Full Paper submission for the 3rd Annual Conference on the Teaching of Computing.

Title: Managing Coursework:
Wringing the Stone, or Cracking the Nut?

Author: Dr. John T. Latham
Department of Computer Science,
The Victoria University of Manchester,
Oxford Road, Manchester M13 9PL
England

Email: jtl@cs.man.ac.uk

Phone: +44 161 275 6250

Fax: +44 161 275 6236

Managing Coursework: Wringing the Stone, or Cracking the Nut?

John T. Latham
Department of Computer Science,
The Victoria University of Manchester,
Oxford Road, Manchester M13 9PL
England
email: jtl@cs.man.ac.uk

August 1st, 1995

1 Abstract

We briefly describe ARCADE, a system which has successfully solved the problem of effectively and efficiently *managing* continuous assessment of all coursework undertaken by first and second year undergraduates in one of the UK's largest Computer Science Departments. The system is based upon a simple but effective regime of deadlines and their enforcement. It monitors and extrapolates the behaviour of all students in all of the modules taken, so that it can automatically give feedback (via email) to the students who need it. It collates individual exercise marks and computes a single mark for each student for each module. Thus it does much of the work of an overall laboratory manager, individual module managers and individual personal tutors. The system is suitable for use in any environment where the students have email access, with the coursework being of any nature and done in any medium. The paper concludes by claiming that there is much to be gained by automating coursework management.

2 Problem and solution context

In the current climate, there is much pressure on all of us to increase productivity, but naturally we worry about allowing standards to fall. There is evidence, or at least a feeling, that more students are arriving at university who need some considerable hand-holding, yet staff have less and less time to do so. This was, and is, the context for the system described in this paper. However, there was never a project with a noble aim, as such. At the outset, the author had just been (perhaps reluctantly) placed in charge of managing the entire first year laboratories, in addition, of course, to his existing duties. This appointment was necessitated by the start of proper semesterization and a major new syllabus: the finely-grained modularity of continuous assessment in the new syllabus made it impossible to 'manage' in the ad-hoc fashion previously enjoyed. The author was keen to use computers to maximize his efficiency in this job – hoping to make the machine do all of the donkey-work, leaving only the more sensitive issues to the Human. The system was developed from scratch during its first year of use (much on a 'night-before' basis) and the following year it was used, by another person, to manage the entire second year laboratories.

ARCADE is an acronym for Administering and Running Continuous Assessment with Deadlines and Extensions.

3 Problem scope and size

The problem is familiar enough:

- There are 220 to 250 students in each year.
- Most students take perhaps 12 modules, chosen from, say, 24.

- Each module typically has 10 to 16 hours of scheduled laboratory per student, with different groups of students scheduled at different times. By laboratory, we mean any form of continuous assessment work (including pen and paper based).
- The scheduled hours of laboratory for a module are divided into timetabled sessions, with compulsory attendance, and some with deadlines.
- The students are divided into groups for each module taken, and different groups have their sessions on different days.
- There are around 30 lecturers and 50 postgraduate students staffing the undergraduate laboratories, each with their own view, or small set of views, of the curriculum.
- There is one year manager for each year. A year manager may thus be required to coordinate and manage around 30,000 hours of student coursework in a year.

In addition to being able to manage this complexity, desirable features of any solution to the problem include:

- Consistency in administration and general organization across the many different laboratory modules. This is mainly to minimize the confusion of (especially) the students.
- The ability to easily monitor the whole progress of individual students, and automatically identify people who are struggling and/or missing. This is difficult if information is not centralized quickly.
- The regular provision of feedback to the students on their overall on-going *and predicted* performance – particularly to those who are falling behind. This can be very expensive in staff time, unless it is largely automated.
- Encouragement for the students to work consistently throughout the year, in both the scheduled sessions and in their own time, and with a sensible balance of effort between their different modules. Again this can be very expensive in staff time if not automated and/or inherent.
- An even spread of assessment effort over the year for the staff, rather than a huge mountain at a few times. This clearly links with the desire for regular feedback.
- The ability for a year manager to easily monitor the complete progress of individual modules, so that problems (such as an exercise being too hard) may be corrected immediately. This can be so time-consuming that it will not happen, unless the information is readily available in a summary and consistent form.

