Australian Journal of Educational Technology

2001, 17(3), 244-264

Plagiarism, learning dishonesty or just plain cheating: The context and countermeasures in Information Systems teaching

Judy Le Heron
Massey University

The incidence of cheating in Information Systems courses has been increasing since the mid-1990s. University entry standards, the nature of the student population, student experiences, student concerns, financial influences, class size, practical skill attainment and employer expectations influence the tertiary education environment in Information Systems in New Zealand. This paper documents our attempts over the period 1996-2001 to counter cheating. This began with a paper based model, moved to computer based assessment, and resulted in the development of the Student Online Assessment Program (SOAP) as an integrated approach which has allowed us to extend the types of computer based assessment we can carry out. The advances we have made and the limitations we have encountered are explored.

Introduction

Cheating has a long and ancient history and the euphemisms to match (Cizek, 1999). University regulations world wide, condemn the practice and threaten disciplinary procedures but seem reluctant to enforce them for fear of litigation. The general experience of staff is that if a student is challenged about cheating it is the staff who are 'on trial' not the student (Innerst, 1998). Institutions prefer to avoid acknowledging that cheating occurs within their walls (Cizek, 1999). This paper firstly looks at factors which influence the educational and instructional setting of Information Systems students and places cheating by this group of students at a major Auckland university in that context. What constitutes cheating is briefly raised and the countermeasures we have employed between 1996-2001 evolving from an 'ad hoc' to a systematic approach are discussed. Finally, the advances we have made in assessment techniques and limitations of our current systematic approach are summarised and some observations made on future directions.

The context

The number and complexity of the factors influencing students studying at university in New Zealand, makes minimising cheating by "reduc(ing) the need and desire to cheat" (Hudspeth, 1997, p. 354) difficult to achieve. Since 1990 tertiary student fees have climbed from near zero to 25% of the cost of tuition (Reidy, 2000a). This paper begins by briefly looking at eight factors influencing the tertiary sector in Auckland with particular reference to Information Systems entry standards, the student population, student expectations, practical skill attainment, class sizes, student concerns, financial influences and employer expectations. Although some research suggests competition for grades is a factor (Cizek, 1999), our experience is of students who cheat just to obtain a pass.

Entry standards

For school leavers in New Zealand, entry to university requires a pass in three bursary exams but there is an 'open entry' policy for those over 21 years. In addition, the discipline of Information Systems is not related to any subjects in the high school curriculum (Wells, 2000) and many diploma students have little computing experience. As a result, few students enter with the basic understanding of the logic necessary to cope with the analytical and programming concepts of Information Systems and many find it more difficult than they had anticipated.

The student population

The student population in Information Systems is dominated by mature students as more people enter tertiary education, re-train or update their qualifications. Approximately 38% of our undergraduate students entered university straight from school. In 2001 the component of new migrants from non-English speaking backgrounds (NESB) in our classes is approximately 67% and although the majority are from Asia, over 20 different nationalities are represented. It is "an increasingly heterogeneous student population with different commitment levels, attitudes and work patterns" (Howells & Piggott, 1992, p. 97).

Student expectations

Student expectation is that their study will qualify them for a high paying job. Many are mature students re-training and in order to re-join the workforce quickly they often take more papers than they can cope with. Some students have the expectation that will pass because they have paid reasonably high fees.

Student concerns

Student representations at Staff-Student Liaison Committee meetings and student approaches to individual Information Systems staff over the last six years have highlighted three main concerns - to "get their money's worth" which is seen primarily as a staff responsibility to deliver a qualification with less emphasis on their own responsibility to learn, that students shouldn't get by in group work on the efforts of others yet they are reluctant to report non-contributing peers, and that widespread cheating (by others) degrades their qualification.

Financial influences

Financial pressures result in many students taking on full or part time employment decreasing the time they can commit to study (Holland, 1995). NESB students often struggle with the full time program of study required to be eligible for the Government Student Allowance.

Class size

In concert with world trends, Government pressure to upskill the population and university management search for economic efficiency has been reflected in increased class sizes. In 2001 class sizes are 400-450 at first year, 200-300 at second year and 120-180 at third year. Personal contact is reduced and students feel less able to ask for help (Gibbs, 1998). Some international students see it as culturally inappropriate to contact staff to ask for help (Collis & Remners, 1997).

Practical skill attainment

Progression through the Information Systems program is aimed at student attainment of analytical, design, modeling, programming, software and project management skills. While individual mastery of these skills is expected so is the ability to work as part of a team. Many students struggle to cope with Information Systems concepts but as long as a group of students can between them produce the desired deliverables for assessment it is difficult to differentiate between the abilities of the individuals in the group.

Employer expectations

Although employers advertise for people with Information Systems skills, they also rank highly strong communication skills, the ability to work in a team and the ability to work within tight timeframes (Clear Communications, 2000). While it would seem a reasonable expectation that graduates have the skills that their degree infers, increasingly employers carry out literacy, numeracy and aptitude testing because, according to a British consultant, "a degree no longer signifies quality" (Reidy, 2000b, p. A13).

