

An interactive learning environment for teaching and learning of computer programming

Sheung-On Choy, Sin-Chun Ng
School of Science and Technology
The Open University of Hong Kong, Hong Kong
E-mail: sochoy@ouhk.edu.hk

Abstract

This paper is to investigate how new technology can be most appropriately applied in the service of teaching computer programming at a distance. Main elements include: building an online system that provides an integrated environment for students doing programming activities and coursework; how to provide students with timely and useful feedback about programming activities without the need to have instructors and students meet at the same time and the same place; how to build effective instructor-to-student interaction; and how programming assignments can be handled and managed in a much more efficient and effective manner than the existing ways.

1. Introduction

We consider the current technology is advanced enough to improve the service and increase the efficiency of teaching computer programming at a distance. The point is how to make good uses of technology to improve the teaching quality. The main characteristics of distance education are that both teaching and learning processes are asynchronous and distributed, and the learners are heterogeneous. The structure of our electronically networked environment closely matches these characteristics. By taking advantage of the capabilities of the Internet as much as possible, the teaching in an asynchronous learning environment can surely be improved in terms of both learning effectiveness and cost effectiveness.

2. Background

Teaching computer programming has never been a straightforward process. Programming is a skill that requires practice. Therefore, practical activities play an important role in the teaching and learning process [1]. In a traditional university education setting where students can often meet their fellow students and instructors, the teaching process is significantly

improved by conducting practical workshops in which teachers are present to serve the learning needs. Students can also interact with their fellow students in a computer laboratory to enhance the learning process.

However, in a distance education setting, the education process as a whole is distributed. The main actors in the education process — students, tutors, and faculty members — are geographically and temporally remote from each other. The university's academic faculty staff work on campus during office hours, while both students and tutors normally work at home during non-office hours. Distance education students normally spend most of their time working alone, have only telephone or email contact with tutors, and have little or no contact with fellow students.

In our university, students who take programming courses will receive a set of study materials that consists of traditional textbooks and printed study units. To enable students to carry out programming activities at home, a set of software including a compiler and program development environment is also provided. To complement the learning process, optional face-to-face tutorials and surgery sessions are provided and conducted by the appointed tutors. Students are also required to submit several programming assignments to their tutors during the study. Tutors then mark the assignments and provide feedback to students.

3. Design Requirement

One key challenge in teaching programming at a distance is how to provide students with timely and useful feedback about practical activities without the need to have tutors and students meet frequently in a computer laboratory, or meet at the same time and the same place, neither of which is viable for a distance education environment. The existing ways to facilitate tutor-to-student interactions are not adequate for teaching programming. From a student's perspective, when a student encounters a problem while doing the

practical activities, it is also difficult for the student to express the problem clearly in a textual message or vocally to the tutors. It is also difficult for a tutor to answer a student's problems without watching the student's computer screen.

Another challenge in teaching programming is about assignment handling, which includes receiving, grading and giving feedback to student submissions. Currently, students submit both softcopy and hardcopy of their program sources for assessment. Tutors then look at the program sources, compile them and execute them. This marking process is usually time-consuming. Another common problem is: a program that works fine in the student's machine may not work in the tutor's machine for some reasons.

The design of our system is to tackle the above challenges based on current Internet computing technology so as to enhance the quality and improve the efficiency in the service of teaching computer programming in a distance education environment.

4. The Online Interactive Learning System

The following outlines the major components of the proposed online system that enhances and improves the teaching and learning of programming in a distance education environment.

An online environment for practical programming activities: From a student's perspective, it is desirable to have a system that provides an online and web-based (i.e., anytime and anywhere) workplace for performing the programming activities. The system provides an online and integrated environment for writing, compiling and testing of programs. Moreover, the system's file server stores all the student's intermediate and submitted works. Therefore, the system enables students to do programming activities in a convenient way, without worrying about the locations of their coursework and whether the computers they are going to use are capable of performing the activities. Deploying and launching a specific user application via a Web browser, on any platform, and from anywhere connected to the Internet is now possible.

An asynchronous system for effective student and tutor interactions specific to programming activity: Building effective interactions among students and tutors is important in the learning process, especially in learning programming. However, this is also not an easy task, particularly when students and tutors are geographically and temporally remote from each other. The system is capable of accessing a database that

stores students' work and logs their behavior while doing the programming activities online.

Software agents for observing and analyzing students work and their interactions: Students use an online integrated environment for doing programming activities, and their work and performance are all stored in a centralized database. Tutors and students interact with each other via an asynchronous interaction system. The database and the interaction system provide a good source of information that is important for the faculty member to analyze student progress and the effectiveness of the learning. However, it is also difficult and time-consuming for the faculty member to monitor, collect, and analyze information from the system, which contains huge amounts of data from a great number of students and tutors. To aid the faculty member in this area, software agents [2] are constructed to observe the system, to extract information from students' work and interactions, and to perform some analysis and send the result to the faculty member.

A system for effective and efficient management of programming assignments: Electronic submission and marking of general assignments are attractive and not new. However, our focus is on handling programming assignments, and we aim at reducing the workloads of both markers and faculty while providing useful and timely feedback to students. The system also provides other functionality that assists the grading process. This includes automatic checking, compilation, testing, and plagiarism detection of submitted programming assignments.

5. Conclusions

An interactive system that enhances and improves the service of teaching computer programming at a distance is proposed. The system provides an integrated environment for students to do programming activities at anytime and anywhere. It also allows instructors to effectively monitor the learning progress of students and provide timely feedbacks to them.

6. References

- [1] Thomas, P. G. and Paine, C. B. (2000). How student learn to program: Observation of practical work based on tasks completed. Research Report 2000/03, Department of Computing, The Open University, United Kingdom.
- [2] Johnson, L. and Shaw, E. (1997). Using Agents to Overcome Deficiencies in Web-Based Courseware. AI-ED'97 workshop on intelligent education systems on the world wide web.