4 A discussion on deadlines

Clearly, some of the desirable features identified above can be achieved by having many deadlines throughout the year. This helps in spreading the load of assessment and encouraging the students to work evenly. It also provides the data for regular and timely feedback to the students. One of the traditional arguments against having many deadlines is the huge cost of their administration, and this is often used as the excuse for not having them. The development of ARCADE held the automation of deadline management as the highest priority.

Another argument often used against having many deadlines is that the students are adults and should be allowed to pace their own work. The author believes that when this isn't a disguise for the reason quoted above, it is a genuine view held from a rather naive and idealistic perspective. It is true that some students can be successfully treated in this way, but the author's experience is that the vast majority simply cannot. If, for example, one has a 10 week period with a deadline at the end and no other monitoring of progress, then the mode pattern of work will be to start in earnest in about the 7th week, and fail to complete the work. An alternative to many deadlines would be to introduce a regular monitoring of progress by a person that the student respects (e.g. his or her tutor). However, whilst this would be certainly less patronizing, it is simply unaffordable in most universities for first and second year coursework, whereas deadlines are relatively cheap.

Another argument against regular deadlines is that they are incompatible with the students undertaking a single large exercise or project. However, experience from Manchester suggests that even in this scenario, deadlines which are just ticks and are of the type to 'convince a demonstrator that you are N% through the effort', still work well to help ensure the students have a consistent pace (but of course, they do not spread the assessment load).

More evidence that students mostly work to deadlines is obtained from observing a typical laboratory session and deadline structure, of the kind previously used at Manchester for many years. In this we have a list of sessions with associated exercises which the students should in principle do at those times. However, each of these exercises has a deadline perhaps two weeks later. So, in effect, we give the students the same list of work twice, but with a two week difference in the dates. The author's experience observed over several years is that after a very short while the students simply ignore the first list and work to the second! This begs the question: why have two such lists? One answer is that it is just the obvious thing to do. Perhaps a more useful reason is that it can allow a single deadline to be set for each exercise, regardless of which laboratory group the student is in. This only works fairly if the distance from the last scehduled session to the deadline is at least a week, say, more than the spread of time over which the different groups have their scheduled session. Having a single deadline per exercise is a great saving of head-scratching in situations when there are many groups, but only in the absence of a machine managing the deadlines for us.

5 Deadlines and late flags

In ARCADE there is no distinction between the deadline and the scheduled session for an exercise. A module is taken by a number of students divided into groups. The work for a module is divided into a number of sessions. Each group is allocated a date (and time) for each of the sessions. Sessions may be given a deadline. This means that the deadline for the work associated with that session (and previous sessions since the last deadline) is the end of that session, for that particular group of students. However, an individual student has a right to obtain an extension of that deadline to the *start* of the next session, without any penalty. If he or she does not complete the work by the deadline, and either fails to obtain the extension, or fails to complete by the start of the next session, then that piece of work is flagged as being late.

Late work carries a heavy penalty, but one which inherently cannot cause a student to fail a module. Late work is marked in the normal way, but when the overall mark for the module is being calculated, the late and non-late portions are computed separately. If the non-late mark is already bigger than the pass level, then the late work is completely ignored! Otherwise, if the sum of the two is greater than the pass mark, then the pass mark is awarded. If neither of the above are true, then the overall mark is the sum of the late and non-late marks.

The author's experience of using this regime for two successive sets of first year students and one set of second year, is that it works very well in achieving the stated aims of encouraging the best work pattern from the students. They quickly realize that it is in their interests to do much of the work for an exercise *before* its scheduled session, so that they can make use of the laboratory staff should they need to.

There is, however, an axiom here that the only (normal case) reason for students to have a late flag is that they are not working hard enough. (There are mechanisms for dealing with exceptions – see the next section). This is how it should be, but sometimes those responsible for setting exercises are slow to accept that they are expecting too much! The author believes this to be a good feature of the regime: it (perhaps eventually) enforces the correct pitching of the difficulty of the work, rather than allowing a poor match to be hidden in the students' bad time-management.