Cheating

Yet the extent of cheating increases if it goes unpenalised (Cizek, 1999). This section looks particularly at the ambiguities around cheating. The intent of assessment is to determine whether each student understands the technical skills and concepts and can apply them. It has a diagnostic purpose for both staff and student (Snow, 1989) and provides the opportunity for remedial feedback (Crooks, 1988). What constitutes cheating? Put simply, cheating is submitting the work of other people as your own or breaking the rules governing the assessment task. There is a certain ambiguity about when 'collaborating in a learning community to extend knowledge and understanding' stops, and submitting 'only your own work' starts. This is particularly pertinent to courses which encourage or require discussion and group work. Any practical work submitted is likely to be the result of peer group discussion and debate. There is "an inherent conflict between the university's need to recognise collaborative work that serves students well in their careers and lives, and the need to teach students to do their own independent work" (Lathrop & Foss, 2000, p.118).

Cultural interpretations of the rightness or wrongness of 'cheating' differ also (Cizek, 1999). Although there are many pressures on students which may influence their decision to cheat, a deciding factor is often that of opportunity. It is an interesting irony that in a business context the ability to exploit opportunities is applauded. Research in the United States between 1970-1999, reveals that 60-91% of students admitted to cheating at university (Cizek, 1999). At our institution, final examinations are rigorously supervised by invigilators and there is little evidence of examination cheating. However, cheating on coursework is widespread and has been increasing. As class sizes have tripled, familiarity within classes and between staff and students has diminished and the emergence of sub-groups, with no allegiance to the class, has increased. When individual assignments were paper based up to 15% showed an extraordinary level of similarity. Lathrop and Foss (2000) suggest that the best defence against cheating is to make it clear that staff are concerned about cheating and know how to combat it.

A history of countermeasures

This section outlines the problems with paper based assignments in Information Systems courses and our response, which was to make cheating on some elements of coursework difficult, hopefully, impossible (Holland, 1995). We piloted our ideas on the first year course. The different methods we have used and their advantages and problems are summarised in Table 1.

Paper based assignment problems

First year students are expected to become competent in four basic software applications (currently Microsoft Office Professional). Originally, students submitted paper based assignments that demonstrated a complex set of in depth tasks. A mid-semester test included questions on various software operations to verify student knowledge of practical skills. However, it is difficult to phrase this type of question and even competent students found it difficult to describe these operations in writing from their memory of a computer screen. In addition, it discriminated against NESB students who make up the majority in our classes. The paper based assignments focused on achieving excellence by motivating students to extend their own understanding. Marking was carried out by a team of student markers. The marking period was long, partly because of class numbers and partly because student markers gave their own studies priority over their marking responsibilities, which delayed feedback to students.

Table 1: Assessment methods: competency in use of basic software applications

Assessment Method	Advantages	Limitations
Paper based assignment Ability to detect cheating: *low* Ability to prove cheating: *low* Ability to verify student skills: *low*	• Complex set of in depth tasks	• Difficult to maintain marking consistency • Frequent student challenge of their mark • Long marking period • Delayed feedback • Difficult to recognise cheating
Paper based assignment marked in student's presence with oral explanation by student Ability to detect cheating: *medium* Ability to prove cheating: *low* Ability to verify student skills: *low*	• Complex set of in depth tasks • Oral explanation deterred some cheating • Immediate feedback with remedial help	• Difficult to maintain marking consistency • Increased time to mark • Intensive experience for markers • Difficult to prove suspected cheating
Paper based assignment marked in student's absence plus computer based performance consistency test Ability to detect cheating: *medium* Ability to prove cheating: *medium* Ability to verify student skills: *medium*	• Complex set of in depth tasks • Detected seriously incompetent cheating • Positive student response to attempts to prevent cheating	• Increased time to mark • Delayed feedback • Time required to devise tests plus assignments • Time required to devise operational strategy • Difficult to prove suspected but 'competent' cheating

Computer based test Ability to detect cheating: *high* Ability to prove cheating: *high* Ability to verify student skills: *high*	• Basic plus some advanced tasks • Immediate verbal feedback • Majority of students accepted their mark • Students challenging fairness dealt with immediately by academic staff • Minimal cheating attempted • Positive student response to plagiarism control • Marking consistency • Positive student response to transparent marking process • Short marking period • Automatic timer starts with first action and stops test activity at full time • Automated recording of marks • Sense of camaraderie within marking team between senior students and staff	• Time required to streamline operational details • Time required to devise equivalent sets of tests and their marking criteria • Reduced complexity of tasks

Student challenges were frequent due to inconsistencies between student markers and insufficient written feedback. Cheating was very difficult to detect unless two identical pieces of work were seen by the same marker. Anecdotal evidence from students suggested that cheating was widespread making it possible to pass the paper without acquiring the practical skills (Richardson, 1998). These problems triggered the first changes in assessment strategy.

First countermeasure: Oral explanation of skills

Students presented their paper based assignment to a student marker who marked it in front of the student and asked a series of questions requiring him/her to explain how certain tasks were carried out. Students could elect to accept the initial mark or go away and re-do the work correctly (and extend their skills) and be re-marked later. This made it possible to get full marks for an assignment. Once the student decided to accept their mark, the marker could provide remedial feedback. The mid-semester test to verify student knowledge of practical skills was also retained.

