

ASSIGNMENT WORKFLOW SUPPORT SYSTEM

Slavomír ŠIMONÁK

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice,
Letná 9, 042 00 Košice, Slovak Republic, tel.: +421 55 602 3021, e-mail: slavomir.simonak@tuke.sk

ABSTRACT

The paper address the issues connected with increasing the effectiveness of some aspects of teaching process. With the risen number of students in bachelor study programs as well as requirements of non-pedagogical nature in last few years, questions related to the support of knowledge assessment process are those of increasing importance. While in past years a personal conversation was the essential part of the knowledge assessment process, sometimes it is hard to use nowadays in such an extent. A system for support the students' assignments workflow was developed and is employed at author's home institution. Descriptions of system features as well as practical experiences are presented within the paper.

Keywords: knowledge assessment, LMS, programming teaching, assignment workflow

1. INTRODUCTION

The goal of the assessment process is to measure the level of knowledge reached by the students. While in past years a personal conversation was the essential part of the knowledge assessment process, sometimes it is hard to use nowadays in such an extent for the reasons mentioned above. In some technical fields, like programming, practical part of study is of very high importance. A lot of teacher's effort is necessary connected with this part of teaching process, like creating suitable assignments, distributing them to students, consulting different questions arising during their work, checking the validity and originality of proposed solutions, assessing the student's work and archiving results to name at least some of them. We will use the term *assignment workflow* to cover these activities. Although in general, the problem of automating the whole process of knowledge assessment, mainly when we consider higher levels of cognitive goals [1,2] is out of scope of this paper, we try to address some of the tasks connected with programming assignments, which can be automated, but are not covered by LMSs currently used [3]. The objective of the paper thus is a description of the system developed and employed at the author's home institution,

which supports our work within the subject Assembly Language Programming, currently taught in second semester of the bachelor study program.

2. RELATED WORK

In the last decade, there is a vital research activity connected with the evaluation of students' programming assignments [4,5]. Systems like PROGTEST [4], ASSYST [6], MARMOSSET [7] and WEB-CAT [8] available today have been developed in order to provide the support for submission and evaluation of students' programs. While all the systems evaluate the students' solutions according to various parameters (correctness, efficiency, complexity, style, etc.), all of them are designed to work with programs written in high-level languages like Java or C. In our case, however, assembly language programs are to be evaluated. So the systems mentioned are not directly usable, although they offer many advanced and inspiring features. Another widespread problem in educational community is the plagiarism. Many plagiarism detection methodologies have been proposed and tools developed [9–11] in order to cope with the problem. Simple detection mechanisms thus are implemented into our solution too.

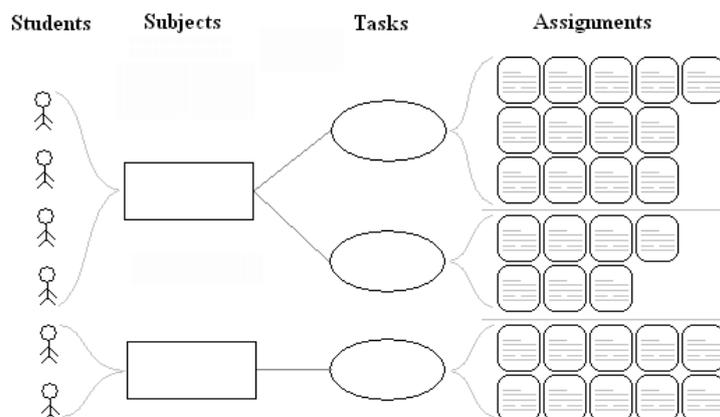


Fig. 1 Subjects, tasks and assignments

3. SYSTEM OVERVIEW

Some notions used further within the paper should be clarified first and the scheme in Fig. 1 can help us. *Subjects* correspond to those of students' curricula, which are supported by the system (Assembly Language Programming in our case). *Tasks* represent the sets of assignments covering the specific part of curriculum of a given subject. In our case, tasks include assignments for different computer architectures (i8080 and x86 respectively). Finally *assignment* is a formulation of specific problem to be solved by students.

