Guest Editorial

Special Issue on Computer Engineering Education

Computer engineers develop, design, test and maintain computer hardware, software and computer systems and may specialise in one aspect of computer engineering such as hardware or software engineering. New pedagogical approaches and application of intelligent technologies in the area of computer engineering education (CEE) have become more and more popular. CEE systems and environments allow students' access to the online resources provided by academic staff and discussions is possible with instructors in any location, at any time. Students not only download materials, but share relevant experiences and information with other classmates.

This special issue is focusing on developing new directions on e-education, teaching and learning, informatics engineering in school education and higher education in an increasingly globalised world. The goal of the special issue is to bring together researchers involved in teaching computer engineering or using digital technologies in computer engineering teaching and learning. In this special issue, the authors explore the potential of new pedagogical approaches and technologies in CEE. Additionally, the special issue analyses the state of the art in theoretical foundations and technological applications of these areas of research in the context of CEE.

The special issue's topics of interest include application of intelligent technologies in CEE, approaches for creative learning in computer engineering, CEE systems and environments, computer programming education, knowledge sharing and knowledge management in CEE, mobile learning approaches in CEE, promoting and supporting CEE at schools, software engineering education issues, and students satisfaction and motivation with CEE.

This special issue included eight papers selected from two traditional international conferences in the area: IFIP WC3 Woking Conference ''A New Culture of Learning: Computing and Next Generations'' (1–3 July, 2015), and ITiCSE 2015, the 20th Annual Conference on Innovation and Technology in Computer Science Education sponsored by ACM (6–8 July, 2015). Both conferences have been organised by Vilnius University Institute of Mathematics and Informatics, and have been hold in the Seimas Palace in Vilnius, the site of the Lithuanian Parliament.

Toll et al. have analysed fine-grained recording of student programming sessions to improve teaching and time estimations. The authors consider that it is not possible to observe directly how students work in an online programming course. This makes it harder for teachers to help struggling students. By using an online programming environment, one has the opportunity to record what the students actually do to solve an assignment. These recordings can be analysed to provide teachers with valuable information. The authors developed such an online programming tool with fine-grained event logging and used it to observe how their students solve problems. The tool provides descriptive statistics and accurate replays of a student's programming sessions, including mouse movements. In this paper, the authors compare fine-grained logging to existing coarse-grained logging solutions to estimate assignment-solving time. It was found that time aggregations are improved by including time for active reading and navigation, both enabled by the increased granularity. The authors have also divided the time users spent into editing (on average 14.8%), active use (on average 37.8%), passive use (on average 29.0%), and estimate time used for breaks (on average 18.2%). It was found that there is correlation between assignments' solving time for students who pass assignments early and students that pass later but also a case where the times differ significantly. The tool can help to improve computer engineering education by providing insights into how students solve programming assignments and thus enable teachers to target their teaching and/or improve instructions and assignments.

Jasute et al. have explored personalised learning methods and activities for CEE. Their paper aims at establishing interconnections between Felder-Silverman learning styles model (FSLSM) and inquiry-based learning (IBL) activities. FSLSM is known as the most suitable for engineering education and e-learning. IBL is known as very helpful for students while studying STEM (Science, Technology, Engineering and Mathematics) subjects incl. Computer Engineering. Interconnections are established using expert evaluation method based on trapezoidal Fuzzy numbers. The established interconnections are useful while creating suitable IBL-based learning scenarios for students having different learning styles. These learning scenarios could be created by using ontologies-based recommender systems for CEE and STEM subjects using created interconnections.

Juskeviciene et al. have studied application of 1:1 mobile learning scenarios in CEE. The paper aims at

presenting case study on personalised mobile learning scenarios on Computer Engineering and other STEM subjects in Lithuania. The authors have used VARK learning styles model to personalise CCL project's mobile learning scenarios by establishing suitable learning components for particular students according to their personal needs. Personalised mobile learning scenarios have been compared against traditional "one size fits all" learning scenarios. Comparison was performed on the base of the analysis of Lithuanian CCL teachers' answers on questionnaire during the 2nd cycle of CCL project implementation and previous research results on expert evaluation of these types of learning scenarios. During this cycle, IBL activities and a number of mobile apps were used by Lithuanian teachers while implementing the mobile learning scenarios, and interconnections between personalised IBL sub-activities and mobile apps were established. The authors have also analysed CCL observation visits' final report in terms of learning personalisation, creativity and innovation in schools. Based on this three-fold research, the authors concluded that pedagogical change is necessary to improve learning outcomes for students, and the main success factors in implementing mobile learning scenarios in Lithuania were identification of students' learning styles; identification and application of suitable learning activities, methods, learning objects, tools and mobile apps according to students' learning styles; and use of proper sets and sequences of learning methods while implementing mobile learning scenarios.

