The

TELRI

Technology Enhanced Learning in Research-led Institutions

Project

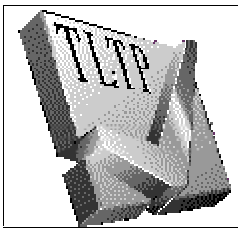
CASE STUDY

Peer Assessment in Computer Science

Evaluation from implementation in courses

The TELRI Project
Centre for Academic Practice
University of Warwick
Coventry CV4 7AL

Email: telri@warwick.ac.uk
Web site: www.telri.ac.uk



The TELRI Project is a three year project funded under phase 3 of the Teaching and Learning Technology Programme (TLTP).

	**CASE STUDY** **Peer Assessment in Computer Science**

Title
Peer assessment in computer science programming assignments

Department	Computer Science
Institution	Warwick

Description of the course

The course aims to enhance the development of students' capabilities in constructing and evaluating programming scripts, in reflection on the approaches of others and written communication of their ideas.

The course introduces students to basic data structures, how to use them and how to construct their own, currently using the Java programming language. The practical skill of programming is acquired only through experience highlighting the need to develop students' capabilities in the construction of programs.

The course involves lab sessions based around a worksheet of programming exercises that students work through with assistance from postgraduate demonstrators, followed by a test where marks contribute a small portion of the credits for the course.

Course details

- Title: Design of Information Structures
- Second term, core module for first year students of Computer Science, Computer Systems Engineering and Computer and Business Studies, as well as an optional module for a number of other degree programmes (and hence with widely varied experience.)
- 240 students
- 30 hours of lectures
- 8 hours of lab sessions
- Web-based test comprising MCQs, Permutational MCQs, which seek to assess high level learning and overcome the problem of guessing, and open response problems.

Details of the ICT intervention and how it aimed to enhance the learning and course activities?

Web-based worksheets are used as a basis for lab sessions, each presenting the student with problems to be solved and including links to supporting explanatory material.

In each lab session, students first spend 1.5 hours working through the exercises with help from postgraduate demonstrators. The last 30 minutes of the session is used for a web-based test including "free response" exercises conducted under exam conditions where their answers are recorded in a database.

In view of the large numbers of programming scripts, the tedious task of marking and the need to provide more valuable and timely

Other teaching methods used to support activity (e.g. tutorials, lectures, oral presentations, lab classes)

- Lectures
- Practical coursework assignments
- Lab sessions guided by web-based worksheets

The lecture support for the course was designed to replace previously poorly attended seminars with more practical lab sessions based around programming exercises.

