Session 2220

Continuous monitoring and class performance

N.W. Scott & B.J. Stone
The University of Western Australia

Abstract

Most lecturers faced with large classes find it difficult to know the current state of the class. For example, are there a significant number of students who are not keeping up or having extreme difficulty with a topic? If problem classes are provided for smaller groups then sometimes feedback may get to the lecturer. However, often the students at risk do not even attend such classes, perhaps wishing to be saved any embarrassment. The consequences, particularly at first year level, are that many of these students fail. At the University of Western Australia an attempt has been made to address the two problems noted above (feedback to lecturers and identifying students at risk). This has been achieved for a first year engineering dynamics class which historically students have found hard. All example problems are now presented on a computer with the same questions for each student but with unique numerical values. Students type in their answer and are required to get a problem correct before being allowed to proceed on to the next. Approximately two out of ten problems are formally assessed and contribute to 20% of the mark for the unit. The computer provides diagnostic help on all non-assessed problems. A server records all student responses and various displays are instantly available to the lecturer so that the class may be continuously monitored. This paper describes the methodology for this approach and discusses the outcomes for lecturers and students.

1 Background

The starting point for the developments described here is that most students are opportunistic: that they will generally do whatever they perceive is necessary to ‘get through’, or as Ellison (1992) politely puts it, ‘assessment draws learning through a course’. Students are under a lot of pressure and they generally develop survival strategies which are not in harmony with deep academic exploration.

In the years before 1995 the engineering dynamics course at UWA was taught using a fairly common tutorial model: groups of perhaps twenty students with one postgraduate tutor. Each week every student was expected to attend two one-hour classes of this kind, and each week there was a new assignment sheet with two to four quite difficult problems. Students were told to attempt these problems and hand the results in to the tutor for feedback; however no formal ‘marks’ were attached to this important work or for attendance at the problem classes. After the first month of these classes tutors observed a marked downturn in both attendance and the quantity of work submitted. In a sample of 53 students from one tutor’s classes in 1990 and 1991 there was a strong correlation between non-participation in the tutorials and failing the unit.

We assert that this syndrome is fairly common. One interpretation is that after four weeks of classes students begin to realise that no-one is ‘watching’ them to ensure attendance and compliance; and being busy, they choose to spend their time elsewhere. It is also possible that these students discover that the problem classes do not ‘work’ for them: that they are too difficult, or too noisy, or that the tutor is on a different wavelength – and then opt to study in some other environment, such as the cafeteria.

What is our response to this situation? The tutorial model used in many maths/science subjects has been around for a long time: is it now suddenly failing to meet the students’ needs? If so, do we moralise about good study habits, or do we change the tutorial system to make it both profitable and *unavoidable*? If we adapt the system to the peculiarities (failings?) of students, are we *pandering* to them when discipline is in order? Do we reinforce or even sanction inappropriate adult behaviour by removing the element of free will?

Our approach has been to attempt to introduce changes to improve the match between learner and teaching system, making use of new technologies where appropriate. Our most significant ‘experiment’ in this area was to require the Dynamics class of 1995 to interact with a computer-based tutorial system we developed (see Devenish 1995 for more information about this system). This paper focuses on the part of the tutorial system that monitored the class as a whole and the progress of each student in particular, and the effect that this policy had on the life of the class.

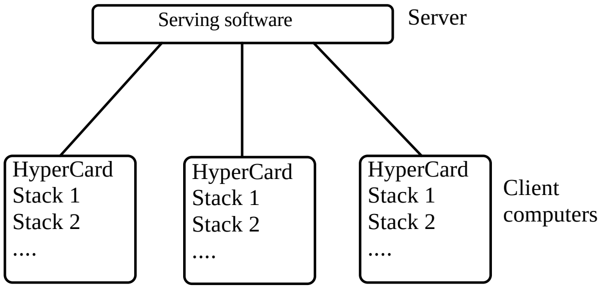


Figure 1 Schematic relationship between client and server computers.