Velazquez-Iturbide has explored two tools (GreedEx and OptimEx) to experiment with optimisation algorithms. The tools have some common features but differ in their degree of generality and scaffolding. GreedEx is a tool for novices aimed at the active learning of the foundations of greedy algorithms. It currently supports six optimisation problems. OptimEx is a more advanced general experimentation tool that can be used with any kind of optimisation algorithms. If both systems are used in an algorithm course, they should be used at different stages. The paper presents two contributions. Firstly, it presents the novel system OptimEx. Secondly, it gives recommendations of use for both tools, based on the author's experience using and evaluating them. Of particular interest is a list of incorrect outcomes that may be produced by OptimEx which are symptoms of students' misconceptions.

Dolgopolovas et al. explore computational thinking (CT) of software engineering novice students based on solving computer science tasks. The authors argue that during the recent years CT has been actively promoted through the K-12 curriculum, higher education, contests and many other initiatives. CT skills are important for further students' educational and professional career. The paper focuses on CT for software engineering novice students, a term meant to encompass a set of concepts and thought processes that are helpful in formulating problems and their solutions. Annually organised international challenge on informatics and CT named "Bebras" has developed many tasks to promote deep thinking skills in this area. It is important to motivate students to solve various informatics or computer science tasks and evaluate their CT abilities. The paper presents a study conducted among first-year students of software engineering studying the structured programming course. As an instrument to measure CT, a test of internationally approved and well-preselected tasks from the "Bebras" challenge has been suggested and validated. The correlation between the students' test results and the structured programming course results has been investigated. The authors conclude with a discussion and future directions to enhance CT skills of novice software engineering students.

Matsuzawa et al. analyse visualisation of compile error correction history for self-assessment in programming education. The authors have developed Compile error Collection Viewer (CocoViewer) for learners in programming to enable them to conduct self-assessment for compiling error records. CocoViewer generates charts that show a trajectory of reducing the correction time of the compile error that is calculated by logs recorded in students' computers during a programming course. Students can see lists of charts for many kinds of compile errors, as well as a particular detailed circumstance of error that is selected by a student. The authors hypothesised that the system promotes clear understanding regarding their compile error learning, which leads to encourage more experiences of compilation error correction, as well as to reduce unarticulated anxiety for the compile error. Research results showed that the students appreciated the system for a reflection of their process, and it was succeeded in reducing unarticulated anxiety for students. The results indicate that self-assessment with CocoViewer enables to boost students' motivation in programming education, which forms the basis of CEE.

Kubilinskiene et al. have performed empirical study on robotics application in Lithuanian schools. The aim of the paper is two-fold: fist, to perform systematic review of the literature on application of educational robots in schools in order to identify the experience in use of robotics in primary, basic and secondary schools, and, second, to conduct empirical study in Lithuania on the attitude towards use of robotic technologies in education, the related experience and demand, and identify the causes of low use of robotics in teaching and learning. Systematic literature review has shown that robotics has been paving its way as a teaching aid in a more intensive and flexible manner. The findings of empirical study have demonstrated the potential in use of robotic technologies and current related implications in Lithuanian schools.

Yagunova et al. present the case study of international online competition ''Beaver'' and evaluate difficulty and complexity of its tasks. The study focuses of the assessment of difficulty and complexity of tasks for schoolchildren. Based on the analysis of results of 6588 participants in the international informatics competition 'Beaver-2012', it was shown that a priori evaluation made by the organisers of the competition does not correspond to its real difficulty for the participant. It was distinguished a cluster of tasks the difficulty of which was underestimated. The correlation between the length of the statement and difficulty for primary school children was shown. In order to make the results of the test more valid, a way of dividing the tasks according to their difficulty and complexity was found. Based on the method, recommendations for the organisers of the tests for general public were formulated in order to make the measures of educational outcomes of computer engineering knowledge more valid and accurate.

Eugenijus Kurilovas
Vilnius University Institute of Mathematics and Informatics,
Akademijos str. 4, 08663 Vilnius, Lithuania.
E-mail: jevgenij.kurilov@mii.vu.lt