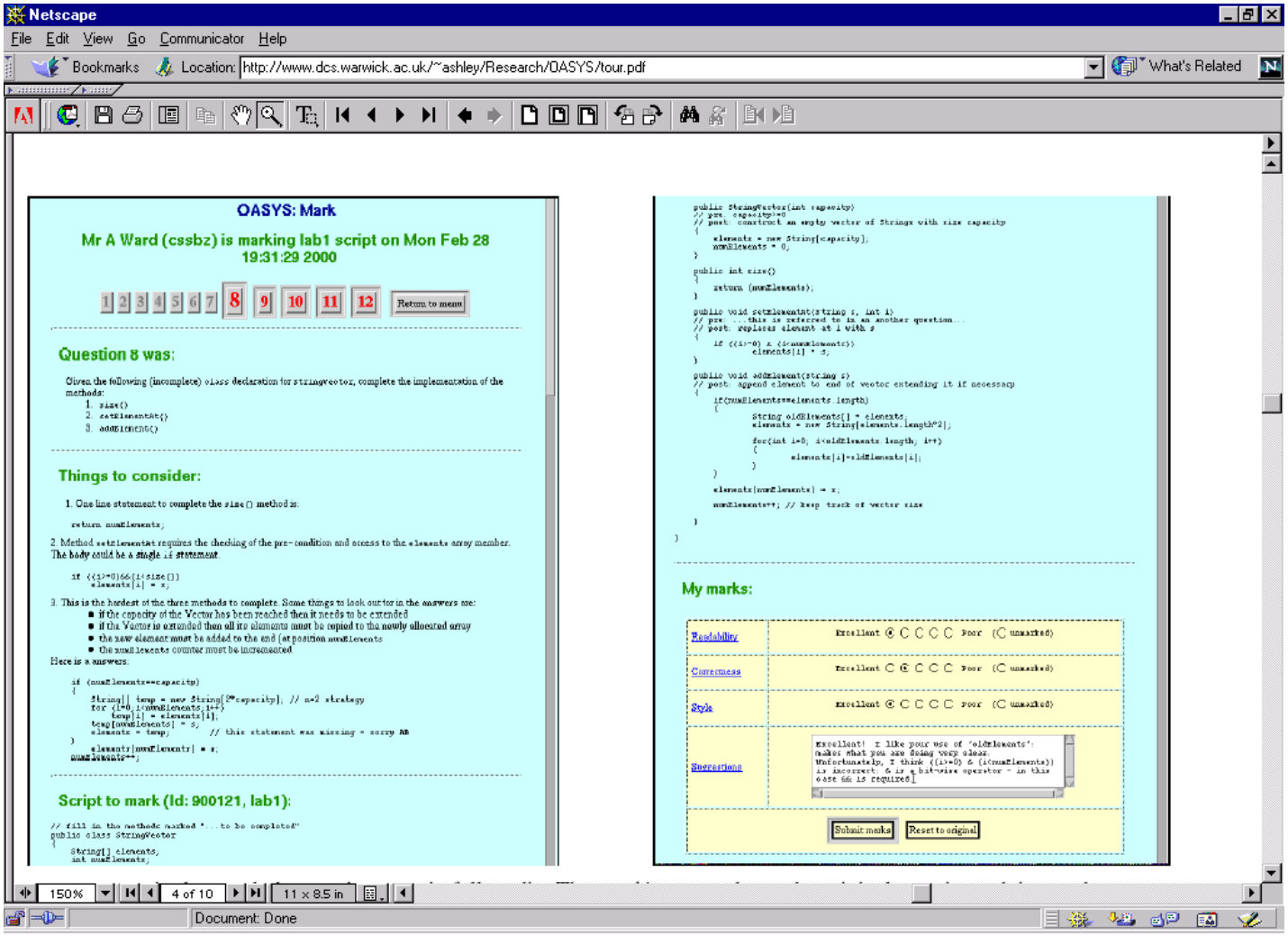
formative feedback on the students' approaches, a peer assessment system was introduced. After the test and before their next session, the students become assessors, marking three of their peers' scripts in the form of a grading to various criteria and an optional free-text comment. As soon as a script has been marked, the original author can view their grade and feedback. Several online assessment tools are available, but none of these appear to offer support for peer assessment management. A database-driven and web-based system, OASYS (Online Assessment SYStem) was developed in the Department to automate script distribution and provide anonymity to support the approaches. The technology employed is a web server, a scripting language (used to program dynamic pages) and a database. We used Apache, PHP and mySQL respectively, which are "free" open-source products.	**Other technologies used to support activity** (e.g. CD-ROM, web resources, lecture notes, references, online tests) Web resources and other supporting material is provided via links within the exercise guidance and feedback. An automated multiple choice (MCQ) based system alleviates the tutor load caused by paper based marking, but obviously higher order learning outcomes and capabilities such as style and elegance, which the course aims to develop in students, cannot be assessed by objective testing.
How the course was previously run The course had previously been taught using lectures to deliver the material, augmented with tutorial sessions and tutor marked assignments. Attendance at the tutorials was not compulsory. The lab sessions and programming exercises were each followed by a test marked by graduate demonstrators.	**Problems with previous teaching methods** As might be anticipated, attendance at tutorial sessions was generally low until the assignment drew near. While the new lab sessions improved attendance, due to the number of students and lab sessions, they generated a total of 800 small exam-like paper scripts. These needed to be marked, checked and collated within just a few weeks. Feedback was limited with students typically receiving a simple grade rather than informative formative feedback. The feedback also took a long time to arrive to students due to the low ratio of demonstrators to students. Furthermore, the demonstrators found the marking a repetitive and unrewarding task.
Intended capabilities to be developed ▪ Ability to read, understand and identify problems in someone else's code ▪ Ability to reflect upon one's answers that lead students to lead to understanding effective practice ▪ Ability to communicate within a student group	**Methods of assessment** The 30 minute tests at the end of each lab session carry only a small amount of credit, augmenting the existing assessment methods of a programming assignment and a more traditional examination. Open questions are the main emphasis in the approaches. Students might be asked to