2 Description of the monitoring system

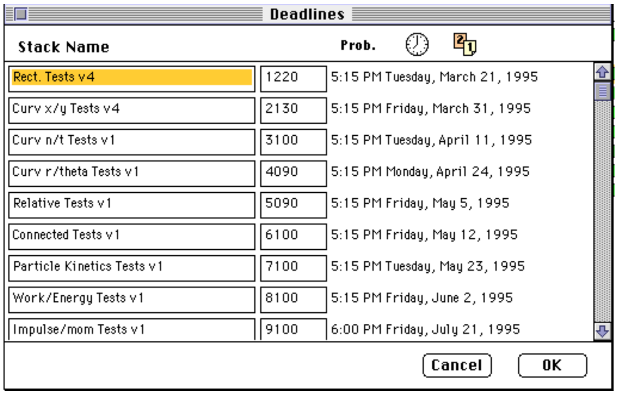
270 first-year engineering students used the tutorial system during 1995. Each week there were two 45-minute lectures and two one-hour problem classes. These classes were held in a large computer laboratory with 128 Macintosh™ 'terminals'. Class sizes were about 40 to 50 students with one tutor present during the assigned times. Most students chose to do their work during their assigned times.

```
Tuesday, 22 August 1995
11:35:03 AM     LOGI
11:35:09 AM     stack "Login to Dynamics 100"; card "loginCard"; goToProblemAfter 12090
11:35:19 AM     stack "Kinematics Tests 1 v1"; card id 13074; 12100; openCard
11:35:39 AM     stack "Kinematics Tests 1 v1"; card id 13074; 12100; studentAnswer = -16.52m/s^2
11:35:47 AM     stack "Kinematics Tests 1 v1"; card id 13074; 12100; dialog: Your answer is incorrect. Try again for 2 marks, or view the help for 2 marks.
                (Help, *Try Again, )
11:37:43 AM     stack "Kinematics Tests 1 v1"; card id 13074; 12100; studentAnswer = -18.59m/s^2
11:37:44 AM     Attempt at problem 12100; DN = 1; SN = 5; ED; LD
11:37:47 AM     stack "Kinematics Tests 1 v1"; card id 13074; 12100; dialog: Excellent! You have been credited with 2 marks out of 10. (Solution, Same,
                *Next)
11:37:59 AM     stack "Kinematics Tests 2 v1.1"; card id 21026; 13000; openCard
11:38:13 AM     stack "Kinematics Tests 2 v1.1"; card id 21026; 13000; dialog: You must complete all the problems in this stack by Friday 1 September at 6
                pm (i.e. when the Maths Lab closes) or forfeit all marks for the assessed problems. (*OK, , )
11:38:24 AM     SEPR
11:38:47 AM     stack "Login to Dynamics 100"; card "loginCard"; dialog: You have solved 133 problems of 146 problems attempted.
                So far you have acquired 249 marks out of a possible 260. (*OK, , )
11:39:07 AM     stack "Kinematics Tests 1 v1"; card id 21026; 12000; openCard
                CLOS
```

Figure 2 Some 'event' data recorded by the server, representing the actions of a particular student.

Each student was expected to log in to the system using a password and work through a sequence of about 200 problems. The problems were grouped into sets of about ten. Each set began with fairly simple problems in a particular topic, and ended with two assessed problems of examination standard. The problem sequence was supposedly inviolable: the current problem was to be solved before the next one could be attempted. However – opportunistically – some students found ways to 'get around' this requirement and attempt only the assessed problems in some sets. This behaviour – and many others – was observable from the 'server' computer.

The server was another Macintosh computer, located in a locked room a short distance from the computer laboratory. This computer stored a separate file for each student in the course, and communicated with the students at their 'terminals' using a local network (see Figure 1).



Stack Name	Prob.	
Rect. Tests v4	1220	5:15 PM Tuesday, March 21, 1995
Curv x/y Tests v4	2130	5:15 PM Friday, March 31, 1995
Curv n/t Tests v1	3100	5:15 PM Tuesday, April 11, 1995
Curv r/theta Tests v1	4090	5:15 PM Monday, April 24, 1995
Relative Tests v1	5090	5:15 PM Friday, May 5, 1995
Connected Tests v1	6100	5:15 PM Friday, May 12, 1995
Particle Kinetics Tests v1	7100	5:15 PM Tuesday, May 23, 1995
Work/Energy Tests v1	8100	5:15 PM Friday, June 2, 1995
Impulse/mom Tests v1	9100	6:00 PM Friday, July 21, 1995

Figure 3 The deadline editor window.

