ASPECTS OF WEB-BASED PEER ASSESSMENT SYSTEMS FOR TEACHING AND LEARNING COMPUTER PROGRAMMING

Ashley Ward, Jirarat Sitthiworachart, Mike Joy
Department of Computer Science
University of Warwick
Coventry CV4 7AL, UK

Abstract

This paper discusses the potential for web-based peer assessment, based on the numerous possibilities for different communication patterns that the technology affords. It describes and compares two novel web-based peer assessment systems for computer programming courses, and discusses their deployment on large programming modules. The results indicate that these peer assessment systems have successfully helped students to develop their understanding of computer programming.

Keywords

Web-based education, peer assessment, computer programming, P2P communication

1. Introduction

1.1. Peer assessment in general

Assessment is a tool for learning, but traditional assessment methods often encourage surface learning, characterised by memorisation and comprehension of information. Deep learning, such as creating new ideas, and critical judgement of a student's work, can be encouraged by the use of peer assessment [1,2]. When students evaluate each other's work they think more deeply, see how others tackle problems, learn to criticise constructively, and display important cognitive skills such as critical thinking [3,4].

Falchikov [5] defines peer assessment as "the process whereby groups rate their peers". Somervell [6] states that peer assessment engages students in making judgements on the other students' work. In the peer assessment process, students are involved both in the learning and in the assessment process. Peer assessment is primarily a tool for learning rather than for summative assessment [7]. Dochy and McDowell [8] remark that "peer assessment is not only a tool to provide a peer with constructive feedback which is understood by the peer. Above all, peer assessment is a tool for the learner himself."

1.2. Potential for web-based peer assessment

The WWW attained popular public awareness in around 1994, where it initially reproduced an existing role, that of *broadcast* media, where central powerful authorities publish information (such as marketing brochures) to a weaker audience. However, many other communication patterns are possible and are increasingly being realised. For example, web *forums* use a many-to-all *bulletin board* communication pattern, where many people can post, and everyone can read. Another example is the peer-to-peer (*P2P*) one-to-one pattern, which achieved notoriety in the Napster service, used by many to copy music files from one computer to another without the use of a central authoritative server. The P2P pattern is also illustrated in the recent phenomenon of the *weblog* ("blog" for short), where peers communicate personal diaries or thoughts to other peers, usually small in number.

Most current electronic assessment systems (such as CourseMarker [9] and Webassessor [10]) use a broadcast pattern extended with a return path, for example in objective testing, where the tutor publishes some tests to the student population, and the students' answers are communicated back to the central broadcasting authority (the tutor) along the return path. Other systems (MLEs, such as Blackboard [11] and WebCT [12]) with broader scope additionally provide a web forum, using the bulletin board pattern. Many other possible types of electronic information flow for teaching and learning remain unexplored, however.

Peer assessment is an activity that requires the use of a P2P pattern. Without electronic assistance, information must be communicated verbally or by using paper [13]. Although paper is a well understood, reliable and flexible communications medium, it has the inherent limitation that it cannot concurrently exist in more than one physical space without first being copied. The best paper copying systems invented thus far (photocopiers) provide acceptable performance when the broadcast pattern is used, for example when 250 copies of lecture slides are required. In the peer assessment P2P context, however, many copies must be made of many different students' pieces of work for assessment. With 250 learners, the logistics of the copying alone overwhelm the task.

Electronic communication, however, can transport copies of information in "near zero" time (from the human perspective). Assessment of peers' work can therefore potentially begin as soon as the work is complete, or even whilst it is in progress. The assessment feedback can be provided to the learner even as it is constructed, or it can be instantly copied to someone else for moderation. The entire current state can be copied and viewed, providing the tutor with the ability to see all learner interactions as they happen.