Marking can also be portrayed as educationally useful as it requires active evaluation, judgements and decisions to be made. The approach therefore encourages students to reflect on their own answers and approaches taken. The ability to read code as well as write code is an important skill in industry, where systems are rarely built from scratch. The aim of the approach is to ensure that student feedback on their programming approaches is timely, specific, discursive (fast and human feedback in triplicate!). One of the course objectives is to "learn how to program". This involves not only learning skills and procedures, but importantly developing a student's ability to "construct a program". Students should therefore develop capabilities in evaluation of subjective outcomes such as correctness, elegance and style.	outline some Java code or explain something. A peer assessment approach immediately raises questions about the reliability of the grading for tests marked by students. These concerns have been addressed in several ways in the overall course design: 1. The marking interface displays some guidelines for each question, pointing our potential aspects of a good answer. 2. Each script is assessed by three different students. The variation in marking is calculated and in the event of a large disagreement, the script is automatically highlighted for moderation by the tutor. A learner can also request moderation should they feel they have an unfair mark. 3. The distribution of the scripts can be controlled such that each script is marked by a good, intermediate and poor assessor, as determined by the results from a computer-based multiple choice test or other available information about the learners.
Problems in setting up the course or technology The feedback interface took time to implement and this meant that the students did not receive the speedy feedback we were aiming for. There are many possible combinations of cases of unmarked, partially marked, auto-marked or unanswered questions, which cause problems when collating data into a complete script. There were very few other problems with the technology. The initial system took just two weeks to build starting from the initial concept and ending with a functional prototype which was improved and extended during its use in the early stages.	**How these were solved** We will be using the system for a second time and hope to improve in this respect. The potential flexibility itself creates problems, as yet unresolved, though straightforward to deal with.
Extent of development of intended outcomes **TELRI evaluation** (e.g. leading to changes in the course design or ICT approaches) The main area where there is significant benefit to student learning quality is where students have access to each other's work. In the current way the course is organised where	**Academic tutor evaluation** (e.g. leading to changes in teaching practice) There certainly appears to be a bilateral form of dependence between electronic and peer assessment. We cannot have good (i.e. speedy and anonymous) peer assessment at this scale without electronic assistance and,

sharing of work is linked to the marking exercise, this limits each student to only three examples. While this is manageable for grading purposes, we feel a preliminary research-based or open-ended exercise would provide a forum for discussion of approaches that would benefit students in undertaking the assessed exercise. The students could use the OASYS facility to submit their work to a web site and to view and comment on all the approaches *before* submitting a final assessed assignment exercise. This is plausible only when the assignments are of a more open-ended nature where the responses are evaluative and judgement-based rather than procedure-based. The tutors need to find ways to embed the programming skills development within an interesting and challenging exercise linked to professional practice. Access to the marked work of all the students would provide a transparent view of the assessment criteria and enable more open-ended exercises to be supported. Of interest are the responses to the question of whether marking of the tests by the students should be anonymous. 94% of respondents felt that anonymity is important or were indifferent to the question. Evidence from other TELRI case studies suggests a *lack* of need for anonymity. The differences may be subject-related, since certainly in humanities and social sciences, students often do not pick the same assignment or essay topic and the *perceived* risk of plagiarism is less apparent.	conversely, we cannot have good marking (i.e. quality feedback) of scripts without humans. The approaches described here combine the two in an effective and efficient manner. The students understanding and skill in programming is enhanced and their ability to evaluate approaches of others and reflect on their own work is facilitated more than with previous teaching methods. The more marking the students did, the better their own results became. The largest remaining problem appears to be that of convincing the students that this is formative, not summative feedback – an issue confused by awarding 10% of module credit for the lab tests to encourage active attendance. If the issue of anonymity were addressed, the approach could be extended to allow not only administrators access to the students responses and feedback comments entered whilst marking, but all the students too. **Students evaluation** (e.g. leading to changes in learning practice) In general the students had a positive response to the approaches. 90% of respondents stated that they "reconsidered their own answers whilst marking" in our module questionnaire, which is certainly an improvement from never revisiting the test at all. Comparing student achievement of the IT supported approaches with paper-based method used in the previous year, there was no difference in the average mark. It was clear that learners with a strong reproducing orientation tended to receive lower marks and those that tended to attempt to memorise material stated that they found marking difficult. However, there was less spread of marks within the student group as a whole.
Unintended benefits or costs **TELRI evaluation** The evaluation undertaken by the course tutors exploring the possible correlation between exam results, coursework mark and the lab session grades indicates links between the three results. In the previous course	**Academic tutor evaluation** A valuable spin-off of the course design is that staff are freed from the onerous task of marking hundreds of scripts and can concentrate their efforts on teaching and moderation.

