

VIOPE – Computer Supported Environment for Learning Programming Languages

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...

Abstract

Learning programming languages is a demanding task for both the student and the teacher. The proposed computer-based system offers time- and place- independent learning environment capable for automatic assessment of student work via the Internet. It relieves teachers from time-consuming procedure of exercise evaluation, and provides for higher quality teaching and course development. The system can be efficiently used within the existing programming languages courses taught in Universities. It is also suitable as a standalone solution for short introductory programming courses for summer schools, high schools or interested individuals.

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 (Ben-Ariet al. 2000, Kölling 1999), and *Course Management and Learning Environment Systems* (WebCT 2002, Chan et al. 2000,

Haataja et al. 2001). 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 typical example of such a virtual environment is the system developed within the *Virtual Certificate* project ongoing during years 2001-2003 and funded by Ministry of Education of Finland (Haataja et al. 2001). The system delivers over the Internet 15 credits of studies in the field of introductory Computer Science and Programming to high school students. Most of the courses have similar structure: students receive learning material, do weekly exercises, submit solutions electronically to the teacher, and receive feedback and grade for their works. On-line tutors work at the University. They evaluate student work and play a role of course instructors.

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 (Foxley et al. 2001, Higgins et al. 2001).

Here we present VIOPE computer-based learning environment for teaching 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

Technology (Vihtonen et al. 2002). Currently VIOPE is undergoing trial test as an augmentation tool in a "Programming Language C" course in University of Joensuu for the first year students majoring computer science.

The preliminary feedback analysis shows that the students *didn't have* or *had very little* programming experience before entering the course, and thought that programming was *quite difficult* task (answers range from *adequate* to *very difficult*). After passing the course the opinion about programming had improved (answers range from *quite easy* to *quite difficult*). Overwhelming majority found the level of exercises used in VIOPE course *adequate to their skills* and *helpful for studying*. Though many students suggested automatic assessment system *too strict*, almost everyone found feedback given by ART to be *useful*. In the final exam 73% of the students participating in VIOPE exercises passed the final examination while 38% of the students participating in traditional exercises passed the same examination (50% of students in Summer School participated into traditional exercises).

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.

The following features for VIOPE environment are currently under development: teacher's ability to add own programming exercises which can be assessed by ART; course contents based on the request from the teachers and the students; courses for various programming languages and topics; localization to major European languages.

Conclusions

The computer-based teaching system has been proposed. The system 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 relieves 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. 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.

Acknowledgements

Authors thank Prof. Arto Kaarna and Asst. Satu Alaoutinen from Lappeenranta University of Technology for experimental evaluation of the VIOPE environment in 2001 Summer School.

References

- Ben-Ari, M et al. 2000, An Extended Experiment with Jeliot 2000. In Sutinen, E. (ed.), proceedings of the First Program Visualization Workshop, Porvoo, Finland, pp.131-140.
- Kölling, M. 1999, Teaching Object Orientation with the Blue Environment, *Journal of Object-Oriented Programming*, Vol. 12 (2), pp. 14-23.
- WebCT 2002, WebCT Learning Environment WEB page at <http://www.webct.com/>
- Chan, P.-S., et al. 2000, Networked Submission and Assessment. In Proceedings of the First Annual Conference of the Learning and Teaching Support Network, Edinburgh, UK, August 2000.
- Haataja A. et al. 2001, High School Students Learning Computer Science over the Web. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, Vol. 3(2).
- Foxley, E. et al., Ceilidh a System for the Automatic Evaluation of Student Programming Work, In Proceedings of the Computer-Based Learning in Science Conference (CBLIS), Holland, July 1999.
- Higgins C. et al. 2001, Model of a Semi-Automatic Assessment Tool in Web-Based Learning Environment, In Proceedings of the Ninth International Conference on Computers in Education, Seoul, Korea, November 2001, pp. 1217-1224.
- Vihtonen E. et al. 2001, Computer Supported Learning Environment for C Programming Language. In Proceedings of the First Annual Finnish/Baltic Sea Conf. on Computer Science Education, October 2001, Univ. of Joensuu, Department of Computer Science. Report A-2002-1, pp. 27-32