Each student file had several purposes:

1 To store the identity of the student: their names, password, high-school leaving score, formal examination results and a digitised photograph;

2 To store the current status of the student i.e. which problem they had completed, and a summary of progress and marks for all problems attempted;

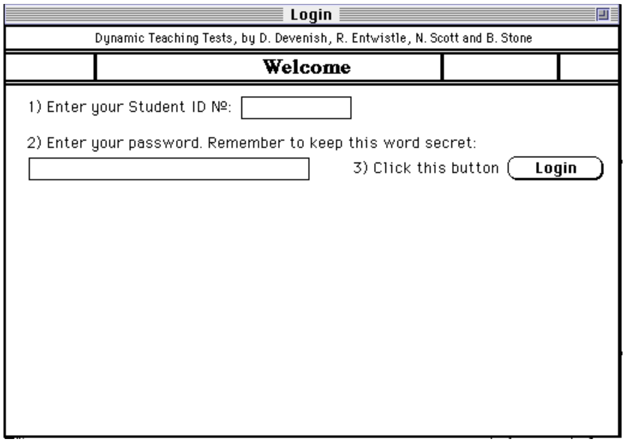
3 To store a detailed record of 'what the student did' while at the 'terminal'. For example, every time the student moved to a different 'page' (card) of the tutorial stack, or clicked a button, or entered an answer to a problem, this

information was recorded in an extended, human-readable form (see Figure 2). In some cases this part of the file reached one megabyte in size over the whole year’s work.

The server also maintained a list of deadlines for each served course. This list defined the sequence of HyperCard stacks that made up the course, as well as the deadline for each one (Figure 3).

The general form of the relationship between the terminal and server had the terminal (client) software sending requests or event information down the line to the server. The server would access the file of the student in question and respond, usually within a second or so. Here is a schematic series of server-client interactions:

- Initially the screen resembled Figure 4.
- A client computer sends a student ID code and password to the server.
- The server looks up the file with this ID code and compares the password to the correct one. An acknowledgment (with an error message if necessary) is returned.
- If no error occurred the client requests a package of information about the student: names, completed problem Nº, authority code, and the name of the particular course the student has logged into (since the server actually served several courses simultaneously).
- The server receives a request for (e.g.) the student’s names, so it reads this from the student’s file and sends it back.
- The client, on receiving the student’s names, declares the session open and changes the appearance of the screen to that of Figure 5. The student then has the option to ‘Go to next problem’. If this button is clicked a message goes to the server.
- If the student has asked to ‘go to next problem’ the server checks the list of deadlines and decides which HyperCard stack the student should be working on. This name is returned to the client.
- The client software opens the named stack and brings up the first problem the student has not yet completed. As soon as the problem card is revealed a message is sent to the server, requesting the appropriate random number seed.
- The server returns a random seed based on a formula that uses the student’s ID code and the problem number.



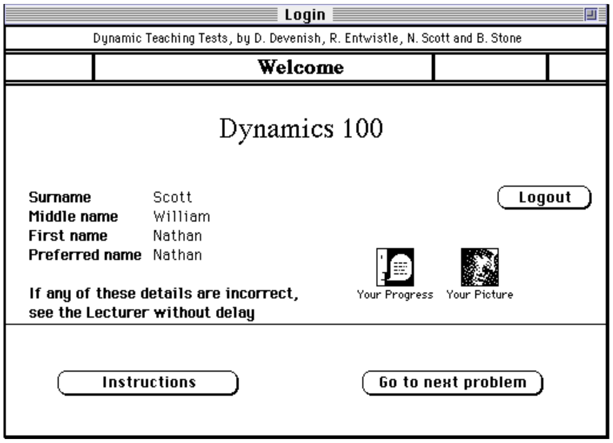
Dynamic Teaching Tests, by D. Devenish, R. Entwistle, N. Scott and B. Stone

1) Enter your Student ID №:

2) Enter your password. Remember to keep this word secret:

3) Click this button

Figure 4 Initial state of login card.



