DEVELOPING WEB-BASED TOOLS TO SUPPORT AND ENHANCE EDUCATION

Mária Bieliková and Pavol Návrat
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava, Slovakia

Abstract. In this paper we have attempted to contribute to the movement towards shifting more and more burden of managing the process, delivering the content and assessing the learning outcomes to the web-based tools. Among the tasks we attempted to support are for example: submission and checking of assignments, managing student projects, e-enrolment of students, formation of study plans, programming contest, multimedia presentation of the content of the Computer Architecture course, contributing to plagiarism detection, distance education in the Machine Level Programming course, or assessment of programs in C language. What is especially to be stressed is the fact that all the above mentioned projects have been developed by our students in the process of their education.

Keywords: web information system, learning management support, support of study and assessment, team project.

1. Introduction

The advent of the Web has opened a wide range of opportunities to enhance the existing educational content, process, and eventually its outcomes. We are still at the beginning of exploring all the possibilities that are becoming available provided we are able to identify them and develop proper tools that would allow making use of them. Equally important, however, we are at the beginning of evaluating the real impact of the education enhanced by the use of new information technologies.

In this paper we describe several systems for learning support. The systems described have been designed and developed as web information systems thus allowing access from anywhere and at anytime. Our aim is to present diversity of projects aimed at supporting learning. It is not our intention to present complete functionality and properties of each system.

Instead, we concentrate on several interesting features of presented systems that may possibly be of some interest for others.

Described systems were implemented using open source software such as LAMP platform (Linux, Apache web server, MySQL and PHP for implementing application logic) or Java based solutions (e.g., Tomcat Servlet Container, STRUTS framework). Using open-source technologies and systems (Open Source, Java, Apache Foundation) enables effective operation of the systems and offers free scope for prototyping.

All the described projects have been developed by our students in the process of their study within the subject "*Team project*" that is part of a postgraduate course in Software Engineering, Information Systems, and Computer Engineering and Networks degree programs at the Slovak University of Technology in Bratislava [3]. Some systems were developed in several iterations, with each version implemented in one academic year (the Team project lasts two semesters).

We found the theme of development of information systems related to learning as very suitable for the Team project. Such themes provide a context known to the students and pose high motivation for development of a usable product. Moreover, the students have a direct experience with using previous student projects, which is also a stimulator for their work. Moreover, developing web-based systems provides another dimension into the motivation together with acquiring up-to-date knowledge and skills.

The paper is organised as follows. In Sections 2 and 3 we present selected tools for learning management support. The systems were developed within the context of information infrastructure of the Faculty of Informatics and Information Technologies. Their aim was to extend capabilities provided by the university information system. This fact influenced the functionality and technologies chosen. Section 4 describes several systems aimed at support of acquiring programming skills. We present a system for organising programming contests, which can be used on-line during a class or in correspondence regime just for training. Section 5 is devoted to educational content development and presentation. Finally, we give conclusions.

2. Managing student projects

It is unthinkable not to include some project work in any course in engineering. Students of informatics or information technologies related degree programs gain during project work knowledge and practice of design and construction of various software intensive systems. Presently we successfully operate a system developed in the course of two team projects in 2003 [5], which is aimed at managing student final projects. It was built based on a prototype developed in academic year 2001/2002, which was used one year in the degree program Informatics.

The system supports the whole project life cycle. The project life cycle served as a basis for work distribution within two teams of students:

- *subsystem for project specification and assignment*: main functionality is the project specification – formulation by a project supervisor from the teaching staff, approval by a degree program director, announcing and publishing the project specification for students, project registration by students, and project assignment by supervisors,
- *subsystem for project outcome submission and assessment*: main functionality is project outcome submission by a student, and evaluation by the supervisor and by a reviewer (optional) together with announcing results of assessment to the student.

Both subsystems operate independently on a common database. The system enables organisation of project tenders, which serve for the project assignment. Besides standard functionality related to project management, it implements a generic scheme of project reviews based on evaluation of particular aspects on a predefined 8 points scale (excellent, very good, good, adequate, satisfactory, sufficient, poor, very poor, unacceptable). The project template varies for different types of projects and for different roles, e.g., supervisor and reviewer. Supervisor (or reviewer) fills an assessment for each aspect into the predefined form and the system automatically recommends an assessment based on evaluated aspects of the project and their importance (predefined in assessment template for particular type of the project). Finally, the supervisor or reviewer decides on the final mark, considering all the factors (not only those expressed by aspects in the review form).

