

Title

The Automated Peer-Assisted Assessment of Programming

Skills

Authors

Stuart Lewis

Phil Davies

School of Computing

University of Glamorgan

Contact Details

sflewis@glam.ac.uk

pdavies@glam.ac.uk

Abstract

This conference paper introduces an “add-on” to the Coursemarker © Programming Environment that permits the use of student peer-support and peer-assessment. The students having submitted their own assignments for automatic checking by the system are required to peer-comment the programmes of a selection of their peers. This paper reports on the findings of the initial study, and suggests methods that can be utilised to automatically qualitatively assess the peer-marker’s ability in performing the marking process.

Introduction

The use of peer-assessment as a tool to develop student knowledge/ability is not new to higher education (Falchikov, 1995; Boud et al, 1999). From our experiences at the University of Glamorgan over the past ten years, the area of teaching programming to undergraduate students is one that has produced increasing problems with regard to high failure rates. The Java programming language, although widely used as an initial language in many programming courses due to its flexibility, provides a considerable challenge to many students (Lewis, 2002). The use of pair-programming has been promoted (McDowell, 2002) as a means of developing programming skills amongst students. This has involved students working together in the development of a single solution to a programming problem. The benefits to knowledge acquisition in developing a collaborative cognitive process is well documented “knowledge is commonly socially constructed, through collaborative efforts” (Salomon, 1993). Tools are currently being developed in an attempt to create a managed a peer-assessed environment that will support students in the field of computer programming (Sitthiworachart & Joy, 2003). However, the practice of using peers to support/assess each other brings into question the fair rewarding of student effort in this role as marker (Davies, 2003a). This ‘reward’ must be of a tangible and qualitative nature. It is also necessary that this reward is both objective and automated.

The essence of this conference paper is to report upon the development of a peer-assessment “add-on” to the Coursemarker © system, in order to utilise the benefits of peer-assessment in a way as to alleviate the inherent problems associated with the teaching of programming. The study described in this paper attempts to make use of peer-support, but not in the initial creation of the programme as is the case in traditional paired-programming, but in the

post-assessment of the submitted work. The study is currently in progress, and by the time of the conference the authors will be in a position to report upon the findings of the study.

Background

The HND introductory programming course had been using the Ceilidh (Benford,1994) system for several years (Lewis, 1997) running under Linux, first with Modula 2 and then Java using the xemacs editor. The new CourseMaster (Foxley, 2000) system that replaced Ceilidh was adopted in February 2000. As this is written in Java, it is platform independent and could have been run on either Linux or any variant of Windows. The availability on the Windows platform was very popular with the students. These teaching environments have been used to aid the marking process and provide feedback to students concerning their submitted programming exercises. They have provided a means whereby the marking load has been removed from the tutor, however this marking load has been replaced by a significant “up front” effort required in the setting up of exercises and comments.

The CAP peer-assessment tool (Davies, 2000) has been used for the past four years as a means of assessing students in the development of essays/reports within the field of computing. The aim of this study is to integrate the recognised benefits of peer-assessment within the teaching of computer programming. A grant from the University of Glamorgan’s Teaching & Learning fund was obtained April 2003, and this was used to employ two Post-HND students in order to develop an add-on to the Coursemarker © system that would provide a peer-marking environment to support and assess programming skills.

Peer-Assessment Module Design

The requirements of the peer-assessment add-on to the system are:

- Ability to create a client environment for marking
- A store for a pre-defined comment bank per exercise (including associated weighting grades per comment) and an area to hold peer free-text comments
- Ability for a student to be able to receive peer-feedback
- A method of qualitatively assessing the peer-marking process
- A process facilitating anonymous discussion between marker and marked (Davies, 2003) to address issues that may arise from the peer-marking process