Dynamic Teaching Tests, by D. Devenish, R. Entwistle, N. Scott and B. Stone

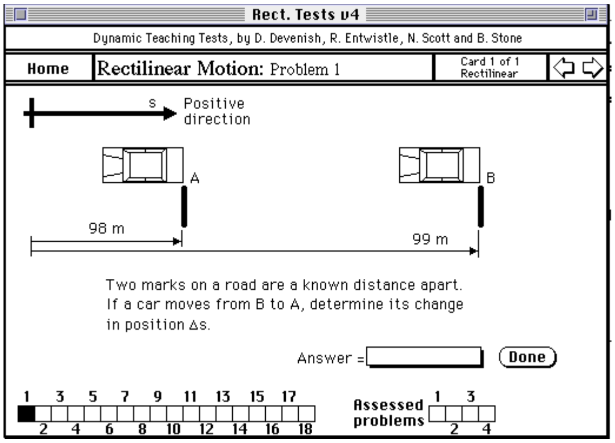
Dynamics 100

Surname Scott
Middle name William
First name Nathan
Preferred name Nathan

If any of these details are incorrect, see the Lecturer without delay

Figure 5 Student is logged in.

- The client installs the random seed in the operating system and then proceeds to calculate and display the parameters of the current problem. Since the parameters are calculated using calls to a ‘random’ function, and since the seed for the pseudorandom sequence has been specifically set, the student will be given the same set of problem parameters each time they view this problem, even if they have logged out in between attempts. At this stage the screen resembles Figure 6.
- The student may then type an answer to the problem and click the ‘Done’ button. The answer is parsed and processed locally at the client terminal, reducing the workload on the server. Regardless of the outcome of this assessment a message is sent to the server.
- The server receives all such messages and continuously updates a visual display (described in the next section) that represents the student’s progress through the problem sequence. These progress displays, when seen in parallel for the whole class, are a striking overview of the status of the whole as well as of individual students.



Dynamic Teaching Tests, by D. Devenish, R. Entwistle, N. Scott and B. Stone

Rectilinear Motion: Problem 1

Two marks on a road are a known distance apart.
If a car moves from B to A, determine its change
in position Δs.

Answer =

Figure 6 A typical problem card.

The server records all of the interactions described above, along with a unique timestamp to the nearest second. The resulting files are a tremendous resource for educational research which is currently being exploited by the authors.

3 Displays for use by the lecturer

The server always shows the status of every student in a graphical manner. A small part of the display for a particular course is shown in Figure 8.

The display is divided into rows; each row represents the progress of one student through the course. The student’s name and ID number are shown at the left; then there are some icons representing whether this student is ‘logged in’ and whether there is a ‘login message’ waiting for them. To the right of these icons there is a shaded display area. Each square in this area represents one problem that this student has either completed or is working on; the row of squares taken together is called a *progress bar*.

There are some conventions associated with the colouring of the progress bar (Figure 7). If the student has completed a problem the corresponding square has a black border, otherwise grey. Non-assessed problems are shown in green. Since this is printed in shades of grey the reader will have to imagine! The darker the green, the more times the student had to enter an answer before they got it right. This feature means that if many students are having difficulty with a particular problem, a v*ertical band of darker colour* shows up in the progress window. In turn this can prompt the lecturer to mention the problem and associated theory in the lectures, within hours of the tutorial session.

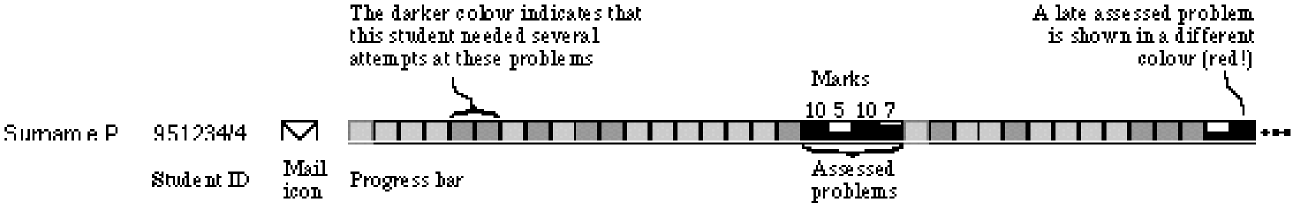
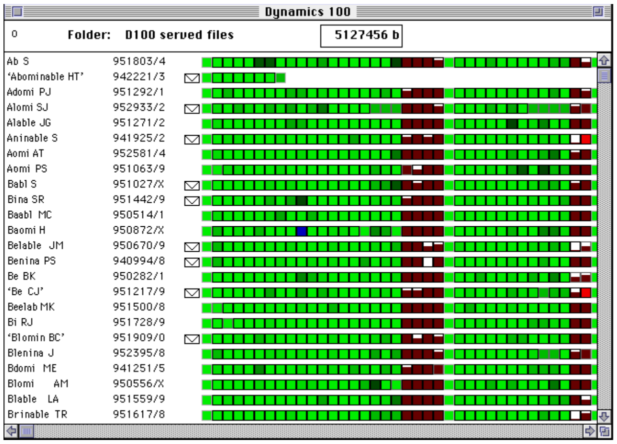