6 Exceptions

In addition to the normal case described above, there are several mechanized exception provisions. A student may be excused for not attending the session, and this automatically grants him or her the extension, and excuses the non-completion of the previous session's work if appropriate. A student can be excused for not getting an extension, if for example he or she forgot to obtain one in the previous session. (This can only be done if the student was there in the session: one cannot have an extension for a session with an unexcused absence.)

A student can also be excused for not completing by the next session, for which there are two levels of excuse. An ordinary excuse can be awarded by a laboratory supervisor, but it only lasts for one more session, after which it automatically expires. A special excuse is one which does not automatically expire, (although it can be manually expired) and can only be awarded by the year manager. Typically, there are only a handful of students needing these, and it is believed to be a good idea for one person to be aware of all such special cases.

Excuses are stored and collated by student name to help identify those who have obtained more than their fair share of, for example, being let off oversleeping by lots of different laboratory supervisors!

7 Data collection mechanism

One might be excused at first for thinking that the data collection requirements of the regime described above make it prohibitively expensive. Certainly there is a measure of necessary bureaucracy, but the real question is whether the gain and the work saved outweighs this, and the author believes it certainly does. The gains are very high, and the level of administrative overhead is, in fact, less than most people expect.

The system produces a printed paper sheet for each laboratory session, containing the module name, the session name, the group name, and the name of each student who should be in that session, along with various columns for the data collection. For a session with a deadline, these columns are attendance, completion by deadline, extension, completion by next session, date of completion (for work completed even later) and marks. An example can be found in figure 1.

The whole set of sheets for a particular module are stored in a ring binder, and are organized by group. These sheets are completed by hand during the laboratory session (and at other times – for example after a bout of off-line marking). This is not onerous: in the vast majority of cases the data entered for a student is just a series of ticks and a mark. For example, one of the two most common cases is a tick for attendance, a tick for completion and then a mark. The other most common case is a tick for attendance, a blank for completion (later interpreted as a cross), a tick for extension, a tick for completion by next session, and a mark.

In between laboratory sessions, the hand-written data is entered on to the computer, which then prints updated versions of the sheets to be swapped for the old ones in the ring binders. Thus at the start of a laboratory session, when one is collecting work from the previous session, the laboratory staff are adding data by hand to a sheet which already has printed data on it.

For obvious reasons, a high priority in the development of the system was to optimize the user interface to the data entry program. In the first year of operation (whilst the system was being developed simultaneously) the author deliberately took it upon himself to enter all of the data, with the result that the program is easy and fast to use. The user is presented with a form similar to the paper sheet, and most data entry is by single key stroke. For example, a single '/' key stroke records attendance, the cursor then moves to the next column where another '/' records completion by deadline, so the program puts a hyphen in the extension column, a tick in the completion by next session column and a hyphen in the date column, and moves the cursor to the mark column. The user could then enter the mark (if it is available) and upon pressing the enter key the cursor moves to the next student. And so on. Additionally, the space bar continually changes its function so that pressing it causes the next most obvious thing to be done – for example, to move the cursor to the next place on the sheet where data can be added.

An additional feature is that the computer can read out a 3 digit identity code uniquely identifying the student over which the cursor lies. This code also appears on the data sheets, and thus much of the data entry can be reliably performed without the user having to look at the screen.

Final proof that the data entry interface is efficient and stress-free is that currently in the author's department, all the data is entered by two operators whose main job is to run all the student printer services. This *additional* role is carried out willingly, efficiently and seemingly effortlessly.

There are several reasons why the system development has so far steered away from direct data collection by computer from the student, or from the laboratory staff in the laboratory, and some are highlighted here in no particular order. One reason is the consciousness that computer resources are precious enough and should be used for the main job at hand: teaching. (For example, getting students to hand in an essay through a computer which they did not use to write it, is considered undesirable.)

