others who attend later sessions, even though they are aware they are in open competition with one another, and we have found that the later test sessions are always booked up first.

The most straightforward solution to this obvious breach of security is to run different tests at different sessions. This poses two significant difficulties - that of fairness, producing tests of equal difficulty (Simms Williams, Maher, Spencer, Barry & Board, 1999), and that of workload, producing seven tests instead of one. Both of these difficulties can be reduced over time as question responses supplied by WebCT are analysed for level of difficulty and a question bank is built up. The ability to analyse question responses was not possible using SOAP. However, it should be noted that this analysis requires a considerable amount of effort and meticulous record keeping each time a course is offered for it to be useful. It also requires continuity of staff in the teaching team to avoid this pedagogical effort and knowledge being lost.

Conclusion

Despite having a course specific piece of assessment software, SOAP, at our disposal we have chosen to move to WebCT largely because of the independence it gives us as assessment designers. In addition given its status within the University it is respected and trusted by the students. We feel that the benefit of cheat resistant assessment has been retained and we have gained the significant features of review and remarking of tests.

Although we spent a lot of time learning the test ‘ropes’, exploring the idiosyncrasies of each question type, the results of the marking options, and the ramifications of the different quiz settings before we went ‘live’, we still encountered unanticipated problems in our initial tests. As well, a significant investment of time was required, not just for understanding and finding our way around WebCT, but also for actually creating a question database and tests. WebCT provides a plethora of statistics about student performance, on both an individual and class level, and about the assessment instruments, overall and in detail, but it requires time to analyse them and time to respond to them.

If the provision of high quality education is to be maintained there must be institutional support for educational software. This requires investment in WebCT support staff with the ability to assist teaching staff trying to come to grips with the complexities of a large application such as WebCT, and the technological expertise to overcome the inevitable problems that arise from the diverse university network platforms. The quality of this support is a significant factor in how comfortable the teacher will find working with a package such as WebCT and whether or not they continue to use some or all of its features. The provision of institutional support is problematic given universities find it difficult to recruit and retain good IT staff at current salary levels. Our own institution lost one of its university-wide WebCT experts at the beginning of the semester who had not been replaced by the end of semester. Some colleges within the university employ their own WebCT support staff while others do not. Even when good staff are in place and stay for a while it is difficult to balance the short-term trouble-shooting role with the longer term educational and development role. It is too much to expect the same people who are developing and running courses on the development of online teaching and learning to be on the end of a phone for emergencies during evening testing sessions.

The emphasis on preparation and review indicates that computer-based teaching, learning and assessment requires a lot of

effort to set up, review and run. There will be problems, no matter how good the software, hardware platforms and institutional support so what is important is preparation, damage limitation and recovery from difficulties and knowing who is responsible for these. We have had technical problems with both SOAP (the occasional unexplained crash and installation difficulties) and with our use of WebCT (browser differences and password problems). We have had problems ranging from seemingly impenetrable network difficulties to insufficient machines working. Whatever the cause, exotic or ordinary, the result is just as stressful for the student who is “ready to go” and the teacher trying to resolve the problem with a rescheduling nightmare threatening. The crucial issue, for both staff and students, is credibility – can the system be trusted, will it work as expected and can the information collected (test answers) and the information reported (results) be relied upon. The view that the university infrastructure, student management systems, computer networks and lab booking mechanisms are working well behind the scenes, should not be taken for granted by the teacher trying to estimate the effort required to deliver a high quality learning experience with online technology.

Acknowledgements

We are currently using version 3.1 of WebCT.

We would like to acknowledge the help and assistance of Dr. Max Burns, professor of Information Systems at Georgia Southern University USA, for getting us started on the WebCT trail. We would also like to acknowledge the help of InduSofat, tutor in Information Systems at Massey University Albany, for her help with the development of questions and course material associated with this paper.

References

Boisot, M. H. (1998). *Knowledge Assets*, New York: Oxford University Press.

Brown, S., Race, P., & Bull, J. (1999). *Computer-Assisted Assessment in Higher Education*, London: Kogan Page Ltd.

Castells, M. (1996). *The Rise of the Network Society*, Malden, Massachusetts: Blackwell Publishers Inc.

Crooks, T. (1988). *Assessing Student Performance*, Kensington, NSW: HERDSA.

Dowsing, R. D. (1999). The computer-assisted assessment of practical IT skills. In Brown, S., Race, P. & Bull, J. (Eds.) *Computer-Assisted Assessment in Higher Education*, London: Kogan Page Ltd., 131-138.

Friedlander, L. & Kerns, C. (1998). *Transforming the Large Lecture Course*, <http://learninglab.stanford.edu/pubs/friedkerns.html>.

Le Heron, J. (2001). Plagiarism, Academic Dishonesty or Just Plain Cheating: The Context and Countermeasures in Information Systems. *Australian Journal of Educational Technology*, 17 (3), 244-264.

Pritchett, N. (1999). Effective Question Design. In Brown, S., Race, P. & Bull, J. (Eds.) *Computer-Assisted Assessment in Higher Education*, London: Kogan Page Ltd., 29-37.

Quirk, B. (1995). *Communicating Change*, London: McGraw Hill.

Race, P. (1995). The Art of Assessing 1. *The New Academic*, Autumn, 3-6.

Richardson, T. S. (2002). The New Tertiary Model and its Low-Level Impact. *Paper presented at the InSITE Conference*, <http://ecommerce.lebow.drexel.edu/eli/2002Proceedings/papers/Richa109Infor.pdf>.

Simms Williams, J. H., Maher, J., Spencer D., Barry M. D. J., & Board, E. (1999). Automatic test generation from a database. In Brown, S., Race, P. & Bull, J. (Eds.) *Computer-Assisted Assessment in Higher Education*, London: Kogan Page Ltd., 71-84.

Snow, R. E. (1989). Toward Assessment of Cognitive and Conative Structures in Learning. *Educational Researcher*, 18 (9), 8-14.

Stewart, T. A. (1997). *Intellectual Capital The New Wealth of Organisation*, New York: Doubleday/Currency.

Ward, A., & Jenkins, A. (1992). The problems of learning and teaching in large classes. In Gibbs, G. & Jenkins, A. (Eds.) *Teaching Large Classes in Higher Education*, London: Kogan Page Ltd., 23-36.

Whittington, D. (1999). Technical and security issues. In Brown, S., Race, P. & Bull, J. (Eds.) *Computer-Assisted Assessment in Higher Education*, London: Kogan Page Ltd., 22-27.



Copyright message

Copyright by the International Forum of Educational Technology & Society (IFETS). The authors and the forum jointly retain the copyright of the articles. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear the full citation on the first page. Copyrights for components of this work owned by others than IFETS must be honoured. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from the editors at kinshuk@massey.ac.nz.

