VIOPE – Computer Supported Environment for Learning Programming Languages

(Extended Extract)

Esa Vihtonen
Viope Solutions Ltd,
Laserkatu 6, 53851 Lappeenranta, Finland
E-mail: esa.vihtonen@viope.com

**Eugene Ageenko*
Department of Computer Science,
University of Joensuu, PB 111, 80101 Joensuu, Finland
E-mail: ageenko@cs.joensuu.fi

Résumé

Le programme d'apprentissage des langues est aussi difficile pour l'étudiant que pour le professeur. Le système informatique proposé offre un environnement d'apprentissage indépendant des contraintes temporelles -et spatiales- permettant de faire une évaluation automatique du travail de l'étudiant via internet. Il facilite les procédures d'exercices d'évaluation pour les professeurs, fournit un enseignement de haute qualité et des cours développés. Le système peut être correctement utilisé sur le programme de cours de langues enseigné dans les Universités. Il peut également être transférable à des solutions d'apprentissage indépendante : cours d'été, collège, particuliers…

Introduction

Teaching programming languages is a challenging task. It is not sufficient to describe the syntactic of the programming languages only, but on the contrary, actual writing of the programs is a key factor in the development of programming skills. Due to various reasons, such as lack of free time because of work obligations, or dissemination of participating students over a large territory, it is not always possible to arrange contact teaching. Inability to attend some classroom sessions often results in a lag in education right up to quitting the course. One-to-one contact teaching is expensive and often impossible due to high workload of the instructors or legal regulations in European countries.

Along with a standard contact (classroom) teaching, more and more various computer-based tools are used in the teaching process nowadays. These include, but not limited to *Program Visualization Tools* used by the instructor in the classroom and the *Course Management* and *Learning Environment Systems*. These systems facilitate course counseling, content delivery, student collaboration, and study assessment. Students basically can plan their studies, access course material at suitable time, be aware of major happenings in the class (new assignments, dead-lines, exam dates), discuss course-related topics and participate in collaborative group work, submit their home work to the instructor and receive the grade and feed-back. The systems usually have web-based interface, which is easily accessible from various location, such as distantly located schools or student homes.

A major problem related to such kind of virtual classes is the greatly increased workload for the instructor. Even though the student without the instructor intervention can access the course content once developed, the number of exercise solutions that has to be evaluated can be huge. There is a great challenge for the instructor to check and give weekly feedback for 500 individual solutions, in a typical course with 125 students and 4 exercises per week. Often it requires hiring temporal instructors from the graduate students of university, whose pedagogical abilities and work experience are at times questionable. The development of automatic study assessment tools for exercise evaluation is therefore highly desirable.

Virtual learning Environment

Here we present VIOPE (Finnish *VIrtuaali OPEtus* – Virtual Learning) that is a computer-based interactive learning environment for learning and automatic assessment of the programming languages. In a nutshell, VIOPE can provide programming related content for the students according to student's skill level. For example, VIOPE Java covers the fundamentals of Java programming. Content is designed for the students without previous experience on programming. Course begins from the very basic and advance to inheritance and applet topics. The system is able to assess student made exercises automatically, to collect data about student results, and provide teacher with a vivid progress report.

Users access the system via web-based graphical interface using a standard web browser. There are two user interfaces in VIOPE environment: for students and teachers respectively. Both interfaces have similar and very accessible design. Using student interface, the student has to answer multi-case questions and write correct programs in order to progress to the next lesson. After questions in certain lesson are answered (either correctly or wrongly), student can review the answers and advance to the exercises. Programming exercises are assessed automatically so that the students have to make working programs that can solve the given problems.