1511 A+Ext+I+L page 2

Please do not write in black ink on this sheet

Year 1, Course 1511, Group A+Ext+I+L
Session 2D (20/10), Page 1 of 1 (24 students)

Name	Attendance on 20/10 (✓/×/E)	Completed by 20/10 (✓/×)	Extension to 27/10 (✓/×/-/E)	Completed by 27/10 (✓/×/E)	OR comp. date (-/Date)	Exercise mark (/10)
009(A1),Antarwillas,Zaibali F	✓	×	✓	✓	-	2
023(I3),Biset,Namanda	✓	×	✓	✓	-	6
028(A2),Bolder,David M	✓	×	✓	✓	-	14
032(I1),Braggs,Beth	✓	×	✓	✓	-	10
047(A1),Chan,Phong	✓	×	✓	✓	-	10
223(?),Coase,Martin	✓	✓	-	✓	-	13
053(I2),Cordell,Andrew J	✓	×	✓	✓	-	6
073(A2),Fung,Chi Wong	✓	✓	-	✓	-	10
075(I3),Garrison,Wilfred S	✓	✓	-	✓	-	10
078(A1),Good,Jonathon	✓	✓	-	✓	-	13
088(I1),Hardy,Mark I	✓	✓	-	✓	-	8
097(I1),Husain,Bhawin	✓	✓	-	✓	-	9
127(I1),Majil,Najma	✓	×	✓	S	9/11	7
137(I1),McGovan,Richard V	E	×	✓	E	?▷	-LATE
153(I3),Parsons,Kathryn J	✓	×	✓	✓	-	10
163(I2),Rajarat,Steve H R	✓	✓	-	✓	-	13
165(I3),Rehmurzi,Saweb	✓	×	✓	✓	-	8
182(A2),Shah,Alkesh B	✓	×	✓	E	3/11	8
183(A1),Shah,Nishma P	✓	×	✓	✓	-	10
184(I2),Sharpe,Colin A	✓	✓	-	✓	-	10
188(I1),Stone,Nicola M	✓	×	✓	✓	-	10
192(I2),Southman,Andrew	✓	✓	-	✓	-	10
205(I3),Waeed,Mustawa I	✓	×	✓	✓	-	9
248(?),Zieplowska,Paulina	✓	×	✓	E	9/11/94	10-LATE

Figure 1: Example laboratory session data-collection sheet

This is for a session occuring on the 20th October, with extensions until the 27th October and ordinary excuses expiring at the end of the 3rd November.

Another reason is the danger of a student being able to escape interaction with a member of the laboratory staff. The author believes such interaction is crucial to the well-being of the students, and the collection of data is a simple mechanism to ensure that every student gets spoken to. (In fact, it is very common for a student to tell you he or she is struggling whilst you are taking his or her attendance.) A third reason is that paper is more portable than a workstation! It is also generally easier to complete in a random order – the students do not form an alphabetically sorted queue to hand in their work, for example, and moving a pen in a hand is still quicker than moving a mouse pointer and then switching hands to the keyboard. More advantages include that one can use the same regime and method equally well regardless of the type of work the students are doing: essays, written exercises, programs, building circuits, and so.

8 Extrapolation of performance

A crucial feature of ARCADE is its ability to predict from an early stage which students are having difficulty, or are being lazy, and which are performing to a suitable standard. The approach used here is to compute an overall mark for each module for each student, right from the beginning of the semester or year. Then as the time goes by, the mark gets more and more accurate and by the end it is the final mark. This approximation is based on two dimensions: which pieces of future work will the student complete, and what the marks for them will be. No attempt is made to *predict* whether future work will be flagged late or not, as the only point of the extrapolation is to determine which students are in danger of failing and late flags cannot make a student fail.

Predicting whether or not the work which is not yet completed will get missed, is done as follows. If the work is already flagged late, we cautiously assume it will be missed. Likewise if it is excused, but the next session has passed (i.e. it is late, but not flagged as such). Otherwise, we predict that it will be missed if and only if the previous two pieces of work are either missed or predicted to be missed.

Mark prediction is even simpler: if the student has had no pieces of work marked for that module, the mark for each piece of work completed (but not yet marked) or predicted to be completed is assumed to be 70% of the maximum available. If the student has had one piece marked, the prediction is taken as the average of 70% and that mark. Otherwise, the prediction is taken as the student's average for those pieces that are marked for that module.

