The CourseMaster Automated Assessment System – a next generation Ceilidh

Eric Foxley, Colin Higgins, Pavlos Symeonidis and Athanasios Tsintsifas
University of Nottingham
ltr@cs.nott.ac.uk

Abstract

While teaching in all its forms can sometimes be fun, for most people marking student work is tedious, boring and in general hard work. A courseware system is presented that not only provides on-line support for courses, but importantly can automatically assess student work.

CourseMaster can mark several types of coursework in a non-trivial manner. That is, criteria can be set against which the work is thoroughly assessed. Specifically, the system is particularly good at marking computer programs in several languages including Java and C++. It can also mark diagrams and assist in the marking of essays. The marking of multiple choice questions is also supported.

In addition to marking, the system also supports the provision of lecture notes and web pages and links. It can be used to collect any on-line work and enforce deadlines. Finally, it provides a suite of web based tools that allow the easy management of courses. CourseMaster has been used "live" at Nottingham with great success for two years and is based on the earlier Ceilidh system which was developed and used for over the preceding ten years.

1. Introduction

A courseware system is presented which not only facilitates teaching and marking, but also enhances the student's learning experience. Courseware consists not only of a suite of material from which students can learn, but also the delivery mechanism for providing this material and for administering the running of courses. In particular the material presented to the students might consist of lecture notes (in a variety of formats from raw text to www multi-media documents), tips and guides, messages to the students (e.g. deadlines and assignment dates), diagrams, animations of algorithms etc. and, of course, exercise questions and assignments to be attempted. Administration requires the collection of registers and assignments, monitoring student and cohort progress, giving feedback and reporting marks etc. Ideally, courseware also includes tools to assist in the assessment of the course; perhaps the most important aspect and certainly the least readily available elsewhere.

Teaching can be fun! Administration can be delegated. However, marking is boring and assessment in general is probably the least liked task of most academics! Hence, whilst the material, delivery and administration systems are an important aspect of courseware (especially in distance learning environments), running a course can be done without them. Many other courseware systems now exist for the production and dissemination of course material to students. Not only was this not the case when the Ceilidh was first developed at Nottingham, but the most important aspect of courseware is still underdeveloped today, that of *automatic assessment*. Thus, although Learning Technology Research group (LTR) is involved in the material and administration aspects of courseware which are highly important in providing a stimulating and enriched learning environment and in supporting the teaching of a course, the main concern is with automatic assessment of student work, and in particular the marking of student programs, diagrams and essays.

The LTR group at the University of Nottingham has been working in the area of automatic assessment for thirteen years. This work has resulted in many tools and papers [1], the major products of this work being the Ceilidh system [2, 3] and more recently the Ceilidh CourseMaster System, referred to as CourseMaster (CM) for short [4]. Importantly, these developments have been used to teach active courses to real students from the outset, resulting in practical, tested, working systems. The feedback from this use has resulted in improved systems incorporating great experience, particularly of automatic assessment.

This paper first describes and reviews the old Ceilidh system and how the new software, CM, builds on the experience gained from it. A detailed description of the CM system follows, including its functionality, design philosophy and usage. Finally our experience of using CM for the last two years is described and conclusions are drawn.

2. The Ceilidh System

Ceilidh is a collection of programs and tools written in a variety of languages such as shell, C and awk which execute in a Unix environment [5]. The first system was used in 1988, and as a result of the continuous incorporation of enhancements suggested by users over the years it has become increasingly difficult to understand, maintain and support. The three courseware dimensions (presentation of material, administration and assessment) are provided in a variety of ways. Material is provided as a set of text files and web pages. Administration is enabled via active web pages and assessment can take a variety of forms. The most powerful form of automated assessment is the automatic marking of programs in a variety of programming languages [6] which is unique to Ceilidh, given the depth to which it can mark. In addition, Ceilidh can mark multiple choice questionnaires and can be used to collect any on-line coursework. A set of tools gives a human marker grammatical and spelling measures to assist in hand marking of essays.

