Overview of tools and innovations on teaching introductory programming, data structures and algorithms in Helsinki University of Technology

Otto Seppälä, Lauri Malmi
oseppala@cs.hut.fi, lma@cs.hut.fi

Helsinki University of Technology (HUT) is the largest university of technology in Finland, with yearly enrollment of over 1300 students. Only less than 150 of these study computer science as their major, but a vast majority of the others take at least one course related to programming and data structures and algorithms. Correspondingly there are several courses with an enrollment of well over 500 students.

To be able to provide a reasonable number of exercises, and give feedback on students' solutions we have heavily applied automatic assessment tools over of period of more than 10 years. This abstract summarizes the main goals and achievements reached during this period.

1 Data structures and algorithms

The yearly enrollment on the data structures and algorithms course (DSA) is today over 500-700 students. Although the numbers were only around 350 students in the beginning of the 1990's, the number of exercises we wanted each student to solve (20-25) made the total number of submitted solutions rather big. To be able to assess and give some feedback on the student solutions in a reasonable time, the TRAKLA system was conceived in 1991.

Typically a TRAKLA exercise requested the student to execute an algorithm by hand on a given input data, that is, to draw diagrams which showed how the contents of the data structure changed during the execution. For example, "Insert the following keys into an initially empty AVL tree and show the intermediate states of the structure, S Y X A F J G L U W". The answer was written into a text file and sent by email to the TRAKLA server for assessment. Assessing the answer was based on comparing student's solution with a model answer generated by an actual implementation of the algorithm. Due to the automatic assessment we could personalize the exercises. Each student had different input data for the same exercise; thus plagiarism was effectively prevented. Moreover, since 1997 the students had a possibility to get immediate feedback on their solution, and resubmit it a few times after rethinking the problem. The nature of the problems did not give them an opportunity to solve the exercises using a dummy trial-and-error method.

The TRAKLA system was a success. With more than 25 different exercises it covered most central algorithms in basic structures, sorting, searching and graphs. During the 12 years the system was in use, two thirds of the students did enough exercises to give them 80% of the maximum available. The students also gave positive feedback on the system. TRAKLA was in production use until 2003. During this time it saved enormous amounts of work by assessing tens of thousands of solutions on each course.

TRAKLA2 and the Matrix Framework

TRAKLA adopted visual representation of data structures in 1997 allowing students to build their solutions using visual tools through a GUI in the web. We call such a process visual algorithm simulation. In 1999-2000 a totally new framework, called MATRIX was built to better support visual algorithm simulation and algorithm animation. Based on this framework, a new exercise system, TRAKLA2 was built in 2002-2003.

The main features of the new system are the following. First, TRAKLA2 exercises are applets in which the exercise is solved in terms of visual algorithm simulation. Second, the system stores the whole simulation sequence and gives feedback on its correctness when the student requested grading. Third, after grading the student can either submit the solution and grading results into the course data base, or reinitialize the exercises and start solving the problem with new input data. The number

of resubmissions is not limited, since the input data is changed every time. Fourth, the student can any time request model solution, which is presented as algorithm animation. Thereafter, however, the solution with current input data cannot be submitted any more, but a new instance of the exercise has to be initialized.

TRAKLA2 has been used in production use in our DSA course since 2003. In 2004 it was also adopted for use at the University of Turku in Finland. The results have been very good. For example, in HUT 30 percent of the students got full points out of the 26 different exercises in 2004.[1] Moreover, student feedback has been very positive both in HUT and in Turku.

Another tool built on the MATRIX framework is the lecturing tool called MATRIXPRO. With it the teacher can create algorithm animations and images for webpages and publications in terms of visual algorithm simulation. The main advantage here is that no coding of needed to produce animations; thus the teacher can show the behaviour of various data structures and algorithms on-the-fly on the lectures. MATRIXPRO was in lecture use in Spring 2004 on the DSA course and the student feedback was encouraging.

2 Automatic assessment systems on programming courses

Ceilidh and Goblin

In HUT we have used automatic assessment of programming exercises since 1994. The first system to be used was CEILIDH, originally developed at the University of Nottignham, UK. The system was used to check the correct functioning of programs written in C and Fortran and later on in Java. The assessment was based on running a series of tests on the target program and evaluating the correctness of the program output.

Over 900 students pass the basic programming course exercises each year. Using an automatic assessment system has given us a possibility to set up a large number of small to medium size programming assignments, all of which are graded fairly. Such an aid is invaluable for our programming education, since afterall, it is impossible to build up programming skills without actually writing code.

CEILIDH was developed locally through the years into a point where there was quite little of the original system left. In 2004 it was finally replaced by a totally new automatic assessment system, GOBLIN, developed at HUT (albeit not by us). The new system has more support for using web and, as overall it is better maintainable than our version of CEILIDH. GOBLIN has successfully been in use on the major programming courses.

Scheme-Robo

Finally, we have still a third automatic assessment system, built specifically for assessing programming assignments made in Scheme. This system, SCHEME-ROBO, has been used since 1999 by around 300 CS major students per year doing 60 exercises. The exercises request the student either to write Scheme code or short verbal answers. SCHEME-ROBO is not restricted to checking only textual output, but can also compare data structures and inspect program code structures, a feature we do not have in CEILIDH or GOBLIN. Thus, it is possible to set up exercises, where specific Scheme functions must be used or must not be used.

3 Problem Based Learning in introductory programming

The large courses in computer programming are based on traditional education methods, i.e., lectures, closed exercises and homework exercises including a programming project.

In 1999 a totally new curriculum, called Information Networks, was launched in HUT. Due to its small size, 35 students per year, the curriculum allowed for testing a totally new approach on teaching

[1] Full points are not required to get best course grade.

programming. We started to apply the problem-based learning method (7-step method) on the course. The traditional lectures have been replaced by weekly cases, where the students analyse and discuss problems related to programming. The groups of 6-8 students and a tutor convene weekly for 2-3 hour sessions. They analyze the cases and decide what are their learning goals so that they could understand the problem better. Each member in the group studies to meet the same goals and the group reconvenes after a week to discuss what they have learned. In addition to this, the students, of course, solve practical programming exercises and do a programming project. Finally they gather the knowledge and experience with reflection into a portfolio.

Since the first year, 1999, the results have been promising: the dropout rate on the course has been very low compared to that of the traditional course. The students are motivated and eager to solve voluntary assignments.