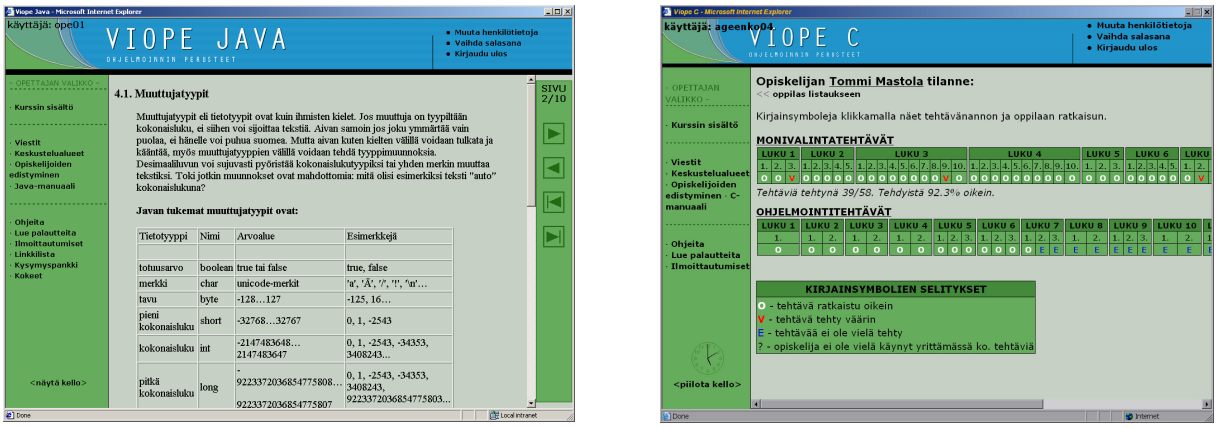

Figure. Sample views for the student (left) and teacher (right) interfaces for VIOPE. Student screen shows lesson material, and teacher's screen shows student progress information.

VIOPE collects information about the progress of the student's study. The teacher can examine this information by reviewing student's answers and exercise solutions. Teacher can also follow the group progress in a visual form by examining the progress table that depicts distribution of students per lesson their currently attending. Progress table helps teacher to identify the most difficult topics and be able to adapt the lecture content accordingly.

VIOPE environment has also additional features supporting both learning and teaching: discussion forum, internal messaging system, programming language manual and help pages.

Automatic Response Technology

Programming exercises in VIOPE are assessed using *ART (Automatic Response Technology)*. Each time student sends the solution (computer program) for a certain exercise, ART examines the code and gives student a feedback. If the program has syntactic errors, the feedback is oriented toward correcting these errors. Problematic area in the program is highlighted graphically to assist the student finding the errors.

Syntactically correct programs are assessed in order to check if the given answer solves the problem specified in the assignment. For that, ART is able to run student program on a virtual machine and compare the program output with the one outlined in exercise. ART is also able to check if the program is done using the certain structure, e.g. for-loop. If the program will not do what it is supposed to do, ART produces feedback with outputs from students program and working example.

Discussion and Conclusions

The proposed VIOPE learning environment provides time- and place- independent environment for learning the programming skills. An individual student can follow own timetable and advance in the course by working out the required exercises. Web-based interface makes the environment easily accessible from any computer connected to the Internet. The automatic assessment system relives the instructor from time-consuming evaluation of exercises. Visual representation of the student's individual and mutual progress helps teacher to identify the most difficult topics.

According to our observation, students are often unsatisfied with the course progression and timing. When one part of the students considers some topic being boring, another requires its prolonged and thorough study. The proposed system alleviates time constraints and makes it possible for the student to advance quickly in some chapters, and concentrate deeply in another. The system can also offer extra possibilities for the students, who are for certain reasons unable to attend classroom sessions, or stay within the strict course schedule.

Another advantage of VIOPE system comes from ability to practice programming via Internet with error feedback for both syntax and semantic errors. This functionality gives new possibilities for the teachers as the time saved from routine exercise checking, detailed information on student group progress, and easy-to-use tools for closer observing of results.

These innovations greatly help the teacher to improve the course content by adapting to the student needs. The system can be efficiently used as an augmentation tool for a classroom-based teaching, or as a standalone solution for short introductory courses for a high school or university students, or interested individuals.