

The Ceilidh System

An Overview and some experiences of use

Eric Foxley, Colin Higgins, Edmund Burke, Cleveland Gibbon, Abdullah Mohd Zin

Learning Technology Research
Computer Science Department
University of Nottingham
NOTTINGHAM NG7 2RD, UK
email : ltr @ cs.nott.ac.uk
Computer Science Department
Universiti Kebangsaan Malaysia

Abstract

This document gives an overview of the state of the Ceilidh system as at Autumn 1996, and then summarises our experience in implementing, maintaining and using it during the last 10 years.

Ceilidh is a courseware system designed for the automatic assessment of student coursework in Computer Science and the administration of the corresponding courses. As far as we know, it is the mostly widely used such system in the world, having been taken by over 200 institutes of higher education in over 30 countries.

In this paper we discuss the main structure of Ceilidh, and then describe experience of its use in a number of establishments over several years. We end by outlining the future plans for development of the system, and describe the ongoing research to extend its areas of application and target audience.

Contents

- [1. The objectives of courseware](#)
 - [1.1. Functionality](#)
 - [1.2. History](#)
 - [1.3. Marking and assessment](#)
 - [1.4. Course administration](#)
- [2. Experience of the use of Ceilidh](#)
- [3. Educational considerations](#)
- [4. Administrative considerations](#)
- [5. The current state](#)
- [6. Current developments](#)
- [7. Evaluations](#)
- [8. Contributing universities](#)
- [9. Further reading](#)

1. The objectives of courseware

The major objectives involved in the teaching and learning process (whether performed by human teachers or computers) can be summarised as follows.

- The administration of courses. The system must keep track of the student coursework and marks, and assist in their collection and manipulation. This part of the system can be applied to courses

and assist in their selection and management. This part of the system can be applied to courses in any discipline.

- The assessment of student work. If the students can be assessed automatically, they can be left to work at their own pace, but can obtain feedback on their progress at all stages of the course.
- The passing of factual and experiential information to students.

1.1. Functionality

We now expand on the above objectives, and illustrate more specific examples of the activities we consider necessary to achieve the objectives.

- (i) The administration of courses

Under this heading we include

- Monitoring individual student progress
- Monitoring overall course progress
- Giving summary feedback to the teacher
- Informing tutors of relevant information
- Detecting defaulting students
- Keeping copies of student coursework
- Detecting plagiarism in student work

- (ii) The assessment of student achievement

We wish to be able to automatically mark student work in various forms. This would allow learners to proceed at their own pace, and to be given immediate feedback as they progress. The objectives set for the Ceilidh system were to mark work submitted as:

- computer programs
- in a variety of programming languages
- multiple choice questionnaires
- question/answer exercises
- single word answers
- single sentence (natural language) answers
- essays or reports

These student solutions may arise as coursework, or as part of an (on-line) examination.

- (iii) The presentation of information to students

This corresponds to the lecture in typical courses. The traditional role of computers in education has been in the presentation of information to a student, with the speed of progress determined by the student. The pace at which the course material is presented, and the order in which the topics are undertaken, are at the student's discretion. The assessment in most such courses is relatively straightforward, and is at the discretion of the student.

The Ceilidh project aims eventually to cover all of these areas, but at present covers only the first two. The third area is being actively pursued as part of the current project, but currently available versions of the system give administration and assessment support for a course of lectures in various Computer Science subjects.

It is interesting to note that the areas of administration and assessment both generate tasks which increase in proportion to student load, and can absorb significant quantities of staff effort in a large department. The third area above (delivery of information) does not necessarily increase in proportion to student load, especially if it involves large class teaching: to lecture to a class of 300 is little more

to student loads, especially if it involves large class teaching, to lecture to a class of 500 is much more time consuming than lecturing to a class of 100. We should therefore put our system development effort into providing computer assistance in the first two areas.

1.2. History

The Ceilidh system has been in use at Nottingham since 1988. An early version then assisted in the teaching of both C and C++ courses with class sizes of up to 200 students. The major current version runs on UNIX systems.

Its success during this period resulted in funding from Nottingham University and the UK government to enable a consortium of universities to extend and distribute it. It is now distributed to about 200 sites in 30 countries on all the continents. It runs on (almost) any version of UNIX.