The darker colour indicates that this student needed several attempts at these problems
A late assessed problem is shown in a different colour (red!)
Marks
10 5 10 7
Surname P
951234/4
Student ID
Mail icon
Progress bar
Assessed problems

Figure 7 Nomenclature of one student’s progress record.



0 Folder: D100 served files 5127456 b

Ab S	951803/4	
‘Abominable HT’	942221/3	✉
Adomi PJ	951292/1	
Alomi SJ	952933/2	✉
Alable JG	951271/2	
Aninable S	941925/2	✉
Aomi AT	952581/4	
Aomi PS	951063/9	
Babl S	951027/X	✉
Bina SR	951442/9	✉
Baabl MC	950514/1	
Baomi H	950872/X	
Belable JM	950670/9	✉
Benina PS	940994/8	✉
Be BK	950282/1	
‘Be CJ’	951217/9	✉
Beelab MK	951500/8	
Bi RJ	951728/9	
‘Blomin BC’	951909/0	✉
Blenina J	952395/8	
Bdomi ME	941251/5	
Blomi AM	950556/X	
Blable LA	951559/9	
Brinable TR	951617/8	

Figure 8 The standard progress display. The names of the students have been scrambled for reasons of confidentiality.

Assessed problems are shown in burgundy (the darkest colour). The mark awarded determines how much the square is filled:

□ 2 marks out of 10

⬒ 5 marks out of 10

⬒ 7 marks out of 10

■ 10 marks out of 10

Late assessed problems show up in red:

🟥 Credit 0 marks.

The student files (named by surname and initials) can be ranked in many different orders: alphabetic, by student ID code, by progress through the course, by marks awarded, by ‘live seconds’ (time spent logged in) and by custom measures of ability derived from the event record.

The time-base display mode (Figure 9) is like the standard progress display except that each problem appears as a rectangle whose horizontal width is proportional to the time spent logged in and working on the problem, two pixels to the minute.

The time-base display mode allows the lecturer to observe the relative amount of time that students are spending on each of the problems in each tutorial. Note that the first ‘lead up’ problems in each set require less time than the later problems; this suggests that the 'live seconds' can be used to crudely rank problem difficulty even though some students are known to work the problems on paper while not logged in. An interesting footnote to these observations is that students achieving lower marks in formal examinations seem to require a great deal more live time per problem.

Since the serving software knows the current status of each student it is possible to quickly produce histograms, scattergrams and other charts representing the state of a selected group (or of all) students. One such display is shown in Figure 10. Along the top of this chart is the list of topics in the course, with the deadline for each. The lines separated by light and dark regions represent a 'snapshot' of the status of the class of 270 students on the deadline for each topic, when ranked by 'completed problem'. During the year this ranking was often used to gauge the pace of the course, since it was clear at a glance what fraction of the class was not keeping up.

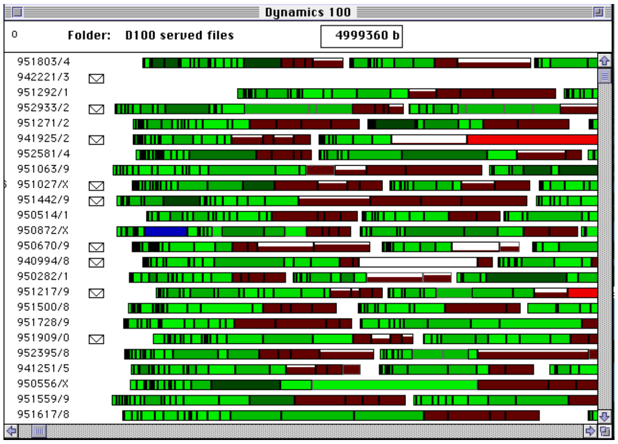


Figure 9 The time-base display.