In addition to instantaneous copying, computing technology can *process* the information needed for peer assessment (for example, the NetPeas Web-based peer assessment system [14]). For example, it is possible for an electronic communications system to contain a large amount of specific guidance for learners that can be shown only when the situation is relevant. Mark calculations can be performed automatically. Some types of assessment can even be performed by the computer (for example, Project Essay Grade [15] and Intelligent Essay Assessor [16] attempt to assess free responses using natural language processing), and when used in a group context, can be integrated with human feedback.

Peer assessment is particularly relevant in the context of teaching and learning computer programming. Due to the size and complexity of modern commercial software products, most commercial programmers work as part of a group, and peer review of code is common. Electronic computing assistance is relevant in the teaching and learning context, since programming (and perhaps increasingly design) is largely a situated task involving the use of a computer.

Electronic communications, then, have a great deal of potential, enabling the use of many possible varied communication patterns, including P2P for peer assessment of computer programming. But with this freedom of choice also comes the responsibility of choice. What is the most appropriate information flow pattern to use? Should individuals communicate only through the computer or can we integrate the use of the computer for communication into some group activity? The problem is now primarily one of *process design* rather than feature provision. We have returned, then, from technical considerations to the basic lesson planning and curriculum design skills that a teacher needs.

2. Two experiments

We have been investigating the possibilities of using computer technology in learning and teaching of computer programming in the Department of Computer Science at the University of Warwick for many years. The electronic submission system *BOSS* [17] was an early product, later extended to perform some automatic assessment and plagiarism detection. In January 2000 and again in 2001, Ward and Bhalerao [18] developed and used the On-line Assessment SYStem, *OASYS*, one of the first implementations of web-mediated peer assessment of computer programming. Sitthiworachart, Joy and Ward continued the work in 2002, and we refer to their system as *OASYS2*. The following sections compare our experiences of web-based peer assessment.

2.1. From tests to programs

OASYS was deployed in the first year Design of Information Structures module in the Department in an attempt to give students effective and timely feedback on their progress in laboratory sessions. 240 students in January 2000 (and then 275 students in January 2001) took laboratory sessions that comprised 90 minutes of group experimentation with Java programs under instruction from worksheets, followed by 30 minutes of on-line testing, run under exam conditions, each student having sole access to a computer running a web browser. The on-line tests (implemented with Apache [19] and PHP3 [20]) assessed the students' understanding of their earlier work during the lab session and were comprised of multiple choice (MCQ) and free response questions. In the latter type of question, students were typically asked to write a few lines of Java code or a few English sentences. Students' responses to the questions (which we refer to as a *script*) were recorded in a MySQL [21] database. The answers to the free response questions were then peer assessed.

OASYS2 was developed from OASYS in 2002 and deployed in the UNIX programming module delivered by the Department. The module aims to give students a basic understanding of the UNIX operating system and competence in programming using a UNIX shell. Out of a total of 300 first year undergraduate students, 215 students responded to an online questionnaire, the results of which forms the basis for the data presented in this paper. There were 189 male and 26 female respondents, 153 whose first language is English and 62 who are not native English speakers. The students worked on three programming assignments in their own time, and submitted online using BOSS. The module tutor marked assignments 1 and 3, but the second was marked with peer assessment using OASYS2. This assignment was also double-marked by the module tutor, to provide an expert reference against which the marks awarded through the peer assessment process could be compared.

OASYS, then, used peer assessment for marking short answer tests taken under exam conditions. OASYS2 has progressed to peer assessment of complete programs, written in the students' own time.

2.2. Use of automated marking

In OASYS, the multiple-choice questions were marked automatically by the system, leaving assessors to focus on the free-response answers. The automatically produced marks were made available to the corresponding script authors almost immediately, but were not shown to peer assessors as it was felt that this might unduly influence their marking.

The BOSS online submission system used in the OASYS2 process is capable of running automated tests, running students' submissions against a set of test cases (the expected output given a particular input) constructed by the assessment designer. Ten different automated tests were used in OASYS2, each test producing textual results and a numeric score. The scores were allocated 50% of the assessment credit, the remainder allocated to peer assessment. The automated test results and scores were made available to assessors along with the original submission at all stages of assessment, in the hope that the extra information would help students to understand the submissions whilst marking. However, we did find that a few students based their marks solely on the results from the automatic test scores.

