

Contents

Special Issue

Trends in Software Engineering for Engineering Education

Guest Editors

Miguel Ángel Conde González—Escuelas de Ingenierías Industrial e Informática, Universidad de León, Spain

Antonio Sarasa Cabezuelo—Facultad de Informática, Universidad Complutense de Madrid, Spain

- | | | |
|----------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------|
| Miguel Ángel Conde-González and Antonio Sarasa-Cabezuelo | 869–872 | Guest Editorial |
| Carlos Garcia, Fernando Castro, Jose I. Gomez, Daniel Chaver and Jose A. Lopez-Orozco | 873–885 | OpenIRS-UCM: an Integral Solution for Interactive Response Systems |

During the last years many different tools have appeared devoted to facilitate the communication between teachers and students in a classroom. Most of these solutions, commonly referred to as Interactive Response Systems, are closed, too rigid, and in many cases too expensive for educational institutions. In this work we present an extended and improved version of the OpenIRS-UCM application, a free, easy to use and open source tool which supports diverse clickers, can be run in different platforms (Windows, Linux and MacOS) and is totally integrated with the well-known Moodle Learning Management System. We evaluate the OpenIRS-UCM application in terms of various software quality metrics (program complexity and usability), the level of acceptance currently achieved and its potential adoption in real life settings. Also, we perform an exhaustive comparison of the features that our application provides to those of various commercial tools, which reveals that OpenIRS-UCM exhibits many advantages over other major IRS players. Finally, we detail how our tool represents a remarkable advance in the research areas of engineering learning systems and engineering assessment.

Keywords: cooperative/collaborative learning; distance education and telelearning; improving classroom teaching; interactive learning environments

- | | | |
|----------------------------------------------------|---------|---------------------------------------------------|
| Cristina Manresa-Yee and Esperança Amengual | 886–893 | Tailoring ISO/IEC 12207 for Usability Engineering |
|----------------------------------------------------|---------|---------------------------------------------------|

Usability is a key aspect in diverse courses in Computer Engineering Education. Although usability has been analysed and discussed in both Software Engineering and Human-Computer Interaction disciplines, there is still a need to provide a global view of usability between these two perspectives that have evolved separately. Aiming at emphasizing on usability engineering, we have compiled the usability practices and techniques reported in the literature and present a mapping between these practices and the software life cycle processes defined in ISO/IEC 12207:2008 *usability process view*. The reported results can help lecturers to plan and coordinate their usability-related courses and give the students a complete, consistent and broad picture about the integration of usability practices and techniques into software development processes.

Keywords: usability; usability engineering; software engineering; human computer interaction; ISO/IEC 12207; engineering education

- | | | |
|-------------------------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------------------------------|
| Ana González-Marcos, Fernando Alba-Elías, Joaquín Ordieres-Mere, Javier Alfonso-Cendón and Manuel Castejón-Limas | 894–904 | Learning Project Management Skills in Engineering through a Transversal Coordination Model |
|-------------------------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------------------------------|

This paper presents an innovative project management teaching approach to complement technical concepts with competences in the area of human skills. In the proposed model, the authors decided to develop a project-based and collaborative learning model combined with a transversal coordination between the same subjects in different engineering degree programs, considering the different curricula of each degree involved. This approach, among other things, allows the student to develop group competences like teamwork and communication, and to implement, in scale, the professional skills that should be deployed by a project manager, like leadership, negotiation and team management.

Keywords: engineering education; project management; technical skills; human skills; transversal coordination model

- | | | |
|--------------------------------------------------------------------|---------|-------------------------------------------------------------------------------------------------|
| Olaf Radant, Ricardo Colomo Palacios and Vladimir Stantchev | 905–914 | Assessment of Continuing Educational Measures in Software Engineering: A View from the Industry |
|--------------------------------------------------------------------|---------|-------------------------------------------------------------------------------------------------|

In a field like Software Engineering, highly dependent on knowledge, continuing education is key to warrantee business sustainability. In this scenario, relying on continuing educational measures is crucial to support organizations in their competence management programs. In order to validate a set of measures, in this paper an interview study regarding continuing education measures is reported. The study was carried out in late 2014, early 2015 with some of the most experienced SE professionals from the consulting company BearingPoint. Study findings suggest an increasing relevance of such educational measures. Furthermore, authors provide insights about relevant educational measures, improvement areas in employees' training, commonly experienced problems, as well as instruments for control and evaluation.