4 Student response

In June and November 1995 students in Dynamics 100 were given an anonymous questionnaire with two parts: a multiple-choice part where assertions could be given an 'approval rating' from 'strongly disagree' to 'strongly agree'; and a written part where comments could be made under some given headings. Only the written comments are relevant to this work, since the idea of active class monitoring was not probed by any of the multiple-choice questions.

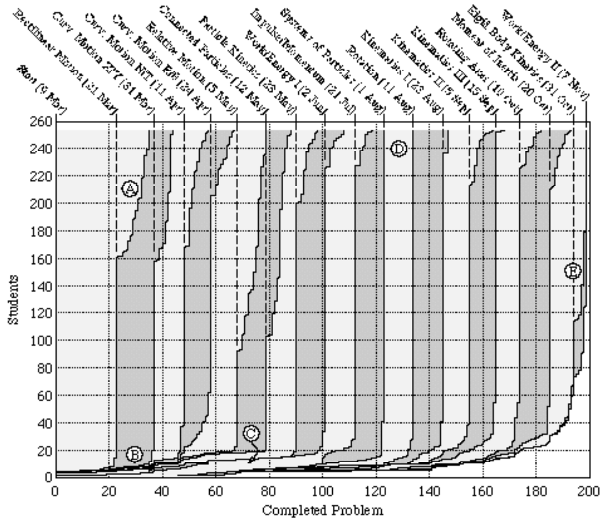
Students said, in response to the heading 'What aspects [of the course] do you feel are the best?':

> 'Weekly tutorials which are compulsory and worth marks help us learn more compared to Maths where the problem sheets are not checked, therefore leading to slackness in completing them.'
>
> 'Regular assessment means that you had to work!'
>
> 'The computer-based tutorials! Very fun and helpful. Makes a person do his work.'
>
> 'Though not being able to proceed until the question is answered is annoying at times, I found this useful as it forces me to work through the problems.'
>
> '{I'm} forced to complete all the questions i.e. {I} can't do the assessed problems until the other problems are completed; reasonable setting out of problems: progression from easy to difficult problems enhances learning.'

Students were also encouraged to write comments under the heading 'Do you have any suggestions for change', but the fact that their progress was monitored was not mentioned in this section.

In videotaped interviews the theme of monitoring was prominent. Students said:

> 'One of the troubles with "classroom" tutorials is that if you want to you can sit back and do absolutely nothing during the year, whereas doing the tutorials on the PC, the lecturers at all times know where the class is up to and they can come and have a talk to you if you're not putting the work in.'



A Students at the ‘top’ of the ranking forge ahead into the next topic before the deadline, while the majority of students usually work only as many problems as they must.

B Students near the bottom of the ranking are not up-to-date at the deadlines. Note the development of the low-ranked section of the chart during the year: the proportion of completely up-to-date students fell from nearly 100% at the start of the year to about 55% at the very end.

C The empty space indicated shows that about ten low-ranking student made a big effort to catch up – just before the June examination. In another paper we have shown that the quality of *not* leaving work too late is a fair predictor of success in this subject (Scott 1995).

D At this stage in the year’s work the problem stacks were being written ‘just in time’; so there was nothing for assiduous students to go on with until after the deadline for the previous topic.

E The last work for the year consisted of one short stack with only 4 problems. About half chose to finish this early, at the deadline for the previous topic; the other half were much less committed to finishing it at all.

Figure 10 Progress histogram.

> ‘Because your marks are recorded on computer and the lecturer and tutors can tell if you’ve been here or not, I find it more of an incentive. If you know that there is somebody who can tell, it makes you a bit more likely to come along.’

The most important ‘student response’ to a teaching innovation in engineering is competence as measured in formal assessment. In this course there were two written examinations: one in June, counting for 30% of the year’s credit, and one in November, counting for 50%. Naturally it is difficult to make direct comparisons between echelons of students, each having taken a different paper, but it was the opinion of the marking staff that the standard of competence demonstrated was *not lower* than in previous years.

5 Use of monitoring powers

The information that was constantly available at the serving computer informed teaching decisions on a daily basis.

- If it was observed that a deadline was approaching, yet most of the class was ‘behind’, the deadline would be put back a few days.
- At several times during the year students who were very far behind were contacted through the Dean’s office, with the result that some of them came in for personal help, when otherwise they would probably have simply

disappeared from view. It might be argued that this mechanism was unsuccessful since most students in this category failed the year anyway, but we believe that every student must be given the chance to recover.