2.3. Assessing in groups

After each test session in OASYS, each student was asked to become an individual assessor. An incentive to become involved was provided in the form of a small amount of module credit for participation. Assessors were guided by the provision of context-specific "model answers" and hints for marking. The marking itself required the assessor to evaluate the free response answers on three criteria (readability, correctness and style), and asked for the optional provision of a discursive textual feedback suggestion. All interactions were performed anonymously.

Assessors were asked to mark three of their peers' scripts before their next lab session. This design decision about information flow was arrived at through a desire for multiple marking from which a majority decision can be determined – hence the use of an odd number, and a small number in order to minimise the amount of marking required of the students. If each student marks three scripts, then it is possible to arrange an algorithm to direct the distribution of scripts to assessors such that each script is marked three times. In OASYS, this was done dynamically: an assessor requesting a script to mark was provided with one from the "pile" – specifically, a non-deterministic choice from the set of scripts which have currently been marked the least number of times, excluding the assessor's own script. This algorithm produces the random-looking distribution of scripts to assessors shown at the left of Figure 1, with three assessors per script and three scripts per assessor (there being the possibility, however, of a marking shortfall if some students do not mark three of their peers' scripts).

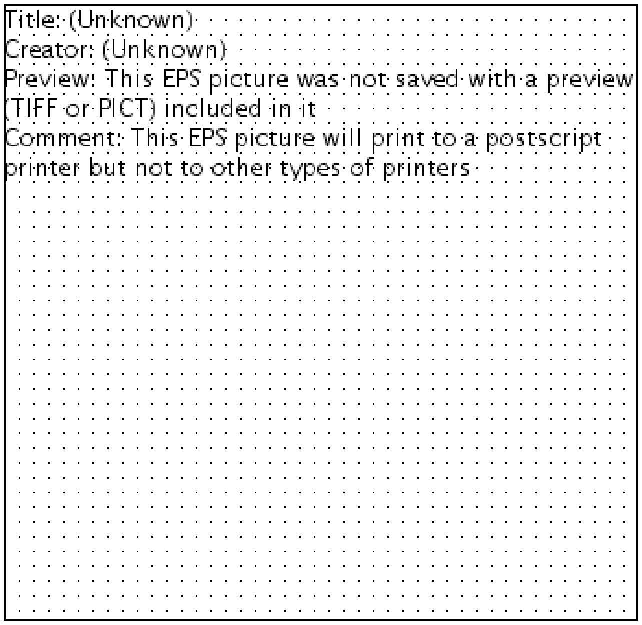
Title: (Unknown)
Creator: (Unknown)
Preview: This EPS picture was not saved with a preview (TIFF or PICT) included in it
Comment: This EPS picture will print to a postscript printer but not to other types of printers

Figure 1: Communication patterns in OASYS

The peer assessment stage in OASYS was performed in the students' own time, usually on an individual basis. But Falchikov [5] defines peer assessment as a *group* activity. Accordingly, in OASYS2, group assessment sessions were formally timetabled. Students were divided into groups of three and the distribution of submissions to assessors was organised ahead of time, mapping each group of three assessors to a set of three submissions authored by learners from another group, as shown on the left of Figure 2 (although note that the real situation with 300 students was not as symmetric as the figure suggests). Each student spent 30 minutes anonymously assessing other students' submissions before discussing their marking with the other students in the group (who had been marking the same set of submissions). They could then revise the marks they had given, in their own time, up until the marking deadline was reached.