3. E-enrolment of students

We use a web information system aimed at support of degree programs admissions [4]. The e-Application system provides an electronic application form, which brings a prospecting student into the virtual space of the university already before he actually becomes a student of the university. The system provides support of the whole admission process, which consists of the following steps: applicant registration, application submission, application processing, applicant performance evaluation, admission decision and publishing of final results. Previously used e-Application system did not support automatic evaluation of entrance exam tests. Tests were evaluated manually and results were imported into the e-Application system.

Our e-Application system has been amended this year by a separate system called SEEE (System for Entrance Exams Evaluation) [6]. It provides support for automatic evaluation of entrance tests together with effective input of data collected during the admissions and related to applicants (such as high school educational achievements not known at the time of e-Application form submission – school leaving exam results, modifications in personal data, phone number for SMS notification etc.). Entrance tests are scanned and automatically processed. To correct errors, results can be entered also manually (even more than once). The system automatically highlights conflicting results entered for particular test and the applicant.

The SEEE system is loosely coupled with the e-Application system through data import and export services. It has been successfully used in last admissions (spring 2005).

4. Support of acquiring programming skills

Learning programming requires not only well prepared textbooks but also a lot of practicing in order to develop a programming skill. For the practice, the idea of organising a programming contest seems to be an effective approach. We view the contest as a learning vehicle. The students participate in the contest and solve problems within a theme announced before the contest (to give the students time for preparation). After the contest the students are provided by solutions together with alternatives.

In order to support such contests a software system for support of programming contests was developed. The system was inspired by the ACM International Collegiate Programming Contest rules[1]. Several systems for the ACM ICPC exist. However, we needed a system that can be used also in programming courses and would support several modes of operation.

The system was developed in several iterations and several teams of students participated. First version was developed in academic year 2002/2003 [2] based on a prototype from 2001/2002. It was used during the ProFIIT contest organised for talented high school students (organised for the first time in spring 2004). This academic year (i.e., 2004/2005), an improved version was developed within the Team project [1].

The system supports various duration lengths of contests. It can be used also for an off-line correspondence contest, which takes several weeks and students solve tasks and submit solutions from anywhere, e.g. from their homes. This feature enables the usage in several modes: training mode, which is organised as distributed correspondence contest and assessment

[1] Main page of the ACM ICPC is available at http://acm.baylor.edu/acmicpc/.

mode, which is organised as on-line contest where the students solve problems in controlled environment with necessary security arrangements (mainly related to restrictions of communication).

It implements the following roles (roles in the brackets correspond to roles in the learning process):

- *principal referee (teacher)*: manages particular contest, she is the highest authority in the contest, who invites the referees to the contest and makes decisions about the contestants,
- *referee (teacher)*: manages and inspects the automatic evaluation of solved tasks and communicates with contestants,
- *contest author (teacher)*: creates particular contest together with definition of problems,
- *contestant (student)*: solves given tasks, solved tasks sends to referees,
- *public user (guest)*: monitors results of the contests,
- *administrator*: insures regular operation of the system and data back-up and archiving.

One of the basic tasks of the system is a collection and automatic evaluation of programs – solutions of given tasks. Automatic evaluation is based on input-output specification. Large sets of input-output pairs are prepared for respective problems, which enable evaluating not only input-output behaviour but also effectiveness of the implemented algorithm. A student can submit several solutions of one task within the time limit. As the solving time is important in the evaluation, the system marks each submitted solution with a time stamp. Communication is implemented as an on-line chat, where students can put questions and teachers produce answers viewable by all students taking part in the contest or by just the particular student.

Support of learning programming was also a theme of several other projects. Some features of described programming contest system are also a part of the Charon system [9], which serves for submission and assessment of programming tasks of various types. It supports assignments in various subjects. This enables a student to submit an outcome from anywhere and at anytime (within a predefined schedule) and a detailed overview of all currently solved assignments. Support of program evaluation is based also on evaluation of input-output program behaviour.