Five roles of users are represented within the system, with each role having progressively more access rights and authority. Firstly, there are the students who can browse notes, read questions, look at the message of the day, "setup" a solution (download a skeleton file etc.) and submit a solution. Solution development is performed externally to the system via a Programmer's Development Environment (emacs in our case). Secondly, there are tutors/lab assistants who in addition can look at model solutions, read tutor help files and monitor any student's progress. Thirdly, there are teachers of a course who in addition can chose which exercises to use and when to set deadlines. Fourthly, course developers can build exercises with their detailed mark schemes and enter new material. Finally, there is the system administrator rôle where new users or courses are added and general maintenance of the system performed. Numerous other rights and responsibilities are allocated to each of these rôles as well as those mentioned here.

Ceilidh was used widely with around 15 different programming languages. While successful, it had several disadvantages. For instance it was based around the Unix operating system and required knowledgeable system staff to install and maintain. Assessment was excellent, but feedback to the students was limited. The user interface was for many years based on ASCII character terminals.

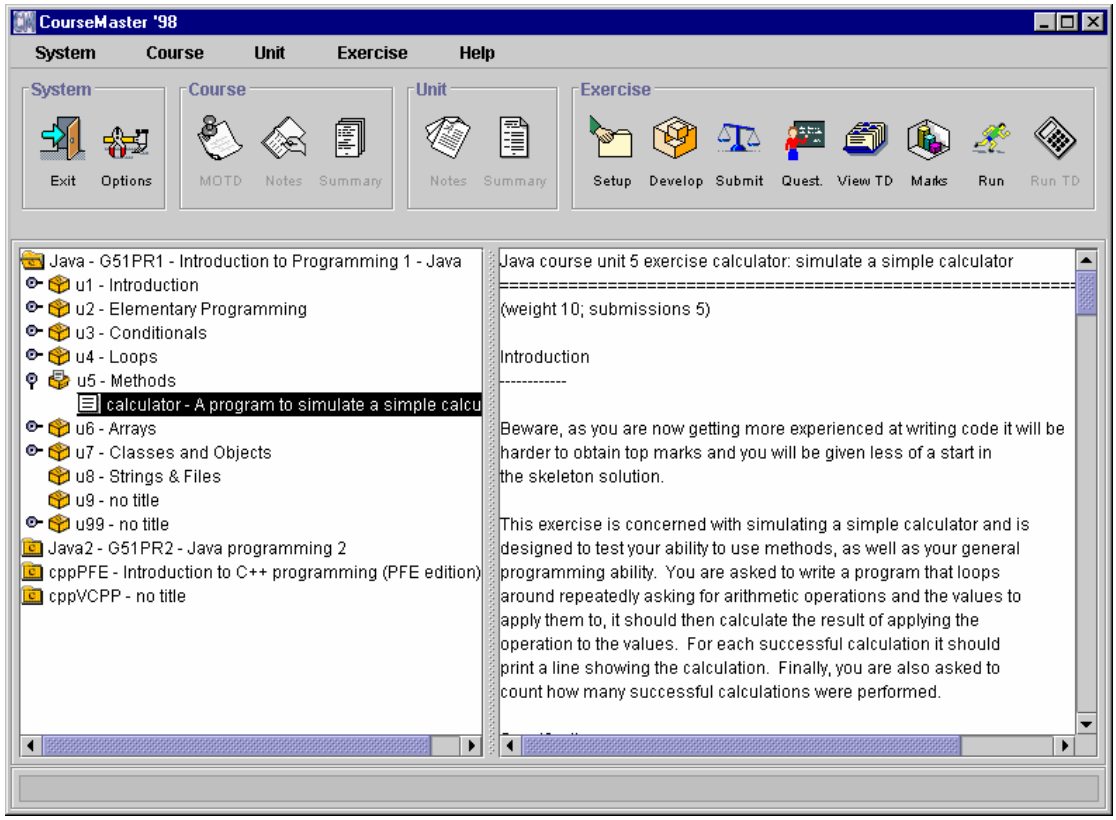

Figure 1 – JFC (Swing) CourseMaster student client

3. CourseMaster – a better Ceilidh

Because of the nature of Ceilidh's on-line development, as time progressed and the functionality was enhanced, it became more unwieldy and harder to support and extend. Finally, in 1997 rewriting Ceilidh commenced, from scratch, to rigorous software engineering standards, ensuring its continued use and ease of support. To this end, the system was redesigned using object-oriented methods and re-implemented to give the new system which was renamed CourseMaster (CM).

Object-oriented methods, including design patterns, were used in the creation of CM so that benefits could accrue from such areas as scalability, ease of distribution and networking, security, encapsulation, abstraction etc. and in particular to allow the easy and controlled later expansion of the system.