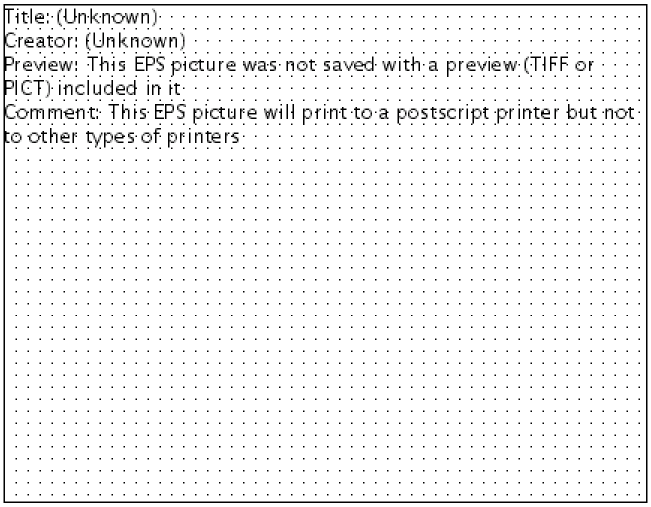
Title: (Unknown)
Creator: (Unknown)
Preview: This EPS picture was not saved with a preview (TIFF or PICT) included in it
Comment: This EPS picture will print to a postscript printer but not to other types of printers

Figure 2: Communication patterns in OASYS2

Additionally, each group of students (and hence set of submissions) was composed of students with a range of ability, as determined from results from an earlier assignment in the module. With this pattern,

- each assessor discusses their marking within a peer group of varied ability,
- each assessor observes and assesses a set of submissions likely of varied quality, and
- each submission receives a set of marks from assessors of varied ability.

The arrangement of assessors into groups is a significant improvement on OASYS. Assessors make *comparisons* between submissions and are encouraged to reflect on their decisions through discussion with their peers, instead of making judgements in isolation.

2.4. Tutor moderation

In both OASYS and OASYS2, the expert tutors could override peer assessment marks if it was considered necessary, in a form of a tutor moderation process. For each submission, the system calculated the standard deviation of the three marks given through peer assessment. These figures were presented on a web page visible only to tutors, where submissions with a high standard deviation in peer marks were highlighted. This design is illustrated on the right of Figure 1. Tutors remarked these controversial submissions. Learners could also request tutors to remark their submission if they considered it necessary.

In OASYS, most peer assessment was not examined in detail by the tutors, and there was a high focus on the exact marks given and not the discursive feedback. A few assessors exploited this, giving the same "middling" marks to each and every script and no discursive feedback: a strategy which in some cases did not lead to a high mark standard deviation but which was perceived as indiscriminating by the corresponding script authors.

2.5. Peer assessment of peer feedback

OASYS2 added another stage to the process, shown towards the right of Figure 2, additional to the tutor moderation shown on the right of Figure 1. Now students took a total of three different roles at different stages in the process. In the new third and final stage, students were asked to assess feedback given at the previous stage. In this *feedback marker* role, students observed and assessed the quality of marking given by other assessors. Feedback markers gave credit worth 20% to submission assessors. Our hope was that this stage should help students to develop their critical judgement skills and encourage them to take the assessor role in the previous stage seriously.

We summarise the three roles each student adopted as follows.

- *Submission author*: a student writes one submission. They receive ten automated test results and scores (for 50% credit) and three peer assessments (for 30% credit).
- *Submission assessor*: a student assesses three of their peers' submissions. They receive three "feedback marks" reflecting the quality of their assessment, for 20% credit.
- *Feedback marker*: a student "marks marks": they give marks on three of their peers' assessments (of submissions). They receive no feedback about their performance in this role – this is possible, but care must be taken to avoid infinite recursion (assessment of assessment of assessment of assessment…).

In section 2.3, we noted that each group was formed from students of varied ability and stated the advantages that follow in this particular communication pattern. Adding feedback markers to the process gives students the opportunity to see assessments given by assessors of likely varied ability and draw conclusions.

In our reshaping of OASYS, then, we have shifted from an emphasis on marks in OASYS to an emphasis on *useful feedback* in OASYS2.

2.6. Summary of OASYS improvements

The changes we made when revising our OASYS system and process design are summarised below.