Keywords: software engineering; educational measures; continuing educational; training

- | | | |
|------------------------|---------|----------------------------------------------------------------------------------------------------|
| K. Indra Gandhi | 915–926 | A Teaching-Learning Model for Software Engineering Courses through Sensor-Based Cognitive Approach |
|------------------------|---------|----------------------------------------------------------------------------------------------------|

There is a big revolution and discussion regarding the education system of the students pursuing engineering education in highly populated countries. Educating Engineers is a highly motivating as well as a challenging task. However, the students from diverse environment perceive the lecturing of the Teachers at different levels. This paper discusses the importance of student's preference in initiating a cognitive based learning environment for engineering education. This includes the evolution in reversing the teaching

mode for industrial perspective courses such as the “System Analysis and Design” course which is pursued by students across various engineering disciplines. Similarly, Software Engineering relates to analyzing and building software systems. Object-Oriented Analysis & Design focuses on the analysis and design of software systems as related objects. A Coordinated Cognitive Thinking (CCT) model to evaluate the cognitive behavior of the students has been proposed so as to analyze the requirements of the students towards Software Engineering related courses. The need for cognitive based learning is assessed through Need For Cognition (NFC) scale with the focus on software system analysis and development. The response is evaluated through wearable sensor technology. Experimental results of around 70 students pursuing the Object-Oriented Analysis course have been taken into consideration. The results obtained revealed the fact that there should be a renaissance in engineering education by continuously reviving the teaching methodology for engineering students.

Keywords: cognitive thinking; teaching-learning process; wearable sensors; object-oriented analysis & design; software engineering; system analysis and design

Tom Adawi, Håkan Burden, Dennis Olsson 927–936 Characterizing Software Engineering Students’ Discussions during Peer and **Rickard Mattiasson** Instruction: Opportunities for Learning and Implications for Teaching

Peer instruction is a method for activating students during lectures, which has gained a considerable amount of attention in higher education due to claims of dramatic improvement in learning gains. The purpose of this qualitative research study is to investigate what types of discussions engineering students engage in during a peer instruction session and what learning possibilities that are enabled by these different types of discussions. We observed twelve students during six separate and simulated peer instruction sessions and the students were interviewed individually after the sessions. An analysis of the data revealed that the students engaged in three qualitatively different types of discussions: *affirmative* discussions, *motivating* discussions, and *argumentative* discussions. We characterize these different types of discussions in terms of the number of alternative answers the students discuss, the extent to which they draw on prior knowledge and experiences, as well as the fundamental difference between an explanation and an argument. A good opportunity for learning is opened up when students are aspiring to find the truth, not simply being satisfied with what they believe to be true. We conclude that students do not always engage in discussions that support their learning in the best way, and we discuss implications for using peer instruction as a teaching method.

Keywords: peer instruction; learning possibilities; argumentation; software designs; UML diagrams

Pamela Flores Nelson Medinilla and 937–947 Persistent Ideas in a Software Design Course: A Qualitative Case Study
Sonia Pamplona

This study aims to discover what persistent ideas students have when designing software, and discusses possible relationships between them. The research was conducted through qualitative case study over an academic period with Master’s degree students in a Software Design course. The ideas obtained as results were grouped in persistence levels: low, medium and high; additionally some ideas have been identified, that could be potentially persistent. The main contribution of this paper is focused on two aspects: (a) Software design education, which allows teachers to identify and address problems related to Software Design course; and (b) Professional impact in the industry, by warning the software industry about the main problems that students carry out, despite of the instruction.

Keywords: software design; persistent ideas; qualitative research; computer science education

A. Pozo-Ruz, M. J. Durán, F. J. Sánchez-Pacheco, E. Rivas-Montoya, P. J. Sotorrión-Ruiz and **F. D. Trujillo-Aguilera** 948–955 Multidisciplinary Power Electronics Courses with On-line Simulation Tools

The advent of renewable energies, electrical vehicles and smart grids are expanding the teaching of Power Electronics into areas such as mechanical or energy engineering. In this multidisciplinary context, the adaptation of the existing resources to the new profile of students is becoming a major need. Aiming to fill this gap, the University of Malaga offers specifically designed multidisciplinary course for students who are unfamiliar with Power Electronics. These courses follow a blended top-down approach and use a custom made free-software on-line simulation tool to improve the interactivity and visualization. The top-down project-based approach allows an active role of the student and the ready-to-use easy java simulation applets eases the asynchronous study of Power Electronics converters in a friendly environment. The accuracy of the applets is verified by simulation and experimental results and the didactic benefits of the course are demonstrated by questionnaires and tests.

Keywords: power electronics; simulation tool; multidisciplinary; non-electrical students

Azeddine Chikh and **Jawad Berri** 956–968 A Software Engineering Framework to Assist Instructors in Eliciting Course Requirements

This paper proposes a software engineering framework that aims to assist instructors in expressing their courses’ requirements. The framework allows an instructor to elicit his know-what and know-how knowledge in teaching a course through a set of learning activities that are the building blocks of e-courses. The instructor’s expertise is elicited through a graphical user interface that provides the necessary tools for producing a course description which is then converted automatically into a course specification allowing software engineers and programmers to implement it as an e-course. We anticipate that this framework would be very helpful for instructors to express their course requirements in a systematic and convenient way. In this paper we present the framework and show by means of a case study related to a systems analysis and design course, how this framework is used to produce the e-course specifications from the instructor requirements.