CM is based on a client-server paradigm. The student level functionality is provided by a set of interlinked servers written in Java, while three Java clients provide the student user interfaces (see Figure 1 for the Swing version). Student feedback is shown in figure 2. The tutor, teacher, developer and administrator facilities are provided by an extensive set of web pages that utilise CGI scripts and Unix software tools such as shell, awk grep etc. (figure 3). The use of Java and the availability of the free tools which allow the "Unix" like tools to run on Windows platforms [http://www.cygnus.com], make the entire system truly portable and, in fact, the system has been tested on the Unix (Solaris and Linux) and Windows (95, 98, NT and 2000) operating systems.

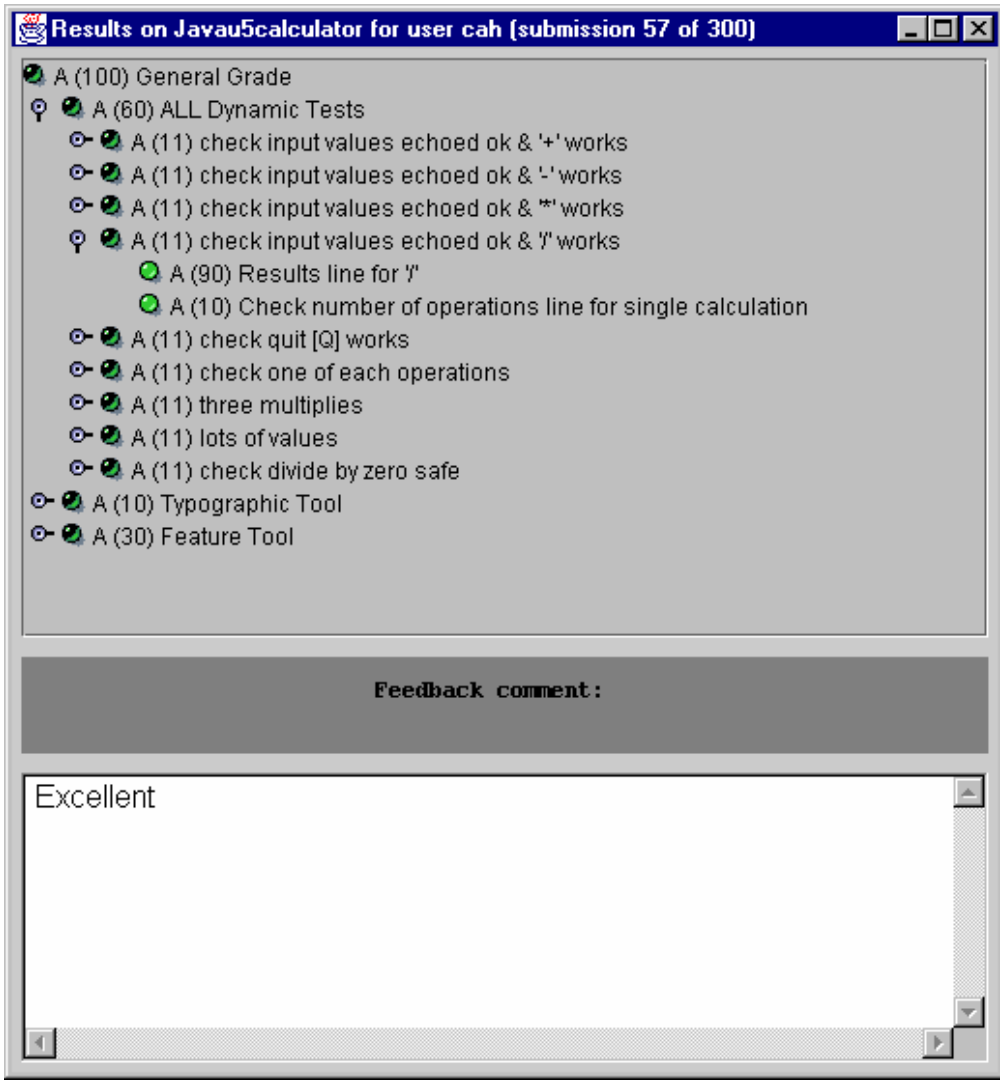

Figure 2. CourseMaster student feedback

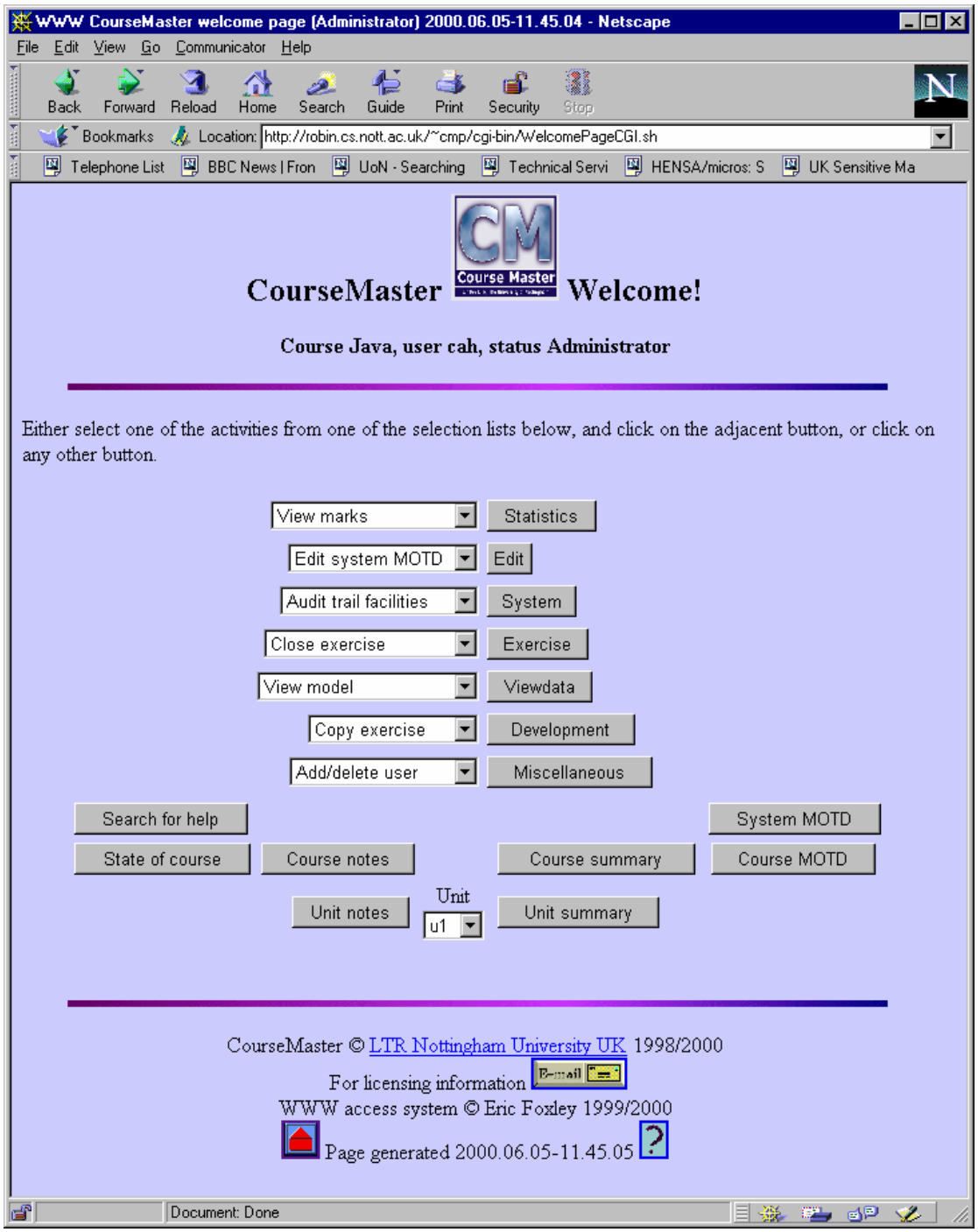

Figure 3. CourseMaster Administrator Interface

During redevelopment many enhancements were made. Most notably are: the development of a much enhanced GUI client for students making the system much easier to use; the greatly enhanced feedback mechanism (figure 1) enhancing the learning experience; and the addition of a powerful new subsystem allowing the marking of diagrams. In general CM is a great improvement over Ceilidh in terms of usability, maintainability, expandability and feedback.

