

Improving Testing Abilities of a Programming Tutoring System

Boban Vesin, Aleksandra Klašnja-Milićević
Higher School of Professional Business Studies
Novi Sad, Serbia
{vesinboban, aklasnja}@yahoo.com

Mirjana Ivanović
Department of Mathematics and Informatics
Faculty of Science
Novi Sad, Serbia
mira@dmi.uns.ac.rs

Abstract— Testing of learner’s knowledge is demanding functionality of the e-learning systems. Multiple choice questions are suitable and easy to implement feature for testing the knowledge in systems for learning social studies. However, these questions are not suitable for systems for learning the basics of programming problems where solutions are not exact. In previous work we have presented the basic functionalities of the Protus system that implements Java programming course for beginners. Testing knowledge in this system is implemented through various tests with numerous multiple-choice questions. The aim of this paper is to present integration of the existing automatic assessment tool named Testovid within Protus system in order to achieve advanced testing functionalities in solving programming tasks.

Keywords— Java programming; Protus; testing; e-learning

I. INTRODUCTION

Education in computer science relies on various software for teaching, learning and course management. Introductory programming courses in particular, require a range of interactive, visualization and collaborative learning media to meet the learning demands of beginner programmers [1].

Problems and assignments in programming domain are considered as essential elements of software engineering and computer science education [2]. Programming assignments can help learners become familiar with the elements of modern programming languages, become familiar with essential tools, and to understand how the principles of software development and design can be applied.

Assessing learner’s programming knowledge and grading their solutions of programming assignments can be very tedious. Even with a small number of learners attending the course, the number of programming assignment solutions that an instructor must grade during a semester can be overwhelming. With the constant increase in the number of enrolled learners, and with continual assessment on courses with programming exercises, this problem becomes even more evident.

Automated assessment systems can help alleviate the effort of grading large amounts of similar learner’s solutions. Most of this workload can be transferred from the humans to the machine freeing instructors for more cost-effective activities in

the learning process. These systems also provide instant feedback to the learners improving their learning experience.

In last several years we have been working on implementation and usage in real educational environment a tutoring system named Protus – PRogramming TUtoring System. Due to its general concept and architecture it also could be used for various online courses [3]. This tutoring system is deeply based on principles of Learning styles recognition and content recommendation for a course personalization [4, 5]. Effective personalization of teaching materials and construction of learner model depend on the results that individual learners have been achieved with specific teaching materials. Thus, there is a need for efficient checking of acquired learners' knowledge after every learning stage.

Up to now testing knowledge in the Protus system has been implemented through various tests with numerous multiple-choice questions. But higher-quality testing in teaching programming languages and skills is necessity and unavoidable component in modern educational systems and tools. So the main intention of our current research is increasing assessment and testing functionalities of Protus tutoring system for Java programming course. New features of the system will contribute to the assessment of learners and provide a more accurate learner modeling and thus improve the personalization options of the course.

The aim of this paper is to present integration of the existing automatic assessment tool named Testovid [6] within Protus system in order to achieve advanced testing functionalities in teaching programming subjects.

Testovid assessment system allows learners to test their assignments in a controlled manner [6]. The system allows instructor to run the same tests on a set of learner’s assignments. The results of the tests are recorded in a log file and are available to both the learners and the instructor. The system accepts any file types as assignments, and the instructor has great flexibility in specifying how and what is to be tested. The system is independent on underlying platform and programming language and it has been implemented using Apache Ant. Therefore it is appropriate for providing additional functionalities to Protus system.

The goal of the implementation of new module within Protus system is to improve assessment of the learners. The system will be available to learners working on their assignments in computer labs, at home or over a network or internet. The testing logic and test data for each assignment has to be prepared by the instructors before work on assignments begin, and the data will be stored within testing system directories on the server.

This new module (in fact integrated Testovid tool) in Protus system will increase the accuracy of assessment of learners programming solutions. With new possibilities for testing knowledge, building learner model will no longer be based only on questions with multiple answers which are significant improvements in learning programming subjects. Questions with multiple answers are suitable and easy to implement feature for testing the knowledge in systems for learning social studies. However, they are not suitable for systems for learning the basics of programming problems where solutions are not exact and need additional evaluation.