OASYS	OASYS2
OASYS gave short tests MCQ and Java exercises	Tutor set a sizeable shell programming assignment
Students created *scripts*	Online *submission*
No group discussion, individual marking	Group discussion and group marking
No control for ranges of ability	All students observe a range of ability
Automatic test results not shown to assessors	Automatic test results and scores shown to assessors
Emphasis on *marks*	Emphasis on *useful feedback*
Assessment judgements made in isolation	Groups make comparisons between submissions
Tutors monitored the quality of marking	Peers mark the quality of marking

3. Results

3.1. Evaluating their evaluation skills

In an attempt to obtain a controlled measurement of the effect of peer assessment, in 2002, the students were asked to complete two tests additional to the assessment process discussed above. Test I, run before the peer assessment exercise asked students to analyse and evaluate a short shell program. Test II was very similar in content but had cosmetic differences. The numbers of times that students commented on various (unprompted) aspects of the code were counted. A summary of the results is displayed in Figure 3, which suggests that when evaluating a shell program, the students were able to characterise more finely after they had been through the peer assessment process.

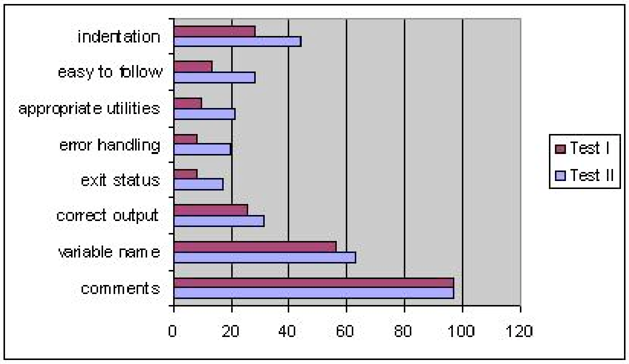

Figure 3: Pre and post peer-assessment test results

3.2. The students' opinions

In the questionnaire-based evaluation of OASYS in 2001, over 90% of questionnaire respondents (N=54) agreed that they had "realised mistakes in their own answers whilst marking", validating at least the use of revisiting the tests.

On average in 2001, students stated that they spent roughly 2.5 hours marking in total – about 10 minutes per script. However, only 22% agreed that they had "found the marks that other people gave useful": 45% were indifferent to this statement, and 31% disagreed.

Students were required to evaluate OASYS2 at the end of the process by filling in a detailed online questionnaire. The quantitative questionnaire results suggest that the peer assessment exercise was beneficial.

- 76% of students discussed with their groups when marking and thought that this discussion helped them understand more about the assignment. A few students felt it was difficult to start discussion.
- 58% of students felt comfortable when assigning marks. A few students did not fully understand the marking criteria.
- 65% of students were satisfied with their mark from the peer assessment, and considered that the peer feedback they received was relevant and useful.
- 80% of students agreed that seeing good and bad programs help them in learning programming, and marking helped them to think more deeply about their own work.

The qualitative students' responses generally appear to support the view that students can learn from each other through this process.

In the students' views, seeing different ways of solving programming problems and marking each other's work helped them to assess themselves and write better programs. The following quotes were submitted in the OASYS2 online questionnaire:

> *"I got the chance to observe two scripts that used different methods than my own solution to satisfy the specification. In order to be confident in my ability to mark these scripts fairly, I had to spend a long time studying them and hence acquiring an improved knowledge of how shell scripts are composed."*

> *"Marking others' work helps me criticise my own work and remind me of my own problems."*

Most students seemed satisfied with their marks, and considered that with adequate guidance the marks from peers could be as reliable as the marks from a tutor. However, some students thought the marks awarded by students were graded using different standards to those awarded by tutors. Markers could only base marks on a comparison of a submission to their own answer, and some students therefore thought that no student is really qualified to mark another student's work, as they are not trained for marking and are not experts:

> *"I would say peer assessment in a better way to learn how to write a good program. Nevertheless, some markers may not have the skill of marking and understanding of script."*