Another scheme is given in Fig. 2 to illustrate relations between subjects, tasks and assignments, used here. Sample configuration contains three subjects (V1, V2 and V3 respectively). Within the subject V1, four tasks are defined, with no particular assignments associated yet. Two tasks are defined within the subject V2, both of them with assignments granted. All the assignments of task u1 come from the same set of assignments, which is not the case of task u2. Similarly, task u1 of subject V3 also has assignments selected from different sets (a,b and c).

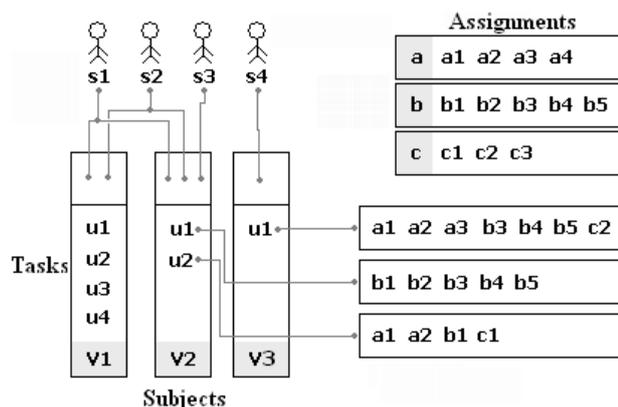


Fig. 2 Sample configuration

Within the system, three user roles are defined, which correspond to three levels of services availability. The roles mentioned above are: administrator, teacher and student. A hierarchy of user roles is defined, with the administrator being positioned at the top level and student at the bottom one. Services of system available to teachers are subset of those available to administrators, and services available to students are subset of services available to teachers (Fig. 3). This allows for very useful feature of the system - every user of the system is able to switch to the interface of user from hierarchically lower level (providing such level exists). It proved to be very useful, when we needed to verify the changes made to the system configuration and see their effect from the other point of view. For example if an administrator working on assigning teachers to subjects wants to verify the correctness of his work, he is able to switch to the interface of some of teachers without knowing the corresponding login and password (Fig. 5). From the security point of view, for this type of relation we use the term *phantom relation* and it is marked with a special flag in log files.

At least privileged (student) level, the system offers services like viewing text of particular assignment, uploading the source code of solution, uploading the documentation (after the successful assignment commitment), viewing the status of assignment submitted, accessing the helpdesk or logout from the system. After uploading the source code, this is compiled by the compiler associated to given type of task and in the case of successful passing the compilation phase, resulting program is executed with the prepared set of inputs. Program outputs are compared with those predefined by administrator and stored in database. In case of successful validity check, the acceptance protocol, otherwise the error message is displayed. The number of submissions is not restricted, and history of attempts is saved.

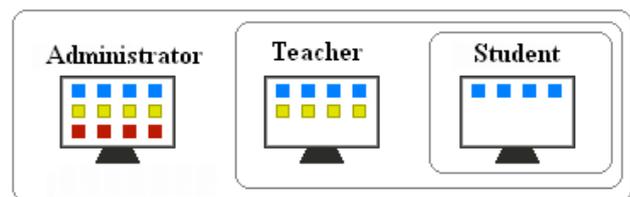


Fig. 3 User roles

Teachers additionally to all the services available to students, have access to details of their subjects, students, tasks and assignments. Tools available to a teacher in phase of assignment assessment include the history of submission attempts (including the files uploaded), lock on given student's assignment in order to make further modifications impossible and originality check, based on comparing the source code with other submitted sources (Fig. 4). Additionally, minimized view of source code (without comments) is available, which can be helpful at the personal conversation investigating the depth of student's understanding of submitted solution.

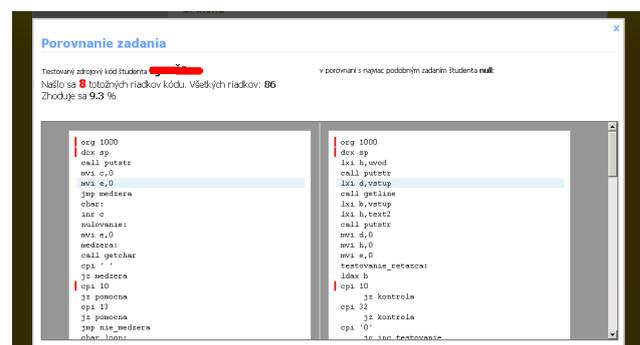


Fig. 4 Originality check

Administrators, additionally to services available to teachers, can configure the system and change the system settings, manage the lists of teachers, students, subjects, tasks and assignments. Additionally they can handle helpdesk, view statistics and perform variety of management tasks, like binding teachers to subjects, distribution

(automatic or manual) of assignments to students, editing the tasks and assignments, etc.