The paper is organized as follows. Section 2 describes some previous research related to our implementation. Review of Protus and Testovid system are presented in Section 3. Section 4 introduces the overall system architecture and describes the proposed method for integrating assessment system within tutoring system. Finally, Section 5 provides the concluding remarks.

II. RELATED WORK

The developers of modern e-learning courses attempt to offer learners more interactive and engaging content, which goes beyond a simple set of static pages. Most frequently they chose to enhance course content with interactive problems of various kinds, from simple questions, to programming exercises [7].

Systems that automatically assess learner's programming assignments have been designed and used for over forty years. Many systems that objectively test programming abilities of learners and perform programming assessment were developed. Authors in [2] reviews a number of influential automatic assessment systems, including descriptions of the earliest systems, and presents some of the most recent developments. The paper presents a review of a number of projects that automatically assess learner programming assignments using a test-based approach. Authors stated that modern systems make use of contemporary web-based technologies, and sometimes provide additional support for educators in the form of assessment management and reporting facilities. Some systems also offer tools directly to learner via web browsers [2, 8, 9, 10].

Authors in [11] presented a systematic literature review of the recent (2006–2010) development of automatic assessment tools for programming exercises. They discussed the major features these tools provide. They also present approaches used in specific tools, from the pedagogical and the technical point of view. A majority of the systems are either targeted only for Java or have support for Java. This fits well with the trend of Java being one of the most used introductory programming

language. Other popular languages supported by the systems include C/C++, Python, and Pascal

Authors in [7] developed QuizJET (Java Evaluation Toolkit), a system for authoring, delivery, and evaluation of parameterized questions for Java. System has been designed to support Web-based authoring and for delivery and evaluation of parameterized questions for Java programming language. QuizJET can be used for assessment and self-assessment of learners' knowledge on broad range of Java topics from language basics to advanced concepts, such as polymorphism, inheritance, and exceptions. Although the huge effort was done in implementation of the system, questions used in their studies were relatively simple.

Introductory programming course named CALMS (Computing Augmented Learning Management System) was presented in [1]. It is ancestor of a collaborative test management subsystem for online learning system named Socrates. It applies question and test construction so that communities of learners can progressively create, edit and update individual questions and tests while maintaining a history of the changes made. Automatic assessment with plagiarism detection is only envisioned for the future in the paper.

The main disadvantage of all these systems are restrictions what can be assessed automatically, that is, only clearly defined questions with completely specified interface for the overall solution. Moreover, unless explicitly specified as a necessary component to the solution, an assessment engines cannot award additional marks for creative design or innovative solutions. Another problem with assessments systems is that they are mostly not programming language independent [6].

Functionalities of Testovid will provide Protus system with advanced assessment features and improve process of learner modeling. Personalization performed in Protus will no longer be run only on the basis on tests with multiple-choice questions, but more advanced methods for testing knowledge will be used for assessing learners complete programming solutions.

III. REVIEW OF EXISTING COMPONENTS

The main motivation for this research is to produce effective and efficient environment for successful and easy learning and testing of basic principles of programming languages especially Java language in the moment. The crucial idea is to include and integrate Testovid system in Protus tutoring system that offers personalized learning of essential Java concepts.

A. Protus

Protus is a tutoring system designed to help learners in learning essentials of programming languages [3]. It is an interactive system that allows learners to use tutorial courses and also to test knowledge they acquired during learning sessions (Fig 1). The first course realized and implemented in the system was Java programming introductory course [5]. The environment is designed for learners with no programming experience. The main purpose of the Protus system is to

recommend useful and interesting materials to e-learners based on their different backgrounds, preferences, learning purposes and other meaningful attributes.