The requirement to provide an oral explanation of their work deterred some people from cheating. If a student could not explain how s/he had done the work and the marker suspected it was not the student's own work, s/he was referred to an academic staff member for further investigation before a mark would be awarded. This approach relied on a student confession and if none was forthcoming it was almost impossible to prove cheating had occurred. Cheating continued to be a major problem according to student reports. In an attempt to verify that students actually had the skills that were demonstrated in their paper based assignments we decided to trial computer based testing.

Second countermeasure: Online performance consistency test

To relate a student's practical competence to their in depth paper based assignment the single mid-semester test was replaced with four computer based tests which paralleled the basic skills required for each assignment. We were reluctant to make all practical assessment dependent on these tests without trialling the concept first so students were still required to submit paper based assignments. These were marked without the student being present. The supervised 20 minute computer based tests did not directly contribute to the final assessment but were used to verify that the students had the basic skills to produce the assignment work submitted. Students were generally nervous about the tests and scored a range of marks but we confined our interest to those who scored near zero. Where a student scored very poorly in the online test but very well in the submitted assignment s/he was asked to talk about the inconsistency. Although there were not many students in this category, they all acknowledged their inability to carry out the basic tasks in the computer laboratory, were unable to explain how they had performed so well in the submitted assignment, and accepted the mark of zero for that specific assignment without demur.

The majority of students reacted positively to this attempt to control cheating and felt it also gave them a better opportunity to demonstrate their own practical expertise. Student reaction was positive also because it took less time to sit an online test than to complete a paper based assignment and because they received immediate feedback about their results (and mistakes). The experience of this trial period gave us the opportunity to fine tune operational aspects of online testing. The operational problems that occurred and improvements that evolved will be covered in a later section.

Third countermeasure: Online skills test

The success of online testing as a check on actual skills and positive student reaction to online testing, encouraged us to waive the paper based assignment the following semester and to test practical work totally online. The duration of the test was extended to 30 minutes which allowed us to introduce more complexity into the required tasks. At the conclusion of the test a team of markers marked the students' work on paper marking register. Students signed to acknowledge acceptance of their mark. Marking consistency improved because any areas of doubt could be raised with academic staff immediately. Initially, we used one test for all sessions but subsequently we developed sets of tests so that students coming later in the day did not have an advantage. Each student's work was saved onto a floppy disk in case the student appealed the mark at a later date. After several semesters it was clear that students raised their concerns in the laboratory at the time of the test, or not at all, so we no longer save their test files.

One potential problem was that it was possible for a student to open the work completed by someone at an earlier session. We monitored this by requiring them to save their work using their student ID number as the file name and watching closely during the test for the file name on the screen to match the ID card on the desk. One problem that emerged was that some students, obscured by rows of computers, kept working after the completion of the test. We countered this by allocating one of the marking team to monitor for this. In addition, it was a complex logistical exercise processing 300 students through the 38 PC computer laboratory.

A few students claimed stage fright or inexplicable loss of work "even though I saved" and there was an occasional computer malfunction. These students were generally permitted to re-sit the test later. We felt that by giving these students the benefit of doubt we contributed to a supportive test atmosphere for all students and we avoided penalising those students with a genuine problem. We noticed over a series of tests that those students re-sitting, whom we suspected had limited skills, made no noticeable improvement on their initial scores. This observation, plus the fact that very few students (over several years of online testing) attempted to re-sit, justified our decision not to contest 'hard luck' stories during a test session. Otherwise, cheating appeared to be limited to working after time.

The evolution of operational aspects of online testing

The experience of the four test pilot gave us the opportunity to improve operational aspects such as registering students for specific test sessions, identifying students, organising marking teams, streamlining marking procedures, moving test groups in and out of the laboratory and keeping to schedule. The procedures we have followed, any complications that have arisen and how we have resolved them are summarised in Table 2 and the main points are explained in more detail later in this section.

Registering students

Prior to each test students were allocated specific test sessions. One test session was left unallocated so that students with timetable clashes could re-book themselves into another session without staff involvement.

Identifying students

In the test laboratory, students place their ID cards on their desk. During the session the ID number on the screen is compared with the ID card on the desk and this is verified again during marking. Once their work is marked the students are asked to sign that they accept the mark they have been awarded and their appearance and signature are compared with their ID card. Identification at several different points in the process was justified when one student assured us that he had taken the test and that we must have 'lost' his marks. However, when we told him that not only was there no mark recorded for him in the course database but also there was no record of him registering his arrival for the test, logging on to a computer or signing to accept his mark, he admitted that perhaps he was confused.