1.3. Marking and assessment

The original version of Ceilidh was developed at Nottingham mainly for marking programs written as coursework for a C programming course. The student would (on-line)

- (i) read this week's question (a file of ASCII text describing what the program has to do, and the format of input and output required);
- (ii) obtain a skeleton outline of the solution program (and any associated header files etc);
- (iii) develop a solution program; and
- (iv) submit the program for marking.

The last two steps could be repeated, so that the student could have several attempts at a solution; after each attempt the system provided feedback about the weaknesses in submitted work. The number of repetitions would be subject to an upper limit set by the teacher.

The marking process evaluates both the static and dynamic properties of the student's program using a number of specialised metrics. The assessment includes marks related to the following.

Static:

program layout
 indentation, choice of identifiers, readability metrics
 use of comments
 program structure
 use of denotations, complexity metrics
 "lint"-type warnings
 suspicious constructs

Dynamic:

the program is run against several test data sets
 or driven using several shell scripts
 the program output is then validated
 using an oracle
 efficiency can be monitored

[An *oracle* is a program which looks for meaning within textual information. For the definition of an *oracle* see E. W. Weyuker's [\[ref 1\]](#) E. W. Weyuker, *On Testing non-testable program*, The Computer Journal, 1974, 17(1), 1-10.

Journal 25; 4; (Nov 1982) pp 465-470 original paper, or the Ceilidh document [\[ref 2\]](#) [\[content\]](#) describing our extended oracle program, available on the WWW.]

Originally (when we were running on a small PDP 11 with relatively little processing power) the marking was done overnight; the student submitted the work during the day, the marking was done (and results emailed to the student) overnight. This implied at most one attempt per day.

Advances in hardware now permit us to mark interactively, providing instant feedback to the student on the mark awarded, and on the major areas where they have lost marks. The marks are archived, and made available also to the course teacher and the student's tutor. In addition a copy of the student program is stored for future reference and possible re-evaluation.

1.4. Course administration

Extensions to the original version of Ceilidh then necessitated the provision of tools to assist the teacher more in the administration of the course, and the development of new techniques to broaden the scope of the student activities we cover. Marks generated by staff can now be entered by hand; these marks may be either amendments to existing marks (overriding the computer assessments), or marks for associated work (such as essays or reports) which would be marked by hand.

In addition work in the form of essays/reports can be submitted on-line by the students (having been generated using a word processor), stored on the system, marked by hand on or off the machine, and the marks then entered by hand. The essays are sometimes part of a programming exercise, and at other times are independent exercises in themselves.

The teacher can then look at the mark statistics for a class, exercise or student, find who hasn't submitted (and perhaps email them and/or their tutors), look at overall class program metrics, and check for plagiarism in submitted work. The overall metrics for a given exercise are useful in keeping the teacher in touch with the current performance of the class as a whole; this is more important when the teacher is not hand marking student work.

The plagiarism metrics for a single exercise may not be significant; it may be a co-incidence that a particular pair of programs are very similar. The plagiarism pattern over a series of exercises can be significant; in general the known presence of plagiarism tests acts as a considerable deterrent to copying. The plagiarism system shows the exact dates and times of submission of programs involved in plagiarism. The later program is not necessarily copied from the earlier, but the common feature that one of the pair was submitted less than 20 minutes before deadline gives a hint of the onset of panic!

A tutor can look at the progress of tutees, and look at their submitted work. We find it better to email summaries regularly to the students' tutor rather than expect the tutor to log in and look it up!

2. Experience of the use of Ceilidh

Experience of use over a number of years at several sites has produced the following observations.

Modifications to the style

- It has proved easy to modify the courses as distributed, to change them and add new features. Different institutions use Ceilidh in different ways (some do not use skeletons at all, while some use almost complete programs as skeletons; some use many small exercises, others set fewer larger problems).