User interface reflects the role of the user currently logged in. For the student, the background color is blue, for teacher it is yellow and for one of administrators, the

color is red. This color-based role identification of currently logged user also works when the phantom relation is established and we believe it can be quite useful in this case (Fig. 5).

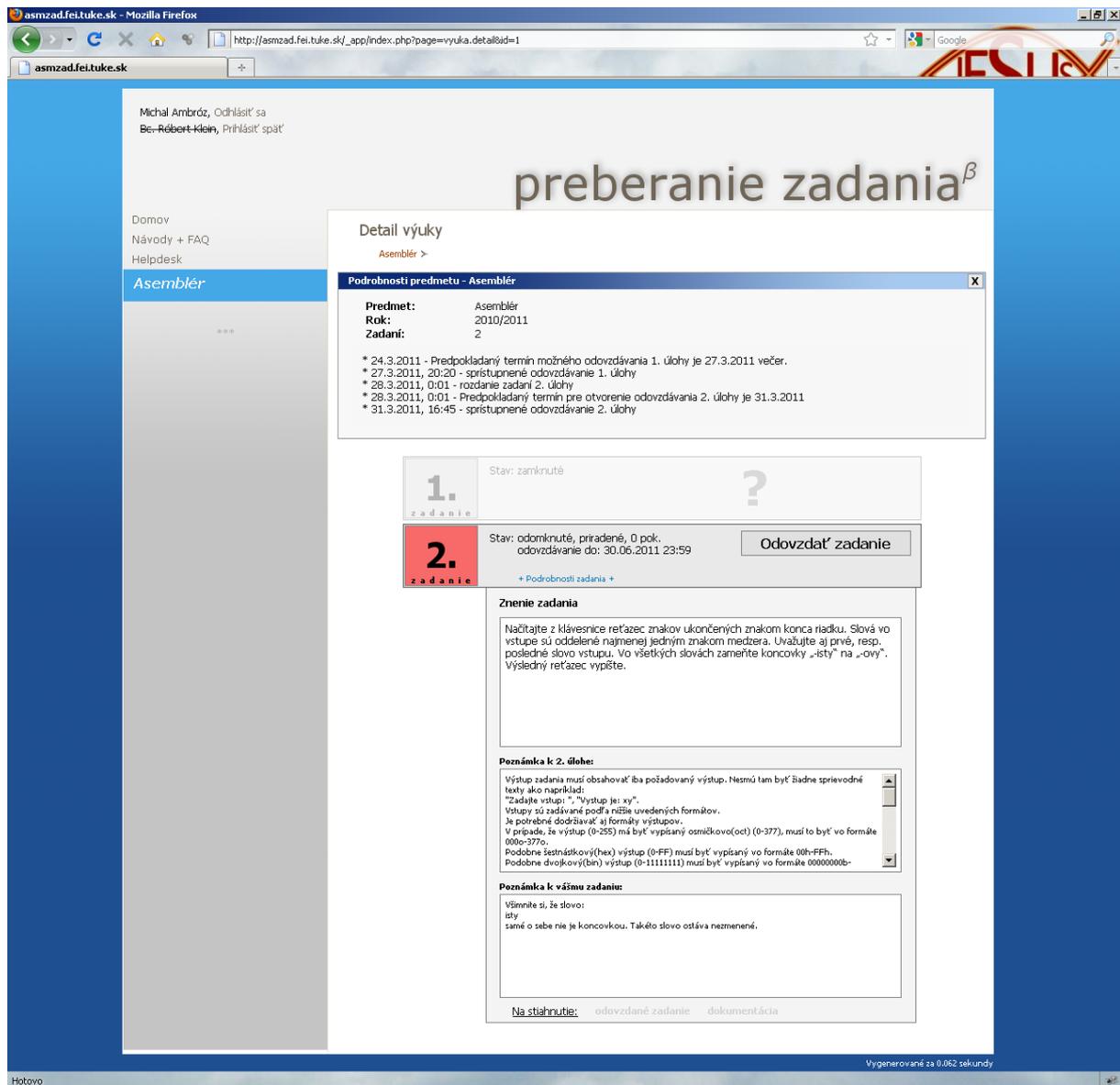


Fig. 5 Student user interface (phantom relation)