Early in the year, after four weeks of the tutorials, a list of the ID Nºs of students who were 'behind' was shown in the lectures. The numbers were used because we did not want to embarrass the students, just to let them know that we were concerned.

- If a band of deep green appeared vertically in the progress display, we immediately investigated. In some cases this meant that the tutorial stack had a 'bug' which meant that the correct engineering answer was treated as incorrect. Since we were watching we could rectify the situation within a very few hours.
- The ability to plot measures of student performance such as the live time per problem against other measures has revealed several interesting correlations, one of which is reported in Scott (1995).

6 Discussion

Although every class consists of many separate students, there is an important sense in which the *group* can have an *attitude*. The reader has probably taught – or been part of – classes with both 'good' and 'bad' attitudes, and may also have experienced how quickly this can change if the teaching approach is 'wrong' in some way. We believe that this quality, which is difficult to define precisely, is one of the more important factors in a successful teaching innovation. We have an analogy from football: a class can be 'on-side' or 'off-side'. A class that is 'on-side' will listen patiently in lectures, and is likely to give the thumbs-up to nearly any reasonable teaching innovation; a class that is off-side will probably not bother to come to the lectures and will be disproportionately scathing when surveyed.

Prof. Burks Oakley II at UIUC set up a computer-based tutorial system which in some ways was the pattern for the one described here. His system was warmly received by students for several years – until one semester when the course was taught by a different staff member. The tutorial system continued to function correctly, but somehow the class became off-side, and the results from the survey were discouraging. The moral of this story is that a teaching innovation is not a separate entity in its own right: it is only meaningful to speak of '*this* system with *these* teachers and *these* students'.

With this in mind, the variables in this study are really:

1) The new tutorial system, including the part that monitors student participation; and

2) The particular teachers and students who used the system in a particular year.

We cannot claim that the monitoring approach we used *in itself* led to certain student behaviour and attitudes; we can only report that we did certain things and the students responded in a certain way. With this disclaimer, a superficial interpretation of the events of 1995 would have us saying that students appreciated the fact that we cared enough about them to 'watch' them, 'forcing' them to do the work. In a sense they admit that, left to their own devices, it would be difficult for them to give this subject an appropriate priority.

The reaction of the studied class to the monitoring was positive; the approach helped bring the students 'on-side'. This is not to say that the same type of monitoring applied to a different class would lead to the same result. More mature students, for example, might well feel that they ought to be allowed to choose whether to attend the tutorials or not.

7 Conclusion

Modern first-year university students are slightly self-destructive in that they *know* that a certain amount of work is required during each semester, but this knowledge alone does not necessarily prompt them to actually *do* the work. We have set up a tutorial system that keeps track of the progress of each student, we have used it to actually *chase* those who fall behind, and we have found that this feature is appreciated by the students.

The detailed and immediate information which our system offers has also proved invaluable in daily teaching practice.

8 Bibliography

Devenish DG, Entwistle RD, Scott N & Stone BJ (1995), 'An assessment package with diagnostic facilities', *Proceedings of International Conference on Computers in Education*, Selected Applications Track Papers, Singapore, December 1995, pp 101-107.

Ellison G (1992), 'Assessment methods in engineering degree courses: a discussion document prepared by the EPC Working Party on Assessment', *Engineering Professors' Conference Occasional Papers* Nº 5

Scott NW & Stone BJ (1995), 'Student behaviour near a deadline as a predictor of academic success', *Proceedings of the twelfth annual conference of the Australian Society for Computers in Learning in Tertiary Education (ASCILITE)*, The University of Melbourne, 4–6 December, pp 462–467.

N.W. Scott

Nathan Scott is a postgraduate student and research officer in the Department of Mechanical and Materials Engineering, the University of Western Australia. He is employed full-time writing software to illuminate engineering dynamics through presentation of moving examples. This work is funded by the Australian Federal Government through its national priority reserve scheme.

B.J. Stone

Professor Brian Stone has held the Chair in Mechanical Engineering at the University of Western Australia since 1981. He obtained his doctorate from the University of Bristol in 1968. He has been writing teaching software since 1987, some of which is now used at universities throughout the world. His research interests include vibration suppression and computer simulation of dynamic systems.