Simpler marking

We have generally moved to simpler marking metrics as experience develops. Originally the thought

- we have generally moved to simpler marking metrics as experience develops. Originally we thought was that, to mimic human marking, we must have a very complicated marking process involving equivalents of all the human marking processes we could find. We now think that it is better for the students to have much simpler objectives; and it must be possible for the teacher to both explain and justify these objectives to the class. It is implicit therefore that the teacher understands the marking process; this is a non-trivial task for new users.
- There were originally six main marking tools (dynamic correctness, dynamic efficiency, typographic layout, program structure, program complexity, and program features). We now concentrate on three (dynamic correctness, typographic layout, and program features). This gives the students much clearer objectives to aim at.

The use of Ceilidh marks

- The marks produced by Ceilidh can be used in different ways. Some courses are 100% ceilidh assessed; the marks produced by Ceilidh need to be scaled to fit with the institution's generally accepted borderlines. Facilities are available to do this.
- Ceilidh marks can be used in other ways:
 - Ceilidh marks count for 40% of the total, a traditional examination counts for the remaining 60%.
 - Ceilidh is used to mark the dynamic tests only, but the program sources are hand marked.
 - Ceilidh marks all the coursework, and and the any student on a borderline has their work hand-marked.

All of these save considerable human effort when compared with traditional marking systems.

The use of essays.

- We make much more use of essays now. These are intended so that the student can demonstrate the thinking behind their program design and testing.
 - The skeleton essay is provided by Ceilidh when the student starts a new exercise. command, in addition to the skeleton program. The skeleton essay contains section headings, to ensure that their explanation covers the areas I am interested in.
 - The essays are collected by Ceilidh as part of submission process, and stored in the Ceilidh solutions area.
 - The Ceilidh submission/marketing process can then be extended to include a simple spelling check by Ceilidh (this is non-trivial, and is worthy of considerable development effort). The spell check result has only a very small weighting, but the students object "You didn't tell us that we had to spell correctly!".
- The essays could be marked on-line, but in general we use Ceilidh to print a single document for each student which can be marked off the machine. The document typically contains:

The essay
 The other Ceilidh mark details
 A list of the actual spelling errors
 A copy of the program source

The essays are then marked by hand.

Innovative marking.

- Many other possibilities arise for student assessment in Ceilidh.
- We have a new "Software Tools" course, which teaches general software principles. It is a challenging new course requiring a new approach to marking. A typical exercise might be to "write a

print spooler". The students are given the scheduling objectives. We assess their solutions using a simulation system to see whether they achieve the scheduling objectives set out. Other exercises involve Inter Process Communication (IPC) techniques; these can be marked by driving the programs from a shell script. We may need to start several processes to run concurrently, and observe the resulting message passing.

- For teaching the use of publishing tools, we need to be able to assess graphical (ASCII or PostScript) output to check that they are inserting headers and footers as specified.
- This course also included units on the Unix Bourne shell and on "awk" programming. These were both treated as interpreted programming languages, and tested using techniques related to those for compiled languages.

Other ways of marking

- For some exercises, the students are required to design and submit a set of test data for the problem under consideration. We then (outside Ceilidh) compile and run all of the student programs against all of the sets of test data. The students get a better mark if

their test data causes many of the other programs to fail
their program can handle all sets of test data

Collection of coursework

- The Ceilidh system can be used purely as a means of collecting and administering coursework completed on-line, such as essays. A number of courses at Nottingham use the system purely to collect coursework, which is then printed and marked by hand. The ability to check on absentees is useful, and is the consistent quality of the printed essays.

Miscellaneous

- It has been possible for a long time for tutors to use Ceilidh to look up the progress of their tutees. However it was deemed a significant development when it was arranged that Ceilidh emailed summaries to all tutors once a week; tutors can now be reactive rather than proactive.

3. Educational considerations

Ceilidh is unique in the way that it automates the assessment process, and provides instant feedback to the student as well as general guidance and course materials. In the "teaching, learning through practice, assessment" cycle we can now offer a more thorough "learning by doing" activity without requiring unreasonably large amounts of staff time. With such practically oriented disciplines as programming, this represents a much better approach.

In this section we will look at the educational implications of the essential student process under Ceilidh.