4. TECHNOLOGY AND SECURITY

From the technological point of view, the system has got the web-based interface and is built on a set of different open-source technologies, like the Linux operating system, Apache web server, MySQL database and PHP scripting language. For increasing the security of the system, several security-related technologies are employed. Network protocol used is the HTTPS, which is a secure extension of the HTTP protocol, where data transmitted are encrypted using SSL or TLS protocol. Role based access control (RBAC) simplifies the process of granting the access rights - those

are simple assigned to groups of users, not to every user separately. Our system does not manage users' passwords, which saves the work of administrator. Every user must pass the process of authentication using the login and password, he uses to access other information systems within university. After authentication by the LDAP server, the user should be authorized to enter the system and use services provided. But there's one more problem to solve. In that case, all the students and employees having their record in university's LDAP could access the system. For that reason we maintain a list of users of the system, and if the

user pass the LDAP authentication, his presence within the list will be checked. In case of three unsuccessful login attempts within 900 seconds from the single IP address, the address will be blocked for next 900 seconds. When three unsuccessful login attempts are registered from different IP addresses, the login name will be assigned the timeout of 900 seconds, which will be displayed in login dialog. Timeout durations can be adjusted in configuration files. Audit records monitoring the activities of users are generated by the system. Those records can be useful in case of attack. Additionally, SQL requests are protected by the package pear MDB2 [12] against the SQL injection.

As it was mentioned above, two different sets of assignments (tasks) are used within the Assembly Language Programming subject. First of tasks is oriented towards improving practical programming skills on a simple computer architecture, like the Intel 8080 is. Plug-in based emulation platform emuStudio [13,14] is used as a compiler and run-time environment in this case. In case of the second task, more advanced computer architecture (x86) is used and sources submitted are compiled using the NASM assembler [15].

5. PRACTICAL EXPERIENCES

A feature of the system, which proved itself very useful, is the helpdesk (Fig. 6). Its main purpose is to establish a platform for communication between students and the administrator in case of problems with submitting their assignments. The helpdesk topic is created by the student and the conversation with the administrator is private. When the administrator decides, conversation can be helpful for other students too, he can make it public.



Od	Téma	Odp.	Vid.	Typ	Vytv./Poslodp.
Michal Žakarovský	zadanie je zle	1	1	súkr.	30.05.2011 23:56 30.05.2011 23:56, student
Lukáš Murko	dokumentácia	1	1	súkr.	27.05.2011 07:29 27.05.2011 07:29, student
Marek Mikita	zacatie c2	1	1	súkr.	24.05.2011 13:42 24.05.2011 13:42, student
Peter Gierl	Vyskytla sa chyba: Error 4100	4	28	súkr.	08.05.2011 18:23 22.05.2011 09:08, student
Dávid Farkas	Problem s odovzdaním	2	6	súkr.	19.05.2011 18:51 20.05.2011 16:32, student
Lukáš Murko	neprebratie zadania	4	8	súkr.	15.05.2011 12:32 19.05.2011 22:12, student
Vladimír Bindžár	Zadanie	4	9	súkr.	17.05.2011 00:30 17.05.2011 21:29, student
Marcel Gavalier	error error a este error a znova error	6	9	súkr.	16.05.2011 22:51 17.05.2011 13:07, student
Lukáš Franko	problem s nactanim .asm suboru	5	14	súkr.	09.05.2011 00:15 17.05.2011 09:28, student
Peter Janus	chyba?	2	4	súkr.	16.05.2011 20:41 17.05.2011 01:40, admin
Marek Mikita	zadanie	1	5	súkr.	16.05.2011 21:08 16.05.2011 21:08, student
Milan Mruz	Neuspesne odovzdate	2	3	súkr.	16.05.2011 12:52 16.05.2011 16:21, admin

Fig. 6 Helpdesk

Another useful functionality implemented is the automated archiving of all the assignments committed, categorized by subjects and years, together with the documentation in electronic form.