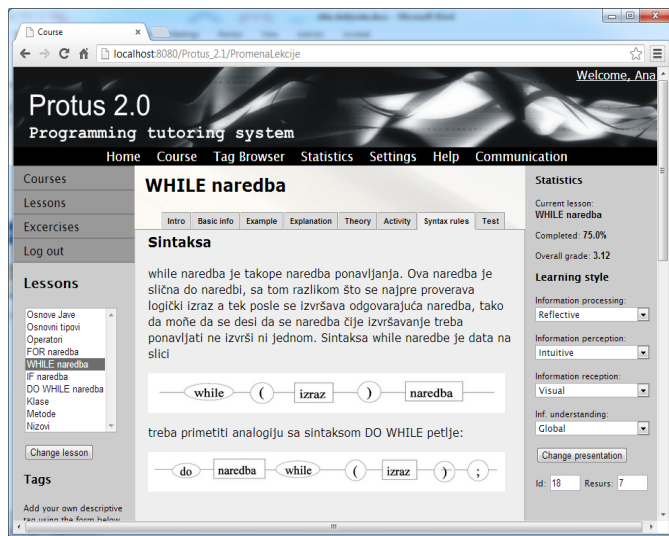


Fig. 1. Protus system

The Protus fulfills three primary goals. The first goal is to provide intelligent tutoring system with web interface. The second goal is to provide the teachers with useful reports identifying the strengths and weaknesses of learner's learning process. Finally, the third goal is to provide a rapid development tool for creating basic elements of tutoring system: tutorials and learning resources. Two separated user interfaces are provided for both student (learner) and instructor (learner's mentor). All data about learner and his progress in the course, as well as data about tutorials, tests and examples are stored in the system's server.

B. Testovid

Testovid system is a general automatic assessment tool that supports testing of any aspect of learner solutions written in any programming language. The main motivation for developing the system was to check learner programs for compilation errors, code style guidelines adherence, implementation correctness and performance.

Furthermore, Testovid system incorporates hints and advices into the testing reports, enabling graders to give learners comprehensible feedback about their programming solutions [12]. This feature of the system fits naturally into idea of using software to provide learners with rich information about misconceptions and errors in their programming solutions.

The main advantage of this automatic assessment tool is that it is built on Apache Ant and thus it is modern and not dependent on programming language or specific building and compilation logic. Also, the system can be used in a wide variety of situations and environments, and is very extensible, modular, and can quickly adapt to new trends.

IV. ARCHITECTURE OF NEW INTEGRATED SYSTEM

In this section, organization of components and their functionalities within integrated system will be described. The primary motivation for integration of mentioned components was to build multi-functional efficient and effective educational system for learning essentials of Java programming language that will allow learners new functionalities. Original version of Protus system allows to a learner:

- go through adequate tutorials,
- test their understanding of new material with series of test with multiple-choice questions.

Integrated version of Protus system brings new functionalities for a learner:

- do on-line programming,
- self-assessment programming tasks during course and final assessment of gained programming skills,
- submit their programs/solutions for programming tasks and receive automated feedback and errors report,
- have final exams via Internet or in computer laboratory.

The original architecture of Protus is modular and it is designed to be extensible with additional modules [5]. Although only Java programming course was implemented, Protus system provides possibilities of adding new courses from other domains. To tailor functionalities that best fit the circumstances, system is also designed for easy replacing existing modules or adding new modules that will provide additional functionalities. It is especially important when a new, better, more efficient and productive module is developed and old-fashioned module existing in the system can be smoothly replaced by new one. This primarily relates to the module for generating recommendations that can be extended with new functionalities and thus allow the implementation of various new options for personalization to meet specific needs and styles of individual learners.

Additional modules can be constructed on top of this infrastructure. These modules could provide more specific or even completely new functionalities. For example, a tutoring system can be improved with use of existing assessment tool - Testovid. Protus can use results and grades of a learner provided by this tool to improve course and help learner in the learning process. New assessment module won't replace any of existing modules but it will be integrated in the system as a new component. Therefore, Protus after integration will provide not only tests with multiple choice questions but additional assessment possibilities for programming solutions.

The architecture of the Protus system with integrated new assessment module will consist of three main parts (Fig 2):

- User interface for learner that presents basic functionalities to learner via web browser. This module will be extended with pages for automatic assessments of learner knowledge and overall progress.