The student reads the question

- The question must be complete, non-ambiguous and consistent. This is surprisingly hard, even in relatively simple examples. Floating point calculations cause particular problems concerned with rounding and printing. A simple exercise, hurriedly implemented after a trip abroad, involved currency conversion between two currency denominations. The basic formula is simple:

$$\text{Local} = \text{Distant} * \text{ConversionRate} - 1.5\% \text{ commission}$$

At which points does the rounding occur? The question MUST be specific. When the question has been

made very specific and detailed, it sometime becomes quite hard for the student to read and understand it. In general, the more specific the question, the easier the marking process.

The student "setup"

- The course developer decides how helpful the skeleton is to be. Some teachers don't allow any skeletons (the students must always start from scratch). We usually just miss out a few key sections of the model solution which which involve this week's topic. We believe that in industry programmers usually start from a skeleton of some sort. Some developers give complete working examples as a skeleton, for the students to explore.
- As well as a skeleton program, it may be useful for Ceilidh to set up:

- skeleton essay (section headings)
- some test data
- a copy of the question
- any other helpful files, useful for remote users

The development (edit/compile/run) cycle

- Nowadays many PC environments (Borland etc) are excellent for program development. Future versions of Ceilidh will make it an option to leave Ceilidh for this stage. This relieves the load on the central system, although there may be incompatibility problems between compilers.

Submission

- The use of the environment variables MAXSUB (the maximum permitted number of submissions) and MINGAP (the minimum time interval between submissions). The use of a large value for MAXSUB enables student to work towards a standard; bright students if they so wish can work to the best grade (grade A), others can work to achieve a pass grade (grade C) only. Students are then encouraged not to spend further time once they have reached their chosen standard.
- A small MAXSUB or a large MINGAP makes student think more carefully before re-submitting.
- With a small MAXSUB the results show student's ability to be right first time the student's ability to be right first time; it can be argued that they are learning to play the Ceilidh marking system better. In the Mathematics courses at Nottingham we set one exercise each week with large weight but only one submission (this assesses their ability) and others each week with a small weight but several submissions (this offers the opportunity for more personal development).

Introducing Ceilidh into a department

- It is vital that anyone planning to use the system should be familiar with the marking metrics, and should adjust them to suit local standards. Any teacher *must* be able to justify the marks awarded.

4. Administrative considerations

For legal and moral reasons to make the system acceptable to students, we think that

- the student must be able to see how the marking works; and
- the student must be able to see data about them stored in the system (including marks and submitted programs and essays).

For academic reasons at our institution we must keep an archive copy of the complete course including solutions and marking schemes for

- an external examiner to check borderline candidates at later examiners' meetings; and
- in case of an appeal by a student at a later stage, when we would need to be able to produce copies of all that student's work.

To complete the archive, the system keeps copies of all *messages of the day* and of the dates and times of all exercises opening and closing instructions.

Ceilidh can keep audit trails of

what a given student is doing
who is using particular Ceilidh facilities

We have found this useful in monitoring monitor defaulting students, and in analysing the used of Ceilidh.

Courses completely assessed by Ceilidh

Some early C and C++ courses at Nottingham are completely assessed by Ceilidh. The results of the Ceilidh exercise assessments are weighted, scaled and submitted as the returned mark for the module. There are certain implications when this happens.

- (i) A complete dump of the file system is taken at the end of the course, to be made available to external examiners, or in the event of an appeal. Any later changes to the on-line information will not affect the archive.
- (ii) The marks and submitted programs must be made available to the student. The students cannot then claim that "That isn't the version I submitted". There are two Ceilidh commands to assist in this: **cks** allows students to check all of their submitted marks and programs as seen by the system; and **vm** lets them view their marks, complete with weighting and scaling factors which will be applied as part of the examining process.

5. The current state

The final Nottingham release is available to authorised parties over FTP, and contains the following material.

Unix Ceilidh

The final Unix version of Ceilidh (release 2.5) has a number interfaces.

Interface	Developer
–	
Dumb terminal menu interface	Nottingham
Dumb terminal command line interface	Nottingham
X-windows interface	Nottingham
<i>Developed, not distributed</i>	
Macintosh Client, Unix server	Loughborough
WWW	Nottingham, Ngee Ann (Singapore), Cardiff, Rotterdam

The WWW version can be tried by going to the page

<http://www.cs.nott.ac.uk/~ceilidh>

Courses

The courses available with Unix Ceilidh are as follows.