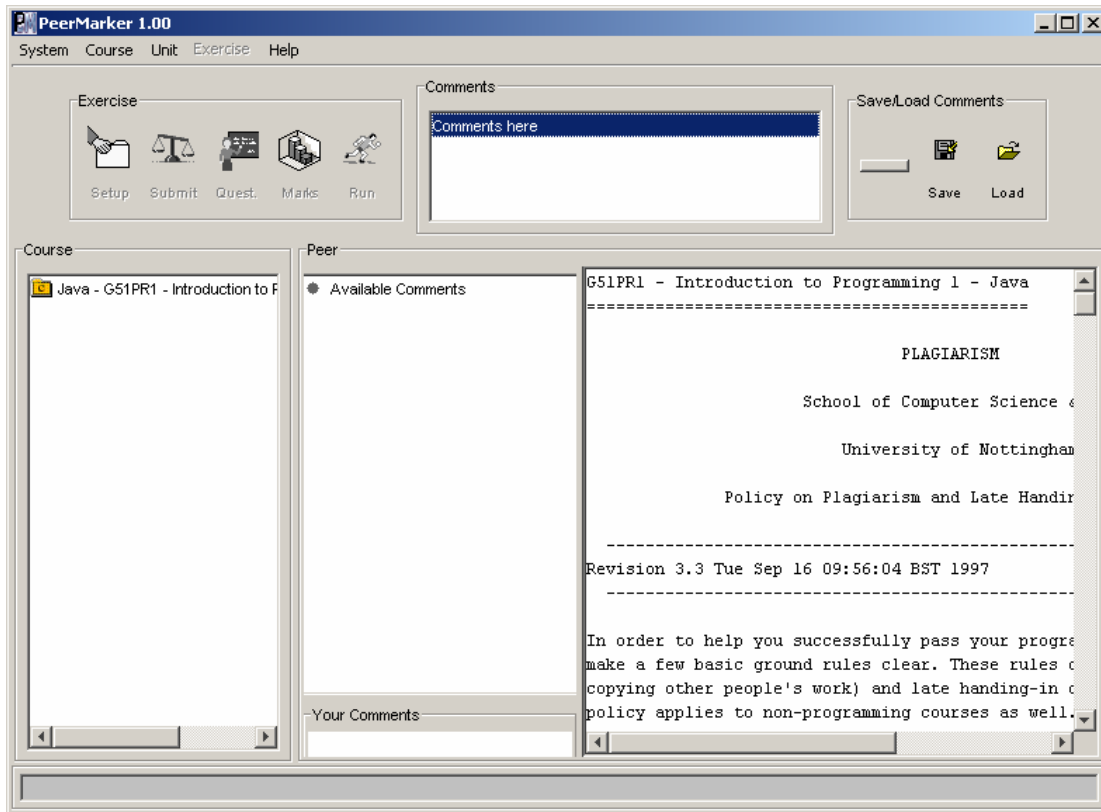
The following shows the stages involved in the creation, marking, peer-assessment and assignment of the mark for the ‘marks for marking’ process.

Initially an exercise is created by the tutor. This includes the setting up of a bank of comments relating to the work to be submitted. Each of these comments has a referential weighted mark associated with it e.g. a comment of “poor structure” would be given a -3, whereas the comment “excellent structure” could be worth a +3.

Once the exercise, marking scheme, and comments bank have been created, the exercise is presented to the student. The student makes use of the Coursemarker © system in writing a programme to meet the needs of the exercise and submit it as usual (described above). This is then marked by the

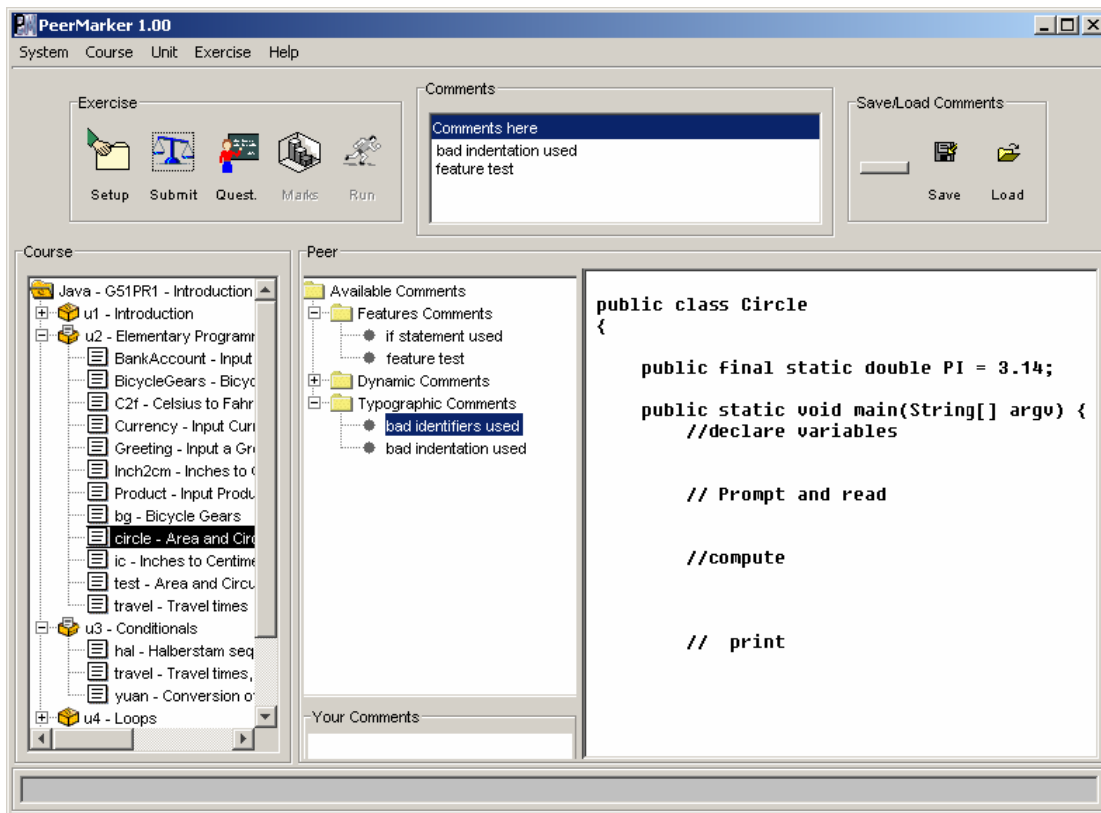
system. This iterative process of submission and feedback may be attempted a tutor defined number of times, typically between three and five times. Once a final programme has been submitted, the next stage is for the students to peer-assess the programmes of their peers (figure one).