Although clearly very crude, this extrapolation of marks is remarkably effective in automatically spotting those students in danger of failing. All students start off with a predicted result of 70% (before scaling). Those, for example, who miss the first two deadlines suddenly get a predicted mark of zero. Those who miss one, and do badly in the next one may fall under the pass level, and so on.

9 Student feedback

Another crucial part of the system is the provision of automatic feedback to the students through email messages. Essentially, this involves the student being informed of all the 'irregularities' in their laboratory data: absences, lack of needed extensions, missed extensions, excused missed extensions (together with expiry date if appropriate), and so on. One of the most effective pieces of feedback is the inclusion of failure predictions for those students whose predicted mark is less than the pass level in some module. The feedback can also complain about the failure of the student to archive their work (e.g. for plagiarism detection) where appropriate.

The purpose of these messages is two-fold: to check that the student agrees with the data (it may be that the student has tried to give an excuse but it has not been recorded properly, for example), and/or to give the student the necessary prod to get their act together! The email messages are copied to the student's personal tutor to enable him or her to be made aware of the students difficulties.

An example message can be found in figure 2.

In previous runnings of the system, the feedback messages were sent to all students who had any irregularities, approximately once a week. However, in future they will be sent nightly to all those students whose list of irregularities has altered since it was last sent: this is more timely if, for example, a student has just given an excuse for absence, etc..

In addition, future runnings of the system will provide the laboratory staff with a printed copy of the irregularities pertaining to students in a particular laboratory group. This will appear next to the

```
Carefully worded general introductory message goes here, including a
reminder that the data could be wrong and the student should not worry if
this is the case, but just help correct it.
....
------------------------------------------------------------------------------
Notes: Each line of summary consists of your login name, your lab group, the
       module number, the session/exercise name, the date of the session and
       finally your "irregularities" with respect to that session.

"Absent" means you were not there at the session, and you have no excuse
    recorded.

"No extension" means you didn't ask for an extension, even though you hadn't
    completed the work.

"Done after extension" means you completed the work after the extended
    deadline (i.e. the start of the next session).

"Not done" means you haven't completed the work at all.

"Late" means that the marks for this work will not be added to your overall
    mark for the module once it has reached 40%.

"Done, but not archived" means that although you have handed the work in for
marking, or had it marked, you have not yet electronically submitted it as
requested. If you do not do so, you will ultimately lose the marks!

------------------------------------------------------------------------------
As you will see below, you have a warning that you are predicted to fail one
or more modules. This means that the prediction algorithm which tries to
extrapolate your performance up to this point in each module has projected
an overall mark below the normal honours pass level. This may be due to
missing work, or low marks. How seriously you take the prediction depends
upon how far through the laboratory schedule the prediction is made: after
only one deadline it typically contains a large proportion of guess-work,
whereas by the end it is entirely accurate (based on the data it has).
However, it is most useful early-on as an indicator of the likely outcome IF
your current performance is CONTINUED.
------------------------------------------------------------------------------
bloggsf Y 1011 Ex 3D (29/9) Absent, No extension, Done but Late.
bloggsf Y 1031 Ex 2D (20/10) Absent, No extension, Not done, Late.
bloggsf Y 1041 Ex 1D (10/10) Not done, Late.
bloggsf Y 1041 Ex 2D (17/10) Done, but not archived.
bloggsf Y 1041 Ex 4D (7/11) No extension, Done but Late.
bloggsf Y 1311 Ex 1D (27/9) Absent, No extension, Not done, Late.
bloggsf Y 1311 Ex 2.1 (4/10) Absent.
bloggsf Y 1311 Ex 3.2D (25/10) Absent, No extension, Not done, Late.
bloggsf Y 1311 Ex 5.2D (22/11) Absent, No extension, Done after extension,
                               Done (on 23/1) but Late.
bloggsf Y 1311 Ex 6D (29/11) Absent, No extension, Done after extension,
                             Done (on 23/1) but Late.
bloggsf Warning: on current performance you are predicted to fail 1311!
```

Figure 2: Example student feedback email message

data sheet for the next laboratory session, so that the laboratory staff can easily see what problems each student is having in their laboratory.