Table 2: Operational procedures: Arriving at the specifics

Timing	Tasks	Complications	Resolution
Weeks before test	• Allocating students to test sessions • Marking team run through test skills	• Students want to change test sessions	• One unallocated test session at end of day - students make own changes
On the day	• Setting up lab • Setting up registration desk • Markers briefing	• Test password security • Student markers own class commitments	• Academic staff log on all computers • Flexitime for markers to allow lecture attendance and breaks
Prior to the test	• Registering student arrival for allocated session • Pre-test briefing for students sitting test outlining procedures and rules	• Students registering for different session than allocated • Students not registering and queue jumping into test • Cellphones with text messaging facilities	• Preference given to students allocated to session, others only if spare computers • Registered students given numbered card - retrieved when they entered lab • All possessions left at front of lab

During the test	• Deterring cheating • Handling late arrivals • Handling machine problems • Marking early completions	• Accessing previous students test files • Working after time • Computer problems	• Student required to save file with own ID number and work only in that file • Automatic timer starts with first action and stops test activity at full time • If valid computer problems student given extra time or re-sit allowed • Student given extra time flagged by sign on top of monitor and observed by marker making decision
After the test	• Marking full time completions • Test security	• Student identification • Saving all student test records in case of subsequent appeal • Test papers removed from lab	• Student required to save file with own ID number • Any queries must be raised prior to signing to accept mark so test records need be saved after test • Signature on test register, ID number on test file and student appearance, compared with student ID card information • Test papers taped to desk inside plastic sleeves • New version of test used after every 2 test sessions

Marking procedures

Markers explain the marking procedure to students as they mark so that students can see and understand where they went wrong and, if time permits, what should have been done. Students have the opportunity to raise any concerns about their marks. Once these concerns are resolved the student accepts his/her final mark. If markers are fast and uncommunicative, and students are confused about how and why they lost marks, the marking experience has the potential to become antagonistic.

Test security

Three levels of security protect the integrity of the supervised test and the course database. Computers in the test laboratory are logged on to the test environment only by academic staff using a password. This provides access to the test database. When students enter the laboratory, each student logs on using his/her student ID.

We have developed sets of similar tests so that one version is only used for two test sessions before being changed. (Students leaving the first session are not able to communicate with the group going into the laboratory for the next session.) However, because the tests are similar we still do not let the test leave the laboratory to avoid advantaging students tested later in the day.

Development of the Student Online Assessment Program (SOAP)

The pilot computer based tests assessed student competence with commercial applications and streamlined our procedures, but once we had demonstrated that the advantages exceeded the limitations, it was apparent that we needed special purpose software to overcome remaining problems. The Student Online Assessment Program (SOAP)1 was developed as an integrated approach to meet that need.

Basic facilities

A marking panel, specifically for assessment of application software competence, was developed which sits on top of the test application. Markers record tasks listed on the panel as being achieved, or not achieved. Students are still required to sign a paper register to accept their mark then the marker, using another keystroke combination, saves the results directly to the course database. The marking panel has made marking decisions visible to students and eliminated the need to transfer marks from paper records.

A test log on screen is displayed at the commencement of the test session and each student must log on using his/her student ID number. The course database is checked to see if the student record exists and if not (e.g. late enrolment) the student re-enters his/her ID number to create a new record. A timer, developed to prevent students working after the completion of the test, is activated when the 'start' button is clicked to

1 The SOAP software is being developed incrementally by team member, Tony Richardson. The work on computer-based assessment is partly supported by a joint grant to Tony Richardson and Judy Le Heron from the Fund for Innovation and Excellence in Teaching, Massey University.

begin the test. At the end of the allocated time access to the test software is blocked by a 'stop' screen. Markers clear the 'stop screen' with a keystroke combination to proceed with marking.

Extending the types of assessment

Once the first year practical work was virtually cheat proof we extended online testing to practical aspects of two second year papers, focusing on assessing competence in Structured Query Language (SQL) and in analysing and modeling key information from a scenario onto skeleton Dataflow (DFD) or Entity Relationship (ERD) Diagrams. Although DFDs and ERDs model totally different information these two tests are similar in concept.

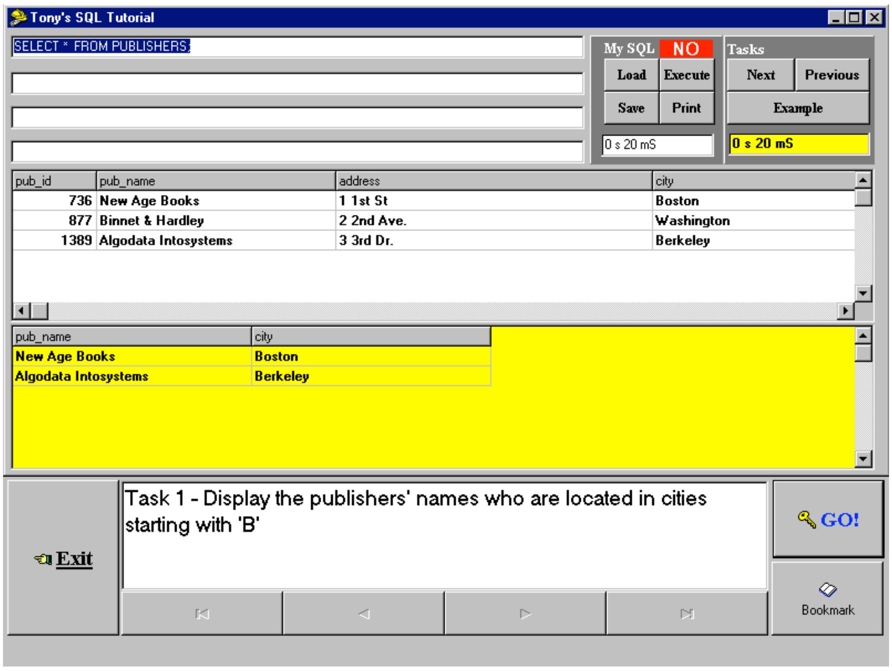
Table 3: Extending the Pilot: SQL programming *and* Analysis with DFDs and ERDs