Course	Developer
–	
C	Nottingham
C++ nuts and bolts	Nottingham
C++ programming in the large	Nottingham
Pascal	Royal Holloway University
SML functional language	Heriot Watt University
SQL	Liverpool University
<i>Developed, not distributed</i>	
Fortran	Helsinki University of Technology
Software Tools	Nottingham

A statistics package is available for viewing details of course statistics.

6. Current developments

The major development currently taking place at Nottingham University (UK) is the production from scratch of revised enhanced versions of Ceilidh and its courses as follows. The dates given are approximate.

- 1998 One or more C++ text books with single user Ceilidh on CD-ROM in the book.
- 1999 Pascal and C text books with Ceilidh on CD-ROM
- 1999 Multi-user secure language independent versions of Ceilidh for UNIX and PC networks.
- 2000 A freestanding CD-ROM with a complete hypertext C++ course.

Some parallel developments are taking place at other universities in a variety of countries, and there is a number of on-going research projects. Other suggestions and contributions would be very welcome. New courses involving new marking tools are being researched as follows.

Course	Developer
–	
FORTTRAN	Helsinki University of Technology
Modula 2	Ulster
ADA	Lancaster
Software Engineering	Lancaster
Prolog	Nottingham
Software Tools	Nottingham
Shell programming	Nottingham
Awk programming	Nottingham

Formal Specifications in Z Nottingham
Digital Circuit Design Nottingham

7. Evaluations

Questionnaires completed by the students (every course has an on-line questionnaire as the last "exercise" of the course) do not necessarily produce unbiased results. The results analysed so far indicate that they agree overwhelmingly system is helpful.

In addition to the questionnaires, we have had discussions with students at the end of each course. The general impression given is that they find the system very supportive.

Some students express concern at having difficulty in finding the last few percentage marks to raise their total from the mid-nineties to the high nineties. Others say that they use Ceilidh to develop their mark up to a certain figure (80%?) and then leave the problem completely.

8. Contributing universities

Ceilidh is the result of collaboration from a number of universities. Their help is appreciated, and is hereby acknowledged.

Institution	Contribution	Available from Nottingham?
—		
Lancaster	PC version	y
Heriot Watt	SML course	y
	Evaluation	n
Manchester Metropolitan	PC hypertext system	y
Liverpool	SQL course	y
Royal Holloway	Pascal course	y
Luton	Evaluation	n
Loughborough	Macintosh interface	n
Sheffield Hallam, Royal Holloway, Nottingham	X interface	y
Nottingham, Ngee Ann (Singapore)	WWW Interface	n

9. Further reading

Further details of the Ceilidh project are available on the WWW at

<http://www.cs.nott.ac.uk/~ceilidh>

Recommended support reading material includes the following Ceilidh documents (distributed with Ceilidh, and available from the WWW page above). The guides give a feel for the system: the Student Guide [\[ref 3\]](#) [\[content\]](#), the Tutor Guide [\[ref 4\]](#) [\[content\]](#), the Teacher Guide [\[ref 5\]](#) [\[content\]](#), the Developer Guide [\[ref 6\]](#) [\[content\]](#) Steve Benford, Edmund Burke, Eric Foxley, *Course Developer's Guide to the Ceilidh System*, LTR Report, Computer Science Dept, Nottingham University (January 1994). Other general Ceilidh documents include the Automatic Program Quality Assessment [\[ref 7\]](#) [\[content\]](#)

paper, and the Design Document [\[ref 8\]](#) [\[content\]](#). For these and further documents, see the WWW Ceilidh page referenced above.