> *"I think it is hard to mark a student when you've never marked assignments before, especially UNIX scripts as I never had any previous experience with it. Additionally, the things I consider good or bad may not be the same for other people."*

> *"Very hard to do when you are marking someone who is technically in the same boat as you."*

3.3. Quality of students' marking

Each student's mark was compared against the mark awarded independently by the tutor. The marks awarded by the students were mostly higher than those awarded by the tutor, the means differing by approximately 18%, and scaling of peers' marks yielded results which, in almost all cases, matched the tutor's. This contradicts the students' opinions, presented in the previous section: with guidance, and design of appropriate information flow, peer assessment can be of a similar quality to that made by subject experts.

4. Conclusions

We have described an evolution of our approach to electronically assisted peer assessment for teaching and learning computer programming. We have moved away from assessing small tests toward assessing larger complete programs, are now using automated test results to inform assessors and are forming peer groups for assessment. The peer groups are composed of students with a range of ability, which has significant impact on the process: each student discusses marking within a peer group of varied ability, observes and assesses scripts likely of varied quality and receives marks from assessors of varied ability. Finally, we have added an novel final stage to the process, where students take the role of feedback assessor, observing and assessing the quality of marking given by their peers. The results from both OASYS and OASYS2 show that exercises contributed positively to the students' learning experiences.

5. Acknowledgements

We would like to acknowledge the assistance of Su Fang, student of Dr Amanda Dowd at Warwick for leading us to references about automated essay grading, and to the students for participating in these experiments.

References

[1] "Deep Learning, Surface Learning", *AAHE Bulletin*, **45**(8), 10-13, 1993.

[2] Topping, K., Peer assessment between students in colleges and universities, *Review of Educational Research*, **68**, 249-276, 1998.

[3] Sluijsmans, D., Dochy, F., and Moerkerke, G., Creating a learning environment by using self- peer- and co-assessment, *Learning Environments Research*, **1**, 293-319, 1999.

[4] Tsai, C., Lin, S., and Yuan, S., Developing science activities through a networked peer assessment system, *Computers & Education*, **38**, 241-252, 2002.

[5] Falchikov, N., *Learning Together: Peer tutoring in higher education*, Routledge Falmer, London, 2001.

[6] Somervell, H., Issues in assessment, enterprise and higher education: the case for self-, peer and collaborative assessment, *Assessment and Evaluation in Higher Education*, **18**, 221-233, 1993.

[7] Brown, G., Bull, J., and Pendlebury, M., *Assessing student learning in higher education*, Routledge, London, 170-184, 1997.

[8] Dochy, F., McDowell, L., Assessments as a tool for learning, *Studies in Educational Evaluation*, **23**(4), 279-298, 1997.

[9] http://www.cs.nott.ac.uk/CourseMarker/cm_com/

[10] http://www.webassessor.com/

[11] http://www.blackboard.com/

[12] http://www.webct.com/

[13] Purchase, H., Peer Assessment: Encouraging Reflection on Interface Design, Proc: 23rd Australasian Computer Science Conference (ACSC 2000), Canbrerra, Australia, 2000, 196-203.

[14] Lin, S., Liu, E., and Yuan, S., Web Based Peer Assessment: Attitude and Achievement, http://www.ece.msstate.edu/~hagler/May2001/05/Begin.htm

[15] http://cea.curtin.edu.au/tlf/tlf2001/williams.html

[16] http://www-psych.nmsu.edu/~pfoltz/reprints/Edmedia99.html

[17] Joy, M., and Luck, M., The BOSS System for On-line Submission and Assessment *CTI Computing: Monitor*, **10**, 1998.

[18]Bhalerao, A., and Ward, A., Towards electronically assisted peer assessment: a case study, *Association for Learning Technology journal (ALT-J)*, **9**(1), 26-37, 2001.

[19] http://www.apache.org/

[20] http://www.php.net/

[21] http://www.mysql.com/