Extensions	Advantages	Limitations/Complications
Computer based SQL programming test	• Can be used as self assessed tutorial • Syntax error feedback when code executed • Student can refine code • Automated marking • Different test for each student by random selection of questions • SQL statement can be evaluated for efficiency	• Requires large question bank • Assessing level of difficulty of questions
Computer based DFD **and** ERD modeling tests	• Can be used as self assessed tutorial • Tests ability to analyse a scenario to determine inputs, outputs and processes **or** stored data requirements • Tests ability to document key features of analysis on the skeleton DFD provided **or** Primary and foreign keys on the skeleton ERD provided • Eliminates cheating	• Requires minimally four different but similar test versions, plus additional tutorial versions • Automated marking requires specific placement of dataflow, process, datastore and external entity names **or** primary key, foreign key, relationship and entity names • Automated marking requires there be no ambiguity in answer choices or multiple correct answers • Requires programming expertise to change the skeleton diagram and link new 'object' positions to the answer database • Adding non-key attributes would introduce ambiguity and complicate automated marking • Adding non-key attributes would lengthen the test time

As with our experience with the first year tests, attempts at cheating have been minimal and unsuccessful. The advantages, limitations and complications we discovered are outlined in Table 3 and detailed below.

Extension One: Online SQL test

At second year, programming competence in basic SQL is required. Assessing students' ability to use SQL from a paper based assignment was complicated not only by plagiarism but also by the variety of statements that could be written to achieve the correct result and the need to check syntax, logic and the underlying data. Software was developed specifically to enable students to write generic SQL code in an environment independent of the idiosyncrasies of specific database software (Richardson, 2000). Students are provided with the structure of the underlying database tables and the software presents them with a series of randomly generated 'requests for information' from these tables. Students enter SQL code on the screen shown in Figure 1.



```
SELECT * FROM PUBLISHERS;
```

pub_id	pub_name	address	city
736	New Age Books	1 1st St	Boston
877	Binnet & Hardley	2 2nd Ave.	Washington
1389	Algodata Intosystems	3 3rd Dr.	Berkeley

pub_name	city
New Age Books	Boston
Algodata Intosystems	Berkeley

Task 1 - Display the publishers' names who are located in cities starting with 'B'

Figure 1: SQL Test Screen

The software attempts to compile and execute the SQL code and to return the requested information. First, it detects syntax (grammatical) errors and generates appropriate error messages to allow students to make corrections. Second, once the syntax is correct, the software will detect errors in the logic. Since there are multiple ways to design a query it is assessed by comparing the query result with the correct answer. Third, the efficiency of each query is demonstrated, by displaying the time it took to execute, and the time taken to execute the model query. This enables efficiency to be incorporated as part of the marking criteria. To ensure the students were confident about using the software itself, and to allow them to practise their SQL skills, the SQL software was made available for tutorial use for several weeks prior to the test. This tutorial use is an advantage that students appreciate but it does require, along with the tests, a large question bank.

The first 30 minute test was run with 6 questions (2 easy, 2 medium difficulty, 2 hard) with an associated differential in marks. The set of test questions was changed after every two test sessions. A problem was that students unable to answer the first 'easy' question often found it difficult to move on to the next which was known to be 'more difficult'. Associated with this was the problem of evaluating the level of difficulty of SQL queries because people's perception of what is difficult varies. The second cycle of this test involved a series of questions (of equal worth and of varying difficulty but avoiding the extremes) randomly generated for each student. This approach received a more positive response. Students were not distracted by any inferred level of difficulty and because perception of difficulty differs by individual they felt that equal marks for each query was more fair.

Extension Two: Online DFD analysis test

At second year, competence in analysis and DFD modeling skills is assessed in a 30 minute test. The software was designed to replicate a CASE tool diagram, but to gain the advantages of automated marking, correct answers had to be recorded in specific locations on the diagram to enable comparison with the correct answers in the database. Students were provided with a printed scenario and paper for trying out their ideas. They were required to analyse and model the data inputs and outputs for a specific information system. The software presented students with a range of appropriate and inappropriate choices from which to label the components of a skeleton DFD as shown in Figure 2. At the end of the test the marker used a keystroke combination to activate automated marking. Incorrect answers were highlighted for the student to see and query before the marks were recorded in the course database. To provide an equivalent test for each student, the scenario, diagram and choice of naming options remained the same for each test but the particular information system to be modeled was changed after every two test sessions.