The team Dagwood developed an enhancement of the Moodle system[2] with program assignment assessment [8]. Evaluation of submitted programs is specifically developed for assembler programs (the Machine Level Programming subject). A component for plagiarism detection is also developed. It is based on comparison of pairs of programs. Compared programs are at first tokenised. Granularity of tokens can be adjusted. Comparison is performed on n-tuples of tokens. Components are implemented as web services, hence usable also for other subjects.

5. Multimedia presentation of study materials

Several projects have been focused on development of e-content related to a particular educational course. The task of content development was combined in several projects with design and development of software tools for effective content maintenance specific to the particular subject.

In the academic year 2003/2004, a project oriented towards multimedia presentation of the content of the Computer Architecture course was completed [7]. The main aim was transformation of existing materials related to computer architectures into multimedia form. As a result a web portal with introductory materials to the Computer Architecture course was developed along with a software tool, which provides functionality for composition of the course materials into chapters and sections and for export a presentation into static HTML pages. This feature enables usage of materials also off-line with only a web browser available.

Theme of content development for the Machine Level Programming course presents part of several projects. The projects were oriented mainly to the evaluation of assembler programs together with plagiarism detection. Content development and its presentation on the web is just one part of these projects. They concentrate on developing animations of specific topics related to machine level programming (e.g., modelling of a interrupt subsystem of selected processors, simulation of register transfers during program execution).

6. Conclusions

The work presented here is modest by definition: we have been hesitating to to use the letter *e-* in connection with learning. Rather, we attempt to contribute to enhancing the learning process by extending the current support

[2] Moodle is a free, open source course management system. It is available at www.moodle.org.

by methods and tools that make use of information technologies. On the other hand, perhaps the question of what is e-learning, or what it should be, is not only a legitimate one, but also worthy of some discussion. We strongly advocate using methods and tools that use, or a based on advanced information technologies. However, if the goal were a new paradigm of learning, then there is more to be researched also on the pedagogy side.

References

[1] ARPÁŠ, R., DÁVID, M., HORVÁTH, E., MAŽGUT, J., SOLÍN, R., ŠÍPKA, M. (2005), *Support of programming contest*, Team project supervised by G. Kosková, Slovak University of Technology in Bratislava. (in Slovak)

[2] BABUŠČÁK, R., HAJAS, M., KRUŽLÍK, Š., RUSNÁK, M., SEBESTYÉN, J., SÍLEŠ, B. (2003), *Support of programming contest*, Team project supervised by G. Kosková, Slovak University of Technology in Bratislava. (in Slovak)

[3] BIELIKOVÁ, M., NÁVRAT, P. (2005), Experiences with Designing a Module for Teaching Teamwork to Students, *Journal of Computing and Information Technology*, Vol. 13, No. 1, 1-10.

[4] BIELIKOVÁ, M., KURUC, J., MARKO, V. (2004), Entry into Virtual University Space through Web-Based e-Application. In *Proc. of ICETA 2004*, F. Jakab, L. Samuelis, Košice, I. Sivý, M. Bučko (Eds.), pp. 403-410.

[5] BIELIKOVÁ, M., FILKORN, R., MAKULA, M. (2003), Software support of final projects management. In *Proc. of Technologies for e-learning*. B. Mannová, P. Šaloun (Eds.), Prague, pp. 22-29. (in Slovak)

[6] BRNÁK, D., HOLEČKA, M., MIHALIK, P., PAVLOVIČ, J., PETROV, M., RUŽIČKA, P. (2005), *E-admission – entry exams evaluation*, Team project supervised by J. Kuruc, Slovak University of Technology in Bratislava. (in Slovak)

[7] FEKIAČ, P., HLÁVEK, L., CHRVALA, E., JÓKAI, D., PÁTEREK, R. (2004), *Multimedia presentation for the Computer Architecture course*, Team project supervised by M. Kolesár, Slovak University of Technology in Bratislava. (in Slovak)

[8] FÜLÖP, L., LACKO, M., MALICH, I., ORFÁNUS, D., STRAKA, I. (2004), *Support of distance education in the System and machine level programming course*, Team project supervised by P. Čičák, Slovak University of Technology in Bratislava. (in Slovak)

[9] JENČO, M., GONO, P., STANKOVIČ, M., ŠNIRC, J., VNUK, M., ŽUŽO, M. (2005), *Assignment management*, Team project supervised by B. Steinmuller, Slovak University of Technology in Bratislava. (in Slovak)