10 Management aspects

The structure of the laboratory is table-driven under the control of the year manager. The number of sessions, groups, deadlines etc., are configured at the start of the year or semester. Each deadline can be given an arbitrary maximum mark, as suits the marking scheme of the person who designed the exercise, and also an arbitrary integer weighting. The advantage of the latter is that during the year, or towards its end, the year manager might decide to alter the weighting of a particular exercise – perhaps because it was too hard, or too easy, without having to alter any of the individual student's marks.

ARCADE readily provides the year manager with many results and tools. For example, he or she can see a one-page summary of what needs doing at any time, so he or she can see that the laboratory is running smoothly (or not). To assist in the monitoring of how the various modules are going, the manager can see another summary showing the current percentage (predicted) failure rate and the number of students with irregularities, for each module.

The system collates all the (predicted) marks for each module into a single table, and produces a summary for each module and a histogram with an indication of the students' pattern of working. Figures 3 and 4 show the summary and histogram respectively for a module which many students found difficult.

At the end of the year or semester, the manager can request a single file containing all the overall laboratory marks for the year: each tuple in the file relates a student, a module, and a percentage mark. (This is provided so the laboratory marks can be sucked into another system which combines them with examination marks, etc..)

The system can also provide lists of marking outstanding, or work which requires demonstrating by the student, or work expected, etc..

11 Success of ARCADE

It is clearly difficult to be completely objective when measuring the success of a system such as this. However, ARCADE has been used for two years for first year work, whilst in the second of these being simultaneously used for second year work. The consensus of opinion of all the staff who have been involved in its use (module managers, laboratory supervisors and post-graduate demonstrators) are that it is a success.

To date, no effort has been made to actively solicit the opinions of the students on a wide-spread basis. The problem here is that the majority of students have no other experience of university education, and all of their coursework was managed under the same regime. A few students who changed from other departments have noticed the difference, but seem to accept it as fair. The only complaints brought to the attention of the author came from two students who were unhappy about the automatic dynamic scaling factor (a feature which can be enabled to even-out the overall module averages into an acceptable range). These students were unhappy because they wanted to get 100%!

One objective statement that can be made is the measure of actual laboratory session attendance. In the year completed just before this time of writing, the first year attendance was 89%, with an explained absence of 6%, and an unexplained absence of 5%. The corresponding figures for the second year students are 82%, 5% and 13%. Attendance figures for the previous regimes are not available, but those who were involved typically state their educated estimate of second year laboratory attendance in previous years as starting at around 75%, dropping instantly to around 40% and then moving to 100% in the last fortnight!

There were of course some problems with the new system. Firstly, some of the staff were sceptical that the bureaucracy was necessary. However, careful diplomacy persuaded them to reserve judgement, and they eventually became convinced of both the low cost and the high gain.

On a related issue, another difficulty was persuading the staff of the need for their laboratory data to be kept accurate and up to date. This teething difficulty resulted from a lack of their appreciation of the *power* of the system, and thus the consequences on student morale if data is wrongly recorded (e.g. an excuse not written down).

Course Summary for module 1001
Final dynamic scaling factor is 0.86

Deadline	1D	2D	3D	4.1D	4.3D	5.2D	Total	Marked
Weighting	10	10	10	5	20	20	75	
Maximum	10	10	10	5	20	20		
Name	1D	2D	3D	4.1D	4.3D	5.2D	Total	Marked
Done	179	180	174	171	168	113		
- by deadline	150	64	31	16	44	10		
- with extension	24	97	110	118	76	95		
- with excuse	2	12	11	7	19	2		
- late	3	7	22	30	29	6		
Still expected	0	0	0	0	0	0		
Missed	4	3	9	12	15	70		
Marked	179	179	174	160	168	113		
- no. >= 40%	179	179	172	163	162	104		
- no. < 40%	0	1	2	0	6	9		
- average (%)	98.3	86.6	92.2	95.2	86.1	68.5		
- minimum	4	3	3.5	2	2	4		
Unmarked	0	0	0	0	0	0		
Average (%)	96.1	85.2	87.6	88.9	79.1	42.3	59.91	63.77
No. >= 40%							168	
No. < 40%							15	
Late marks (%)	1.64	3.22	10.16	15.19	11.99	1.97	5.80	
Lost late (%)							3.86	