The advantage of this test was being able to assess the ability of the student to determine what was relevant to the information system they were modeling, and their ability to document this on the DFD provided. This test software was also available, with a different scenario, for tutorial use so students could feel confident about the software itself before they used it in an assessment situation. It was difficult eliminating all ambiguity in labeling options. This was done by including input and output dataflows in a different combination for each object on the diagram so that relevant information from a scenario could only be placed in specific positions to be correct. It was also difficult providing an equivalent test for different test sessions. It was achieved by creating the scenario in such a way that four simple but different information systems could be created from it, each based on the same diagram structure. The 'choice list' from which labels could be selected had to be structured so that only one label could be correct for any object on the diagram whatever version of the test was used.

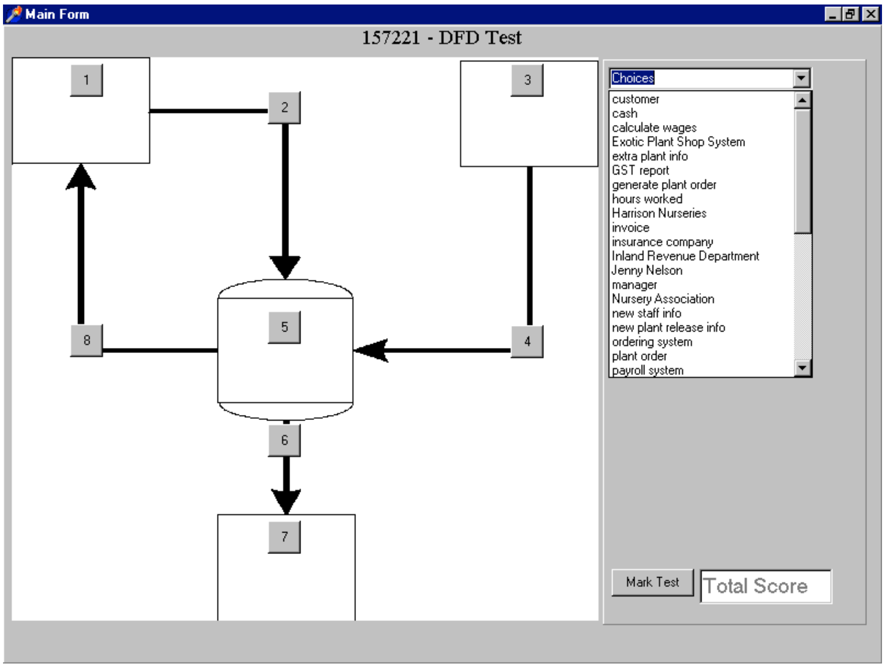


Figure 2: DFD Test Screen

Initially we thought that the multi-choice aspect of the 'choice list' would make it a very simple test. However, a single simple scenario with four threads proved quite challenging for students used to detailed scenarios with a single focus. Some students were not happy that they could choose all the right labels but be marked wrong if the labels were placed in the wrong locations. While there is some justice to that argument, it was explained prior to the test that the diagram had to be viewed as a total picture. So, for instance, if an external entity was incorrectly labeled, the combination of dataflows to/from that external entity on the diagram would not match the information given in the scenario.

Extension Three: Online ERD analysis test

This 30 minute test assessed the student's ability to analyse and model aspects of data on a skeleton ERD. This test was identical in concept to the DFD test previously described. The advantages and limitations of this test were also similar. However, there were some unique limitations modeling the ERD. First, it was impossible to create a complete list of key and non-key attributes without an endless number of valid choices being available for some entities across the four versions of the test which would make automated marking almost impossible. Second, our experience with the DFD test was that students were kept quite busy labeling 8 items in 30 minutes and that adding non-key attributes would require an increase the length of the test without significant benefit. The diagram devised for this test represented a limited ERD model. We decided that the most important aspects of data modeling that we wanted to assess were the ability to chose primary and foreign keys. Our three entity test diagram provided optionality and cardinality, and students were only required to name entities, relationships, primary and foreign keys.

Conclusions: Advances made and limitations and complications to be overcome

Over the period 1996-2000 we have made significant changes in the way we assess some of our Information Systems courses where assessment has both a formative and summative role. These changes were initiated to counter an increase in cheating which was concerning both staff and students. It also became a way we could manage the burgeoning marking workload associated with increased student numbers while maintaining the quality of our instruction (Ward & Jenkins, 1992). This section summarises the process followed to introduce these changes, the advances made and the limitations or complications of our approach. It concludes with some observations on future directions.

The process

Change does not guarantee improvement so our first foray into computer based assessment was piloted on one course and run in parallel with the paper based assessment (Harvey & Mogey, 1999). This demonstrated to both staff and students that our computer based assessment was fair and it engendered strong student support for our efforts to counter cheating (Parry, Darling & Robbin, 1998). Progress towards the current version of SOAP has proceeded in stages. Automatic timing, automated marking, SQL programming and analytical diagrams were introduced individually and in different courses in subsequent semesters and operational procedures were refined (Whittington, 1999). This progression allowed us to evaluate each new aspect of functionality in the context of a successful assessment process (Cava, 1991).

Students have, by and large, accepted that whether the assessment is perceived as easy or difficult it is seen as fair, and both student and student marker feedback has been invaluable (Quirk, 1995). Over this period, as its usefulness has been demonstrated, other staff have become involved in using our computer based assessment approach which is now used in all or part of the coursework assessment for one first year and two second year courses.