Figure 1



A student will utilise the peer-client and request a carefully selected student programme for marking. The marking student will then assess the programme making use of the pre-defined comments bank, and also make some free-text comments concerning their perceptions of the quality of the programme (figure two).

Figure 2



Having peer-marked a tutor pre-set number of programmes (typically between six and eight), the student is then permitted to look at the peer-comments provided for their own work.

The judgment of how well a student has performed the peer-marking process can now be equated e.g. the comments the peer provided for the marking are compared against the Coursemarker © comments, and the weightings of the comments are mapped against the model set of comments. This difference, if any, can be used to calculate the ‘mark for marking’ process.

It was decided from the outset of the study to aim for a graduated introduction of the peer-assessment process. The development necessitated

the production of a new client peer interface to support the peer-marking and the peer-feedback processes. At the server end a new peer-server process was designed to work in tandem with the existing Coursemarker © server environment. The discussion and automated reported elements of the system are to be developed in the future.

Sample Study

It has been decided that for the first application of the peer-assessment system a small set of volunteers would be utilised (with a range of abilities). These students will then be presented with a programming exercise, and will be permitted to utilise the Coursemarker © system to develop and test their solution. They will then peer-mark the work of a selection of their peers. Having accomplished this exercise they will then review the peer feedback comments. Having completed these tasks they are permitted to amend their original proposed solution and submit it again to the Coursemarker © system for assessment purposes.

Future Work

The peer system is to be enhanced to include aspects such as:

- Providing a full reporting structure including the weighting mark aspect in providing a mark for marking
- Permit re-marking by peer if required
- Encourage anonymous peer discussion for clarification, especially in the free-text responses.
- Need for weighting of high to low markers in order to ensure that each student will peer-assess a full range of student work.

This “add-on” tool, when fully developed, will be capable of fully assessing a student’s ability in not only creating a programme of their own, but also in assessing the quality of their peers’ work. The development of evaluative skills in assessing the work of their peers will eventually lead to an enhancement of their own self reflective skills and critical awareness.

Conclusions

Obviously with such a small number of students, and the use of a single exercise, the results of this study cannot be deemed to produce definitive outcomes. However, the success of the implementation of this “add on” has opened up the role that peer-assessment can play both in the assessment of programming skills, and perhaps more importantly, in tutoring students in the skills of computer programming.

Acknowledgments

The authors would like to acknowledge the work of Scott Davies and Richie Mabbett for their efforts in developing this add-on. Their work has been of the highest standard, and their commitment has been beyond question throughout. Also our thanks go to the Coursemarker © development team at Nottingham University for their support to this study in providing the existing Coursemarker © source code.

References

- Benford, S., Burke, E., Foxley, E., Gutteridge, N, Zin, A.M (1994), *The Ceilidh System: A General Overview*, Learning Technology Research, Computer Science Department, Nottingham University.
- Boud, D., Cohen, R., & Sampson, J. (1999) Peer Learning and Assessment, *Assessment and Evaluation in Higher Education*, 24, 4, pp 413-426.
- Davies, P. (2000), Computerized Peer Assessment, *Innovations in Education and Training International*, Vol 37, No 4 pp 346-355.
- Davies, P (2003), Closing the Communications Loop on the Computerized Peer-Assessment of Essays, *ALT-J*, 11, 1, 41-54.
- Davies, P (2003a), Peer-Assessment: No marks required just feedback? – Evaluating the Quality of Computerized Peer-Feedback compared with Computerized Peer-Marking, in Cook, J and McConnell, D (eds). *Communities of Practice, Research Proceedings of the 10th Association for Learning Technology Conference (ALT-C 2003). Held in 8-10 Sept 2003, The University of Sheffield and Sheffield Hallam University, UK.*
- Falchikov, N. (1995), Peer feedback marking: developing peer assessment, *Innovations in Education and Training International*, 32, pp 175-187.
- Foxley, E., Higgins, C., Tsintsifas, A., Symoniedes, P. (2000), *The Ceilidh-CourseMaster System An Introduction* JICC Conference January 2000.
- Lewis, S.F. (1997), “*Developing a Modula 2 course for Ceilidh*,” CTI Computing, Dublin.
- Lewis, S.F. (2002) “*Java - a complete course*”, Continuum Books, ISBN 0-8264-5927-7.

McDowell, C, Werner, L, Bullock, H & Fernald, J (2002), The Effects of Pair-Programming on Performance in an Introductory Programming Course, *SIGCSE'02, Covington Kentucky, ACM*.

Salomon, G. (1993), *Distributed Cognitions: Psychological and Educational Considerations*. Cambridge Press.

Sitthiworachart, J & Joy, M (2003), Deepening Computer Programming Skills by Using Web-Based Peer Assessment, *4th Annual LTSN-ICS Conference, NUI Galway*.