where marking was by expert tutor/demonstrator, the grades for an individual student were a reasonable predictor for exam result, and also, less significantly for coursework marks. Where lab marks were produced mainly by peer assessment, however, they were still a significant predictor of exam result, but not this time of coursework mark. Despite their practical emphasis, therefore, the lab tests may still be testing skills more useful in an examination than in practical assignment building and documentation. It might be useful to review whether the desired capabilities are truly being assessed in the exams, given that the assessment criteria for the peer assessed coursework are more transparent.	We feel the approaches respond effectively to the student who politely asked via email to "please get someone with intelligence to mark the tests" – a request that is actually a little self-deprecating when we consider this learner in the assessor role. The speedy feedback we were aiming for was not achieved in the first run of the course due to the problems with the technology outlined above. **Students evaluation** Evaluation compared lab mark distributions from the two years, that is, between the course where coursework was marked by expert tutors or demonstrators and this new approach using peer assessment. The two distributions are similar in shape and have a very similar average value. However, the students were much harsher assessors than the tutors/demonstrators and this led to stress on the part of the student and requests for more time in the free responses.
Cost-effectiveness analysis Most of the staff time costs are up front with the development of the system and preparation of the marking guidelines. Although computers cannot make expert judgements of scripts, they can certainly be used to simplify document management problems. It would be highly impractical to use peer assessment with our 200 students in this way without some electronic assistance and this benefit justifies the initial investment of time. The current approach has already been shown as an improvement over a paper-based system in terms of efficiency gains. With several reams of paper representing the 800 scripts to be marked by the four unlucky demonstrators, just distributing the scripts for marking, then collating and publishing the results was very time consuming and the team quickly fell behind. The cost-benefit of automating this aspect of the test process is immense. The benefit of the peer assessment approach has been partly validated in the fact that the vast majority of students are now considering their own work more reflectively through marking others' answers. Many of the further developments suggested below will take	**To what extent can the course design approach support higher student numbers?** The course design aimed to support courses with high numbers of students. Often it is hard to both increase benefits and reduce costs and particularly with respect to supporting high numbers of students. The peer assessment system implemented appears to have achieved this by balancing effectively a combination of computerised and human approaches. The main restriction for scaling up would be the extent to which moderation by tutors might be incorporated for more difficult open or expert problems. The OASYS system can easily transfer to other courses. The department is considering wider use to support other computer science courses and there is interest in its use in Mathematics, Physics and Engineering.

considerable time to implement. The cost-benefits of these additional factors is difficult to predict but may certainly enhance overall the learner's experience of peer assessment.	
Further developments planned for this or other courses A number of recommendations were drawn up from the evaluation of the course and the main points are given here: ▪ Faster feedback to encourage students to mark more effectively: perhaps through improving guidance to students describing the process, their expected input and return; by encouraging them to mark in pairs; by publically ranking students in marking effectiveness terms with a prize for the best "marker" ▪ More reference to the labs within lectures ▪ Providing full credit once a minimum standard is reached to engage students with the formative nature of the assignments, giving merit awards for very high achievers to encourage work beyond the minimum. ▪ Asking learners to give feedback on their received marks, cf. TELRI evaluation of Extent of development of intended outcomes above.	**Further details about the course approaches** http://www.dcs.warwick.ac.uk/~ashley/Research/OASYS/



Learners are required to mark three scripts to gain full credit. Marking can be done in the learner's own time. The marking page (left) shows the original question, advice on the correct use of the marking scheme for the question, the answer to be marked and the marking interface. Administrators can browse all students' responses to questions and also their feedback comments entered whilst marking.

Entirely free response answers are a major feature of the system. That these are marked by other learners distinguishes it from a computerised marking system and thus supports open-ended assignments that encourage the development of students' evaluative and innovative capabilities.

The system incorporates the use of HTML links to supporting material and the use of images.

Questions can use mathematical symbols, but these must be created as images as current web browsers do not support such symbols directly.

A "See my marks" page (not shown) allows learners to obtain feedback on their own scripts. These can be viewed anytime, anywhere on the web.