Advances made

We feel that our approach to computer based assessment at second year meets the first four of Bloom's taxonomy of educational objectives, namely knowledge, comprehension, application and analysis (Pritchett, 1999). Although our original concerns were to control cheating and cope with increased student numbers, we have also been able to improve the instructional process for our students by focusing on assessing key learning objectives (Race, 1995) and increasing the range of assessment methods for a course so that a student less proficient in one type of assessment is not penalised (Charman, 1999). This section outlines the advances SOAP has achieved.

First, cheating has virtually been eliminated because students produce their assessed work under supervision (Cann & Pawley, 1999). Students consider that small test groups and multiple supervisors are the most effective deterrents to cheating (Cizek, 1999). Second, assessment is carried out quickly, and consistently across students, test sessions and semesters (Crooks, 1988). Marking is done in front of the student which allows some time, although limited, for explanation of mistakes (Ladyshewsky, 1995). Third, IT skills are being tested with IT tools, both standard and specifically written for our courses (Dowsing, 1999). For our students, as potential information systems developers, there is added value to increasing the range of computing experiences they have. NESB students benefit from the opportunity to demonstrate their practical skills instead of just writing about them (Biggs, 1997). Fourth, students appreciate having access to self testing tutorials for each type of assessment (Howells & Piggott, 1992) which gives them feedback about their mistakes and allows them to improve their understanding and skills before they are assessed (Keyton, 1999). It also ensures familiarity with the type of assessment being used so that the software itself is not an obstacle to their performance (Race, 1995). Computer based self testing tutorials allow students to work at their own pace and this enables staff to give individual assistance when it is needed which enhances staff-student interaction. Fifth, automated marking and recording of marks reduces the administrative burden (Harvey & Mogey, 1999).

Limitations and complications of our approach

Although computer based assessment minimises cheating, speeds up and improves the consistency of the marking process, it comes at a price. The price is the investment of time and effort, not only in the development of the software and operational procedures but also in creating the scenarios, test and tutorial questions and devising answers which are the basis of automated marking.

First, the creation of test banks must be sustainable (Brown, Race & Bull, 1999). SQL tutorials and randomly generated tests particularly require a large bank of test/tutorial questions to challenge the students and avoid rote learning. An additional problem is the difficulty of estimating how hard each question is so that every student has a similar test experience (Simms Williams, Maher, Spencer, Barry & Board, 1999).

Second, a limitation relevant to the DFD and ERD tests is that automated marking requires a specific location on the diagram for each 'object' being named. This is in conflict with Pritchett (1999) who states that a correct answer should not depend on a previous answer being correct, and that there should be only one unquestionably correct answer. However, for us to achieve the latter objective and automate marking, we have an interdependence between answers. It is explained to students and demonstrated in the self testing tutorials that the diagrams have to be seen as a whole picture not just as component parts. The scenarios for these tests/tutorials are written so that only one label is correct for each 'object'. Penalising wrong answers and rewarding correct answers also discourages guessing. It is emphasised to students that these are tests of their analytical skills rather than their diagramming or labeling abilities. They learn to identify information relevant to their task from a scenario which also incorporates information relevant to other tasks that are not required of them (Patel, Russell & Kinshuk, 1999).

Future directions

From a student perspective, SOAP presents an integrated entry and a common mode of operating our computer based assessment while also providing staff with three distinct approaches to test and tutorial support.

This type of assessment provides clear rules and procedures and is unequivocally the student's own work. But our approach is not perfect and future directions are outlined.

At the moment making changes to the SQL, DFD and ERD tutorials and tests is dependent on the programming skills of the developer so it is the intention to develop SOAP so that it is straight forward for any Information Systems staff member to change content (Brooking, 1996). Ideally, SOAP should also be more flexible in its automated marking of 'objects' on DFDs and ERDs but this is not considered achievable at this stage. Some types of assessment are clearly not suitable for computer based testing and until now we have placed object oriented programming in this category. However, after being challenged by a student denial of cheating in this particular assessment "even in the face of overwhelming evidence" (Ryan, 2000, p.58) we will be exploring how that assessment too can be tested online in 2001.

It seems we are not alone. Cizek (1999) warns that a disenchanted student can claim legal redress on three grounds: that the institution was in breach of contract to provide competent instruction; that procedural due process whereby institutional procedures are followed was not observed; or that s/he was denied substantive due process whereby consistent treatment ensures fundamental fairness. 'Robust procedures' is the catch cry and we now explicitly state in our Course Introduction that all students must be able to explain and discuss any material they submit for assessment (Ryan, 2000). The New Zealand Universities Academic Audit Unit makes it clear that universities should have systems in place to reduce cheating especially "in the face of a fee paying and increasingly litigious student body" (NZUAA Unit, 2000, p.11). The measures that we have operationalised to successfully counter cheating have been painstakingly implemented over the last six years yet, like others, we find that unethical student conduct is still difficult to prove to the university's satisfaction. It seems ironic that what was initiated in 1996 to protect the integrity of the institution may well be necessary in the future to demonstrate the integrity of the staff.

Acknowledgement

The helpful comments of the referees were appreciated.

References

Biggs, J. (1997). Teaching international students. *Teaching Education News*, 7(5).