The following journal papers describe new aspects of the system: from the Software Quality Journal [\[ref 9\]](#) **Steve Benford, Edmund Burke, Eric Foxley**, *A System to Teach Programming in a Quality Controlled Environment*, The Software Quality Journal (1993) 2; pp 177-197, the Association for Learning Technology Journal [\[ref 10\]](#) and the Journal Education Technology [\[ref 11\]](#) **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Ceilidh as a Course Management Support System*, Journal of Educational Technology Systems 22; 3; (September 1993)

The following conference papers summarise various aspects of Ceilidh. [\[ref 12\]](#) **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Experience using the Ceilidh System*, Proceedings of the All Ireland Conference on delivering the Computer Curriculum, Dublin (September 1993), Amsterdam [\[ref 13\]](#) **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Integrating Software Quality Assurance into the Teaching of Programming*, Proceedings of the CSR 10th Annual Workshop "Applications of Software Metrics and Quality Assurance in Industry", Amsterdam (1993), CBLIS at Vienna [\[ref 14\]](#) **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Mohd Zin**, *Ceilidh: A course administration and marking system*, Proceedings of the International Conference on Computer Based Learning in Science, Vienna (December 1993) and [\[ref 15\]](#) **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Mohd Zin**, *Experiences with the Ceilidh System*, Proceedings of the International Conference on Computer Based Learning in Science, Vienna (December 1993), York [\[ref 16\]](#) **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Mohd Zin**, *CEILIDH - a management and marking system*, Assessment in Higher Education Computing, York (June 1993) [\[ref 17\]](#) **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Ceilidh as a Course Management Support System*, Journal of Educational Technology Systems 22; 3; (September 1993)

[\[ref 18\]](#)

References

- 1: **E. W. Weyuker**, *On Testing non-testable program*, The Computer Journal 25; 4; (Nov 1982) pp 465-470
- 2: **Abdullah Mohd Zin, Eric Foxley**, *The Oracle program*, LTR Report, Computer Science Dept, Nottingham University (1992)
- 3: **Steve Benford, Edmund Burke, Eric Foxley**, *Student's Guide to the Ceilidh System (2.4)*, LTR Report, Computer Science Dept, Nottingham University (1995)
- 4: **Steve Benford, Edmund Burke, Eric Foxley, N H Gutteridge, A Mohd Zin**, *Tutor's Guide to the Ceilidh System (2.4)*, LTR Report, Computer Science Dept, Nottingham University (1995)
- 5: **Steve Benford, Edmund Burke, Eric Foxley**, *Teacher's Guide to the Ceilidh System 2.4*, LTR Report, Computer Science Dept, Nottingham University (1995)
- 6: **Steve Benford, Edmund Burke, Eric Foxley**, *Course Developer's Guide to the Ceilidh System*, LTR Report, Computer Science Dept, Nottingham University (January 1994)
- 7: **Abdullah Mohd Zin, Eric Foxley**, *Automatic Program Quality Assessment System*, Proceedings of the IFIP Conference on Software Quality S P University, Vidyannagar, INDIA (March 1991)
- 8: **Steve Benford, Edmund Burke, Eric Foxley**, *The Design Document for Ceilidh Version 2*, LTR Report, Computer Science Dept, Nottingham University (1993)

- 9: **Steve Benford, Edmund Burke, Eric Foxley**, *A System to Teach Programming in a Quality Controlled Environment*, *The Software Quality Journal* (1993) 2; pp 177-197
- 10:
- 11: **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Ceilidh as a Course Management Support System*, *Journal of Educational Technology Systems* 22; 3; (September 1993)
- 12: **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Experience using the Ceilidh System*, *Proceedings of the All Ireland Conference on delivering the Computer Curriculum*, Dublin (September 1993)
- 13: **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Integrating Software Quality Assurance into the Teaching of Programming*, *Proceedings of the CSR 10th Annual Workshop "Applications of Software Metrics and Quality Assurance in Industry"*, Amsterdam (1993)
- 14: **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Mohd Zin**, *Ceilidh: A course administration and marking system*, *Proceedings of the International Conference on Computer Based Learning in Science*, Vienna (December 1993)
- 15: **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Mohd Zin**, *Experiences with the Ceilidh System*, *Proceedings of the International Conference on Computer Based Learning in Science*, Vienna (December 1993)
- 16: **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Mohd Zin**, *CEILIDH - a management and marking system*, *Assessment in Higher Education Computing*, York (June 1993)
- 17: **Steve Benford, Edmund Burke, Eric Foxley, Neil Gutteridge, Abdullah Modh Zin**, *Ceilidh as a Course Management Support System*, *Journal of Educational Technology Systems* 22; 3; (September 1993)
-

Notes converted from *troff* to HTML by an [Eric Foxley](#) shell script, email errors to me!