Figure 3: Example course summary of a 'difficult' module

The results show that as the module progressed, the number of students completing the work reduced, the time taken to complete increased, and the average marks, even for the work completed, reduced. This evidence would suggest that the module is too hard!

```
Histogram for module 1001

* = 1 student

 0- 4 |   3 | ***                                |   Work pattern
 5- 9 |   0 |                                    | 1 ################***.
10-14 |   0 |                                    | 2 #######***********=-
15-19 |   1 | *                                  | 3 ###************=--..
20-24 |   3 | ***                                | 4 ##*************=---.
25-29 |   1 | *                                  | 5
30-34 |   6 | ******                             | 6 #####********==---..
35-39 |   1 | *                                  | 7
40-44 |  33 | *********************************    | 8 #**********-........
45-49 |  10 | **********                         |
50-54 |   6 | ******                             |
55-59 |  18 | ******************                 |
60-64 |  17 | *****************                  |
65-69 |  10 | **********                         |
70-74 |  19 | *******************                |
75-79 |  23 | ***********************            |
80-84 |  26 | **************************         |
85-89 |   6 | ******                             |
90-94 |   0 |                                    |
95-99 |   0 |                                    |
100   |   0 |                                    |
```

Figure 4: Example histogram of a 'difficult' module

The bunching around the 40% mark shows the effect of the late flags in this extreme case. The work pattern clearly shows how, in this particular module, the students' ability to cope with the deadlines dropped off as the sessions went past. Each character in the work pattern of a deadline represents 5% of the students. # means completed by deadline, * means completed with extension, = means completed late with an excuse, - means completed late without an excuse and . means missed. Sessions 5 and 7 did not have deadlines.

Perhaps the most difficult problem was reassuring some of the post graduate demonstrators who were affected greatly by the *success* of the system. This was especially true for those who worked in the second year laboratory. The previous regime had few deadlines, and so the demonstrators were used to chatting and drinking coffee in half-full laboratories during most of a semester, but with a mad panic at the end. This was so radically changed – they were kept busy from the start – that some of them worried they would be totally unable to cope if the work pattern followed the same increase as previous years. Of course, it did not!

12 Future enhancements

Development of ARCADE is on-going, and there is still much to do. The author currently has a list of some 100 enhancements planned for the future. These vary from trivial to very complex.

For example, there is currently no special software for editing the configuration files describing the structure of the laboratories and the student details! These are stored in text files and carefully altered with a standard text editor.

A new feature, which has in fact just been added but not yet used, allows the manager to specify certain module sessions as having a 'Non-Registered' group. This contains all of the people who in fact are not registered to take that module, but who might suddenly decide to. This will be invaluable for the first few weeks of a new semester when the students keep changing their options: at least there can be somewhere ready printed to record their marks until things settle down.

Another planned feature, which is related to the above, is a mechanism for informing each student, again by email, of the details which the system database has about him or her. This would be sent automatically as it changes, so the student can check the accuracy. Experience has shown that students are not always quick to recognize the need to tell their department that they are intending to change modules, for example, and this leads to much confusion.

Smaller extensions to functionality include a desire to be able to record a query relating to a piece of work for an individual student. For example, when one suspects that the work was copied and the student is required to come and talk about it. The idea is that a query would be 'opened', and then later 'closed' when the question is answered. The system would keep track of and summarize all currently opened queries.

Perhaps the most complex intended extension is a provision for running unit-based laboratory courses. The idea here is that there would be a number of equally weighted units, or 'milestones' making up the total work for one module. In order to fit in with a deadline structure, the number of units would be a whole multiple of the number of laboratory sessions. Each deadline would require a student to hand-in at least some number of units, and at most some greater number of units. For example, a module might have 10 sessions, with 20 units. The minimum hand-in for each session would be one unit, the maximum would be two. The problem with this scheme is the management of which units correspond to which deadlines: this would be different for each student depending on his or her previous pattern of work!