System was developed within the bachelor and diploma theses [16,17] at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice. The size of source code of the application is about 500KB, and it was written using languages like PHP, Java script, HTML and CSS. Database contains 30 tables. Deeper details can be found in [16].

Introduction of strictly defined input/output relations is necessary condition to put a system of this kind to work, but exact rules sometimes are not welcomed by students. Program output, may be correct from semantic point of view is not accepted, unless it is not produced in required format too. Assignments however are attached with a note specifying the problem to solve more precisely, including the information on formatting the output, if it is required (see Fig. 5 for example). Most of problems reported by the students on helpdesk were caused by violating of output formatting rules.



preberanie zadania^β

Povolené súbory na odosielanie

Systém > Povolené súbory na odosielanie >

Typ zadanií *asm*:

zadanie :

<input type="checkbox"/> 7z	<input checked="" type="checkbox"/> asm	<input type="checkbox"/> bmp	<input type="checkbox"/> c
<input type="checkbox"/> cpp	<input type="checkbox"/> doc	<input type="checkbox"/> docx	<input type="checkbox"/> exe
<input type="checkbox"/> jpeg	<input type="checkbox"/> jpg	<input type="checkbox"/> pdf	<input type="checkbox"/> png
<input type="checkbox"/> ppt	<input type="checkbox"/> pptx	<input type="checkbox"/> rar	<input type="checkbox"/> rtf
<input type="checkbox"/> txt	<input type="checkbox"/> xls	<input type="checkbox"/> xlsx	<input type="checkbox"/> zip

dokumentácia :

<input type="checkbox"/> 7z	<input type="checkbox"/> asm	<input type="checkbox"/> bmp	<input type="checkbox"/> c
<input type="checkbox"/> cpp	<input type="checkbox"/> doc	<input type="checkbox"/> docx	<input type="checkbox"/> exe
<input type="checkbox"/> jpeg	<input type="checkbox"/> jpg	<input checked="" type="checkbox"/> pdf	<input type="checkbox"/> png
<input type="checkbox"/> ppt	<input type="checkbox"/> pptx	<input type="checkbox"/> rar	<input type="checkbox"/> rtf
<input type="checkbox"/> txt	<input type="checkbox"/> xls	<input type="checkbox"/> xlsx	<input type="checkbox"/> zip

Fig. 7 Configuration of accepted file types

6. CONCLUSIONS

The system in its current version, although having some limitations, proved his usefulness within the assignment workflow of Assembly Language Programming subject for full-time as well as for distance form of study. In summer semester of 2010/2011, few hundreds of students utilized its services. We appreciate the features it offers, configuration possibilities (e.g. different compilers associated with different tasks, accepted file types (see Fig. 7), etc.), as well as intuitive user interface. It helps us to utilize dedicated time better, doing personal consultations and teaching, rather than managing assignments. If we consider e-learning not only as a platform for creating the content and its distribution to students, but also for knowledge assessment support, systems like one presented within the paper could be essential part of it.

Possible future improvements include exploring the possibilities of integration with LMS system actually used (e.g. Moodle) to support the teaching process. We suggest two types of integration, while the first being based on incorporation the features of the system into LMS and used

for submitting and checking assignments in current form, the second type could be based on cooperation within the testing subsystem of LMS. Small parts of code, for solving simple problems, could be passed to the system, which, after compiling, executing and checking them would sent the result back to the testing subsystem of LMS. Extending the support for other subjects is also a part of our considerations. Originality check provided is quite simple and it could be worth to improve it. N-gram analysis [18,19] seems to be an option in this case.

ACKNOWLEDGEMENT

This work is the result of the project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