Brooking, A. (1996). *Intellectual Capital*. International Thomson Business Press, London.

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

Cann, A. J. and Pawley, E. L. (1999). Automated online tutorials: New formats for assessment on the WWW. In S. Brown, P. Race and J. Bull (Eds), *Computer-Assisted Assessment in Higher Education*. Kogan Page Ltd, London, pp. 39-45.

Cava, R. (1991). *Dealing with Difficult People*. Judy Piatkus Publishers, London.

Charman, D. (1999). Issues and impacts of using computer-based assessments (CBAs) for formative assessment. In S. Brown, P. Race and J. Bull (Eds), *Computer-Assisted Assessment in Higher Education*. Kogan Page Ltd, London, pp.85-93.

Cizek, G. J. (1999). *Cheating on Tests: How to Do It, Detect It, and Prevent It*. Lawrence Erlbaum Associates Inc, Mahwah, New Jersey.

Clear Communications (2000). Full page advertisement for jobs available. *New Zealand Herald*, 30 Oct, p. A14.

Collis, B. and Remners, E. (1997). Cross Cultural Communication and Interaction. In B. H. Khan (Ed), *Web-Based Instruction*. Educational Technology Publications, Englewood Cliffs, New Jersey, pp. 85-92.

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

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

Gibbs, G. (1998). Teaching large classes: Strategies for improving student learning. Address at Stanford University, California, 24 April. [verified 2 Nov 2001] http://sll.stanford.edu/projects/tomprof/newtomprof/postings/21.html

Harvey, J. and Mogey, N. (1999). Pragmatic issues when integrating technology into the assessment of students. In S. Brown, P. Racel and J. Bull (Eds). *Computer-Assisted Assessment in Higher Education*. Kogan Page Ltd, London, pp.7-19.

Holland, M. (1995). Cheating: Students find academic dishonesty tempting in test situations. [verified 2 Nov 2001] http://www.thejack.nau.edu/19951004/life7.html

Howells, K. and Piggott, S. (1992). Guided reading in biology: A modified Keller System. In G. Gibbs and A. Jenkins (Eds), *Teaching Large Classes in Higher Education*, Kogan Page Ltd, London, pp.88-98.

Hudspeth, D. (1997). Testing learner outcomes in web-based instruction. In B. H. Khan (Ed), *Web-Based Instruction*. Educational Technology Publications, Englewood Cliffs, New Jersey, pp. 353-356.

Innerst, C. (1998). Universities in war on cheating. *Washington Times*, 29 January. Cited by National Center for Policy Analysis, College Student Cheating Widespread. [verified 2 Nov 2001] http://www.ncpa.org/pi/edu/jan98o.html

Keyton, J. (1999). *Group Communication*. Mountain View, Mayfield, California.

Ladyshewsky, R. (1995). Cited in G. Suddaby, *Types of Feedback*. Training Development Unit, Massey University.

Lathrop, A. and Foss, K. (2000). *Student Cheating and Plagiarism in the Internet Era: A Wakeup Call*. Libraries Unlimited Inc, Englewood, Colorado

New Zealand Universities Academic Audit Unit (2000). *Good Practice Digest*. [verified 2 Nov 2001] http://www.aau.ac.nz/june.shtml

Parry, C., Darling, M. and Robbins, S. (1998). Putting best practices into practice. *The Systems Thinker*, 8(10), 1-5.

Patel, A., Russell, D., and Kinshuk (1999). Assessment in a cognitive apprenticeship-based learning environment: Potential and pitfalls. In S. Brown, P. Race and J. Bull (Eds), *Computer-Assisted Assessment in Higher Education*. Kogan Page Ltd, London, pp.139-147.

Pritchett, N. (1999). Effective question design. In S. Brown, P. Race and J. Bull (Eds), *Computer-Assisted Assessment in Higher Education*. Kogan Page, 29-37.

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

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

Reidy, J. (2000). Knowledge v economy - Part I. *New Zealand Herald*, 5 January, p.A12.

Reidy, J. (2000). Exodus of the graduates - Part II. *New Zealand Herald*, 6 January, p.A13.

Richardson, T. (July,1998). First year university Information Systems Papers: Optimising learning - minimising administration. Presented at *Computers in Teaching '98*, Dublin.

Richardson, T. (May, 2000). Technology in education - keeping students honest. Presented at International Resource Management Association 2000, Alaska.

Ryan, J. C. H. (2000). Student plagiarism in an online world. In A. Lathrop and K. Foss (Eds), *Student Cheating and Plagiarism in the Internet Era: A Wakeup Call*. Libraries Unlimited Inc, Englewood, Colorado, pp. 56-59.

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

Snow, R. E. (1989). Toward assessment of cognitive and conative structures in learning. *Educational Researcher*, 18(9), 8-14.

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

Wells, A. (2000). Teachers decry low computing status. *New Zealand Infotech Weekly*, No. 462, Oct 8, p.3.

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

Judy Le Heron is an Academic Tutor in Information Systems in the Institute of Information and Mathematical Sciences at Massey University, Albany Campus, Auckland, New Zealand. Her email address is J.L.LeHeron@massey.ac.nz