It is certainly intended to tie-up ARCADE more closely with plagiarism detection software (currently it can be made to complain that a student has failed to proffer a piece of work for such detection as part of his or her feedback). Similarly, it would be very easy to integrate the system with automatic assessment software, although this is not considered as part of the current scope of the development.

13 Comparison with Ceilidh

At the start of the development of ARCADE, the author was not aware of the availability of any other generally applicable coursework management software. However, perhaps the most obvious apparent alternative is the Ceilidh system from Nottingham University ([Benford,1993]). Whilst the author does not have an intimate knowledge of Ceilidh (and sincerely apologizes if this relative ignorance leads to a misrepresentation in this paper), the two systems do appear to be completely different at first glance.

The most obvious difference is the order of priority of the developments. One gets the impression that Ceilidh started as software for the automatic marking of programs, then became able to manage programming laboratories, and then was generalized to be capable of managing other types of course. If we consider that by such development a system can eventually run an entire bounded world, then certainly one could observe the two systems starting at opposite corners and progressing across the floor of such a world. A concrete example of this difference is that ARCADE runs deadlines automatically (and they are different for the different groups of students, and can be systematically extended and further extended), but has absolutely no 'knowledge' of the actual work done by the students, except for the title of each exercise. In contrast, Ceilidh is given intimate knowledge of the exercises (in the sense that it can then mark them), but the deadlines are published as text rather than held as data, are the same for all students, and are enforced by the appropriate person actually pushing the 'deadline' button at the publicized time.

There must be a reason for this difference, and the author suspects it is simply a result of the project developers starting from different places. ARCADE was developed by a laboratory manager whose job it was to manage a whole year-full of different modules with different types of work associated with them. Whilst he was also involved in marking work in some particular modules, it was the management job he was trying to automate. One might expect an admission that the original motivation behind Ceilidh was the same in general – to reduce workload – but the development was started by someone trying to avoid marking programs, rather than avoid shuffling lists of names, chasing naughty students and adding up numbers!

In that sense, they are the same. And, despite being apparently different in detail, they both succeed because they take an existing manual system which works and automate a part of it in a way so that it still works, but works faster and/or with less effort.

This success is enhanced by another (subjective) factor which they both share: the 'big brother' syndrome. The theory here is that the students (by and large) work more efficiently than they might otherwise do, simply because they feel that their department is able to effectively monitor them, and they want to get good marks.

Another obvious difference between the two systems is that Ceilidh is further developed and more highly polished than ARCADE – Ceilidh started in 1988, has had many people working on it and has had some external funding. It is hence freely available and ready to use. By contrast, the developer of ARCADE is just one man, who does not even have a dog!

14 Conclusion

This paper has described a deadline regime and a supporting system for efficiently managing coursework undertaken by large numbers of students in a number of differently natured modules. It has also made the, difficult to substantiate, claim that the system is very successful. There certainly is the widespread belief among the staff of the author's department that the system not only manages the chaos, but simply smooths away nearly all of the administrative overhead of the various module managers, and at the same time, actually achieves a substantial increase in the work effort of the students.

ARCADE was designed to manage and monitor the work and marks of students independently of the kind of academic work undertaken. This common framework could be augmented with subsystems designed specifically for the support of certain topics (e.g. plagiarism detectors for programs, machine assessments, etc.) were appropriate. The author believes the system could, perhaps with some modifications, be used in other departments and even other disciplines. To start investigate this, it is likely that another Computer Science department will pilot its use in the forthcoming academic year – it will be interesting to see whether the success is an advantage of local idiosyncrasies, or really could be achieved elsewhere.

However, the point of this paper was not to sell ARCADE. The primary message is that computerized work management can be very cost effective and is an attractive option for any large department to consider. The author believes that the development effort he spent on the system is far less than the work already saved by his department in the system's short life so far, and is not much more than the work he has already saved himself as a laboratory manager. With increasing pressure to gain greater efficiency, we can feel we are forever wringing blood from stones. This, however, is an example of a stone which turned out to be tasty nut worth cracking.

References

[Benford,1993] Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Mohd Zin. *Early Experiences of Computer Aided Assessment and Administration when Teaching Com-*

puter Porgramming. Department of Computer Science, University of Nottingham, U.K..
Email: ltr@cs.nott.ac.uk