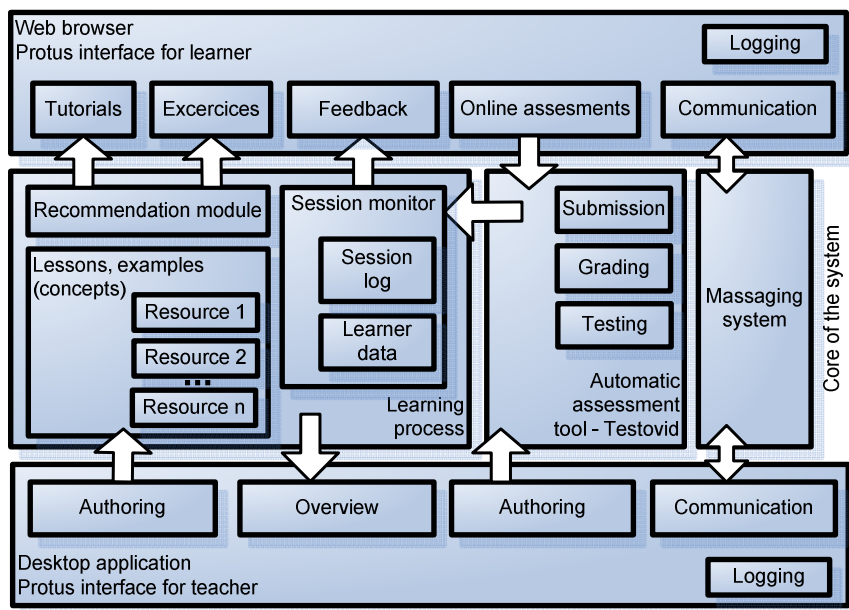


Fig. 2. An architecture of integrated system

- User interface for teacher that allows him to monitor learners, track their progress through course, generate new teaching material, use newly integrated automatic assessment tool, etc.
- Core of the system that basically presents original architecture of Protus expanded with module for automatic assessment of learners' programming solutions. This module will be covered in more detail in next session.

A. System's core

- Present learners with a personalized learning material with use of integrated recommendation system.
- Monitor user sessions, forms learner model and generate feedback to learners based on the results of assessment module. With additional assessment features inherited from Testovid, learner module will contain more precise data about learners and therefore personalisation options will be improved.
- Provide messaging system that allows easy communication between learners and teachers,
- Provide modules for the evaluation of learners' programming abilities and knowledge with use of automatic assessment tool.

Automatic assessment tool is basically integrated Testovid system and it will be used for learner's assessment. There is no need for extensive adaptation of Protus system. It is only necessary to add user interface elements that will allow invoking of Testovid services, and enable submitting of learners' solutions.

B. Assessment module:

Assessment module will be used for assessment of learners' programming skills on a wide range of Java topics from programming basics to advanced concepts such as concurrency, interfaces and inheritance. Therefore, system should provide architecture for easy adding new advanced teaching material besides basic that are already implemented.

Assessment module of Protus needs to present various programming tasks and tests to learner and allow evaluation of these assignments within Java programming language course. This module will also provide an authoring tool that will allow instructors to develop new tasks and questions. Desktop application of Protus already allows easy authoring of tasks and multiple-choice questions that are kept in form of html document within Protus. New textual programming assignments needed for assessments with Testovid can be also kept in that form. Therefore, the task is just to enter the already existing textual tasks from Testovid through the user interface of desktop application of Protus.

This new module of Protus system will define infrastructure for the storage of learner programming solutions in Java, submission of those solutions, manual, automated or semi-automated assessment of the solutions, and the storage of learner's results and grades. The integration of this specific assessment tool within Protus should include a sufficient volume of new programming tasks for automated assessment. Significant number of tasks already exists in Testovid system and additional ones can be added with authoring tool of Protus. Vast and quality of database of various tasks will be vital for further experiments with personalized tutoring.

Learners will have access to this automated assessment system and can use it to check their own solutions before final submission. This will provide learners with a faster feedback

and it will eliminate some oversights resulting in accidental submission of bad solutions. System will keep a detailed history of all learners' solutions and their results. This history could be useful for teachers to track progress of learners and for system itself to automatically recommend next tasks to new learners.

A main goal of this software architecture is to improve existing Protus system and equipped it with advanced assessment functionalities of students' programming solutions available through Testovid system. The new integrated system will help learners in the submission of solutions to programming assignments and teachers in handing out assignments and gathering the learners' solutions. Precise grading will be solid base for building learner model and performing efficient personalisation.

Grading system must be precise and efficient. Grading model of Testovid should be altered and adapted to needs of Protus system and it will be similar to one presented in [2]. Marks will be awarded for the various elements of submitted solution, for example some marks will be awarded for successful compiling, for different elements during style checking, and for successful testing of each submitted solution. Grading system should include low marks for complete solutions to relatively easy tasks, and higher marks awarded for complete solutions to progressively more difficult problems. Partial marks should be also awarded for incomplete solutions.