REFERENCES

- [1] CLARK, D.: "Bloom's taxonomy of learning domains," 2010. [Online]. Available: <http://www.nwlink.com/~donclark/hrd/bloom.html>
- [2] GENČI, J.: "Advanced knowledge assessment system functional and conceptual specification," in *Technological Development in Education and Automation*. Springer, 2010, pp. 315–319.
- [3] GENČI, J.: "Systém na preberanie zadaní z programovania," in *Sborník príspevku z konferencie a súťaže eLearning 2007, Hradec Králové, 2007*, pp. 257–261.
- [4] de SOUZA, D. – MALDONADO, J. – BARBOSA, E.: "Progtest: An environment for the submission and evaluation of programming assignments based on testing activities," in *Software Engineering Education and Training (CSEET), 2011 24th IEEE-CS*, May 2011, pp. 1–10.
- [5] BARBOSA, E. – SILVA, M. – CORTE, C. – MALDONADO, J.: "Integrated teaching of programming foundations and software testing," in *Frontiers in Education Conference, 2008. FIE 2008.*, Oct. 2008, pp. 5–10.
- [6] JACKSON, D. – USHER, M.: "Grading student programs using assyst," in *Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education*, ser. SIGCSE '97. New York, NY, USA: ACM, 1997, pp. 335–339. [Online]. Available: <http://doi.acm.org/10.1145/268084.268210>
- [7] SPACCO, J. – PUGH, W. – AYEWAH, N. – HOVEMEYER, D.: "The marmoset project: an automated snapshot, submission, and testing system," in *OOPSLA Companion '06*, 2006, pp. 669–670.
- [8] EDWARDS, S. H. – PEREZ-QUINONES, M. A.: "Web-cat: automatically grading programming assignments," in *Proceedings of the 13th annual conference on Innovation and technology in computer science education*, ser. ITiCSE '08. New York, NY, USA: ACM, 2008, pp. 328–328. [Online]. Available: <http://doi.acm.org/10.1145/1384271.1384371>
- [9] YANG, S. – SHU-CHEN, L. – CHEN-GENG, T. – MING, C.: "Research on anti-plagiarism system and the law of plagiarism," in *Education Technology and Computer Science, 2009. ETCS '09.*, Vol. 2, March 2009, pp. 296–300.
- [10] XIONG, H. – YAN, H. – LI, Z. – LI, H.: "Buaa-antiplagiarism: A system to detect plagiarism for c source code," in *Computational Intelligence and Software Engineering. CiSE 2009*, Dec. 2009, pp. 1–5.
- [11] PRECHELT, L. – MALPOHL, G. – PHILIPPSEN, M.: "Finding plagiarisms among a set of programs with jplag," *Journal of Universal Computer Science*, Vol. 8, No. 11, pp. 1016–1038, Nov. 2002.
- [12] CONVISSOR, D. – JANSEN, M. – MERZ, A.: "Pear manual," 2011. [Online]. Available: <http://pear.php.net/manual/en/>
- [13] ŠIMOŇÁK, S. – JAKUBČO, P.: "Software Based CPU Emulation," *Acta Electrotechnica et Informatica*, Vol. 8, No. 4, pp. 50–59, 2008. [Online]. Available: <http://www.aei.tuke.sk/>
- [14] JAKUBČO, P. – ŠIMOŇÁK, S. – ÁDÁM, N.: "Communication Model of emuStudio Emulation Platform," *Acta universitatis sapientiae: Informatica*, Vol. 2, No. 2, pp. 117–134, 2010. [Online]. Available: <http://www.acta.sapientia.ro/acta-info/informatica-main.htm>
- [15] NASM *The Netwide Assembler*, 2011. [Online]. Available: <http://www.nasm.us/xdoc/2.09.10/nasmdoc.pdf>
- [16] KLEIN, R.: "Support system for student programming assignments," Master's thesis, Faculty of Electrical Engineering and Informatics, Technical University of Košice, 2011.
- [17] KLEIN, R.: "A support system for student programming assignments," Master's thesis, Faculty of Electrical Engineering and Informatics, Technical University of Košice, 2009.
- [18] CAVNAR, W. B. – TRENKLE, J. M.: "N-gram-based text categorization," in *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161–175.
- [19] KESELJ, V.: "Text::ngrams - flexible ngram analysis," 2004. [Online]. Available: <http://www.ai.mit.edu/courses/6.863/Ngrams.html>

Received October 18, 2011, accepted March 5, 2012

BIOGRAPHY

Slavomír Šimoňák was born on 23.9.1974. In 1998 he graduated from the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University of Košice and defended his PhD thesis titled 'Formal Method Integration Based on Transformation of Petri Nets and Process Algebras' in 2003. Currently he is working as an assistant professor with the Department of Computers and Informatics. His research interests include formal methods integration and application, machine-oriented languages and computer emulation.