4. Evaluation

Copies of Ceilidh were taken by over 330 institutions around the world. Usage is more difficult to gauge as the original system was free and not every user bothered to register. Given the numbers that did either register, email or take upgrades, it is estimated that the number of sites using Ceilidh was about 100, the biggest of which used it on approximately 2300 students per year.

The new CM system has been in active use by Nottingham, concurrently with development, for two years. There were 291 students using the system on the latest Java module during the year 1999/2000. Student feedback is extremely positive with the number of emails requiring assistance reduced by a factor of about ten compared to Ceilidh emails from previous years, despite the number of students more than doubling. Complaints were usually from students who had coded before and did not like the stricter constraints placed on their style by the good software engineering practices that CM enforced.

It is noticeable that female, overseas and mature students in general preferred CM. The strongest complaint was due to the scaling process applied to marks at the end of the year; this despite several warnings to the students that this would take place. Because students have multiple submissions with detailed feedback it is not surprising that the average mark for the course is very high (around 92% this year). Due to compensation rules marks with this average are not acceptable and so the marks are scaled down (via the administration sub-system). Even with this problem the number of complaints was in single figures, and after detailed explanation eventually all but one student was satisfied with the process.

Early beta versions of the system have also been tested at Ngee Ann Polytechnic (Singapore) and by Kings College London (UK). These have been used for C and C++ courses respectively. Feedback from academics has been positive with both institutions able to run pilot schemes on real students and to set their own exercises and mark schemes.

A beta of the system was made available in December 1999 and has been taken by nearly 20 institutions. The first full release will be June, 2000 [7].

5. Conclusions

Ceilidh was used successfully for over ten years at approximately 100 sites world-wide marking work on many programming languages. The new CM system has been running for two years at the University of Nottingham covering an introduction to Java course and an intermediate Java course each year. It has also been piloted with C and C++ courses at other academic establishments. Most recently the diagramming sub-system has been integrated and used to mark both object-oriented and electronic circuit diagrams.

Student reaction is overwhelming positive with a few minor exceptions. Staff acceptance is also high, especially given that most staff no longer need to mark programming course-works for tutorials. However, staff can still keep track of their tutees via the administration sub-system.

At Nottingham some students have used both Ceilidh and CM depending on their year of entry and courses selected. Students definitely prefer the use of CM over Ceilidh, showing that not only is the new system easier to administer, but also has a better interface and gives more informative feedback to students.

Enhancements are in hand to improve the learning experience of students while reducing the burden on teaching and administrative staff. CM improvements include: the implementation of submission over the WWW (to allow the easy integration of CM marking with other courseware and material presentation environments); the production of a remote console (allowing CM to be administered remotely and from any number of sites); automatic emailing of student progress to tutors; tools for marking the object-oriented aspects of programs in more depth; increased security and general improvements to many other aspects of the system.

In addition the LTR group is researching into many longer term items. These are all based around helping students whist minimising staff resources and include web based and AI based systems. For instance, agent based systems to monitor students progress and automatically suggest when they are able enough to move on to new topics or where they need to go and what they need to read to cure specific problems. These problems can be identified by a variety of mechanisms such as monitoring CM feedback or course material access or more directly by parsing emails or other requests for help.

References

[1] http://www.cs.nott.ac.uk/~ceilidh/papers.html

[2] E Foxley, C A Higgins, A Tsintsifas, 'Security Issues under Ceilidh's WWW Interface', *Proc ICCE'98*, Vol 1 pp235-240, Beijing, China, 14-17 Oct, 1998.

[3] E Foxley, C A Higgins, A Tsintsifas, 'The Ceilidh System: A general overview', *Second Annual Computer Assisted Assessment Conf*, Loughborough, UK, 17-18 June 1998.CourseMaster General

[4] E Foxley, C A Higgins, A Tsintsifas, P Symeonidis, "The CeilidhCourseMaster System, An Introduction", 4th Java in the Curriculum Conference, South Bank University, UK, 24th Jan 2000.

[5] http://www.cygnus.com/

[6] E Foxley, A Tsintsifas, C A Higgins, P Symeonidis, Ceilidh, a system for the Automatic Evaluation of Students Programming Work, CBLISS 99, University of Twente, Holland, 2-7 July, 1999.

[7] http://www.cs.nott.ac.uk/CourseMaster/