Introducing an automated assessment module in the Protus system will allow the teacher to have insight into problems that learners may encounter during programming and as a consequence to better understand difficulties they face by observing their programs and codes.

V. CONCLUSION

Integrated system presented in this paper is designed to combine personalized tutoring system for Java programming with environment for submission of learners' programming solutions with a main goal to provide a great degree of high-quality automated testing. Protus will provide optimal performance with use of the most appropriate architecture. Efficient tool for testing learner's knowledge and automating assessment of learner programs will be included. All of the actions will be run from the server. That will enable learner to take courses and test his/her progress in learning via Internet.

This improved system allows learners to take on-line course, to test their knowledge and programming skills, to submit their programming solutions and receive automated feedback and appropriate reports, to take on-line exams and be automatically assessed. These educational activities in the integrated system, learner will be able to take in computer labs, at home or over a network or internet.

The main aim of this software architecture is to ease the realization of lectures and exercises on Java programming course. This architecture also allows implementing various additional computer science courses. This integrated educational environment could be successfully used for different programming courses and within them for different

purposes: teaching and learning, repetition and reinforcement, and assessment and evaluation of even more complex programming solutions. The system can also be used in various environments like: in the classroom during lectures, in the computer labs during practical exercises and assessment, at home during individual practicing, or in a network or internet environment for distance learning.

ACKNOWLEDGMENTS

Authors wish to thank Ivan Pribela for providing us with rights for presenting details about Testovid system.

This paper is part of the research project Infrastructure for Technology Enhanced Learning in Serbia supported by the Ministry of Education and Science of the Republic of Serbia [Project No. 47003].

REFERENCES

- [1] N. R. Thotaa, and Whitfielda "Use of CALMS to enrich learning in introductory programming courses", *Proceedings of the 17th International Conference on Computers in Education [CDROM]*. Hong Kong: Asia-Pacific Society for Computers in Education. 2009
- [2] C. Douce, D. Livingstone, and J. Orwell, "Automatic test-based assessment of programming: A review", *ACM Journal on Educational Resources in Computing*, 5, 2005, Article No. 4.
- [3] A. Klačnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, "Integration of recommendations and adaptive hypermedia into Java tutoring system," *Computer Science and Information Systems – ComSIS*, Vol. 8, Num. 1, 2011, pp. 211-224.
- [4] A. Klačnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, Z. "E-learning personalization based on hybrid recommendation strategy and learning style identification," *Computers & Education* 56, 2011, pp. 885-899.
- [5] B. Vesin, M. Ivanović, A. Klačnja-Milićević, and Z. Budimac, "Protus 2.0: ontology-based semantic recommendation in programming tutoring system", *Experts systems with application Vol 39*, 2012, pp. 12229–12246
- [6] I. Pribela, M. Ivanović, and Z. Budimac, "System for testing different kinds of students' programming assignments", *Proceedings of 5th International Conference on Information Technology ICIT 2011*, Amman, Jordan, May 2011, paper no. 535
- [7] I.H. Hsiao, P. Brusilovsky, and S. Sosnovsky, "Web-based parameterized questions for object-oriented programming". In C. Bonk et al. (Eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2008*, pp. 3728-3735.
- [8] M. Joy, N. Griffiths, and R. Boyatt, "The boss online submission and assessment system", *Journal on Educational Resources in Computing (JERIC)*, Vol. 5 Issue 3, 2005, Article No. 2.
- [9] A. N. Kumar, "Generation of problems, answers, grade, and feedback-case study of a fully automated tutor", *Journal on Educational Resources in Computing (JERIC)*, Vol. 5 Issue 3, 2005, Article No. 3,
- [10] E. R. Sykes, and F. Franek, "A prototype for an intelligent tutoring system for students learning to program in Java". *Int. J. Comput. Appl.* 1, 2004, pp. 35-44.
- [11] P. Ihanntola, T. Ahoniemi, V. Karavirta, and O. Seppälä, "Review of recent systems for automatic assessment of programming assignments", *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, 2010, pp. 86-93
- [12] I. Pribela, M. Ivanović, and Z. Budimac, "Testing almost any aspect of students' assignments". 3rd Balkan Conference in Informatics, Sofia, Bulgaria, 2007, pp. 173-182.