Keywords: software systems requirements engineering; education engineering; elicitation; specification; course requirements; unit of learning; activity; metadata

Luis Miguel Serrano-Cámara, Maximiliano Paredes-Velasco, J. Ángel Velázquez-Iturbide, Carlos-María Alcover and **M^a Eugenia Castellanos** 969–981 MoCAS: A Mobile Collaborative Tool for Learning Scope of Identifiers in Programming Courses

This article presents an instructional framework for collaborative learning, called CIF and aimed at the analysis level of Bloom’s taxonomy, as well as a mobile collaborative tool called MoCAS that supports CIF. MoCAS is aimed at the domain of scope of identifiers in programming learning, which is a topic present in programming courses in engineering studies. The specification and development of MoCAS were explicitly driven by pedagogical goals and by the atomic actions declared in CIF as simple items of collaborative activities. Furthermore, CIF and MoCAS were evaluated in an actual educational context with respect to students’ performance and motivation. Students using CIF and MoCAS obtained statistically significant higher grades than students studying in an individual or collaborative basis but not using MoCAS. In addition, we measured statistically significant measures indicating that students instructed with CIF and MoCAS were more motivated than students instructed collaboratively but not using CIF or MoCAS. In addition to CIF and MoCAS, and the evaluation results, the experiences here reported exemplify several software engineering practices: the design of an educational system based on knowledge of the target domain (namely, Bloom’s taxonomy) and the evaluation of users’ satisfaction (mainly, students’ motivation).

Keywords: CSCL; Bloom’s taxonomy; computer programming; learning performance; motivation

Cuahtémoc López-Martín, Ivica Kalichanin-Balich, Rosa Leonor Ulloa-Cazarez and Noel García-Díaz 982–994 A Radial Basis Function Neural Network for Predicting the Effort of Software Projects Individually Developed in Laboratory Learning Environments

Prediction techniques have been applied for predicting dependent variables related to Higher Education students such as dropout, grades, course selection, and satisfaction. In this research, we propose a prediction technique for predicting the effort of software projects individually developed by graduate students. In accordance with the complexity of a software project, it can be developed among teams, by a team or even at individual level. The teaching and training about development effort prediction of software projects represents a concern in environments related to academy and industry because underprediction causes cost overruns, whereas overprediction often involves missed financial opportunities. Effort prediction techniques of individually developed projects have mainly been based on expert judgment or based on mathematical models. This research proposes the application of a mathematical model termed Radial Basis function Neural Network (RBFNN). The hypothesis to be tested is the following: effort prediction accuracy of a RBFNN is statistically better than that obtained from a Multiple Linear Regression (MLR). The projects were developed by following a disciplined development process in controlled environments. The RBFNN and MLR were trained from a data set of 328 projects developed by 82 students between the years 2005 and 2010, then, the models were tested using a data set of 116 projects developed by 29 students between the years 2011 and first semester of 2012. Results suggest that a RBFNN having as independent variables new and changed code, reused code and programming language experience of students can be used at the 95.0% confidence level for predicting the development effort of individual projects when they have been developed based upon a disciplined process in academic environments.

Keywords: laboratory learning environments; software development effort prediction; radial basis function neural network; multiple linear regression

Yu-Hsin Hung, Ray- I Chang and Chun-Fu Lin 995–1006 Developing Computer Science Learning System with Hybrid Instructional Method

Technology has flourished, and diverse teaching techniques have emerged such as eBooks, massive open online courses, and gamification. Accordingly, technologies continue to provide a revolutionary improvement on traditional learning environments. The pedagogic trend is becoming “student-centered”. Problem-based learning (PBL) is the use of problems to motivate learners to apply research concepts and thinking strategies. Gamification provides the learners to actively learn through gamified mechanisms. This study focuses on PBL and gamification to enhance students’ engagement in learning computer science. We developed a computer science learning system (CSLS) comprising three subjects: database management, programming, and data structure. Each subject has problem-solving mini games for students to accomplish tasks using relevant concepts. Results reveal that learners have a 95% probability of obtaining above-average user satisfaction, which suggests that the CSLS can be a good vehicle for cultivating the relevant computer science concepts. Participants’ technology acceptance and their cognitive load when using the system are also determined in the experiment, with the aim of examining learners’ perception while using the CSLS. Further, gamification has indicated a positive effect on students’ learning performance and cognition for learning computer science.

Keywords: gamification; problem-based learning; programming learning; computer science education

Francisco Jurado and Miguel A. Redondo 1007–1014 IMS-LTI and Web-Services for Integrating Moodle to an Eclipse-Based Distributed Environment for Learning to Program

Learning Management Systems (LMS) offer generic services and tools in order to provide eLearning activities that cover a wide scope of teaching/learning methods. However they lack specific tools to support particular learning activities. Taking into account that there is not one unique tool that covers the entire learning/teaching process, an ideal eLearning scenario must use several specific tools for precise learning tasks. If we want to promote autonomous active learning by means of Personal Learning Environments (PLE) or extrapolate this feature to the new tendency of Massive Open Online Course (MOOC), this scenario is not only desirable but also necessary. Nevertheless, as the eLearning environment scales with the integration of more tools, we have to face communication and architectural issues. This paper exposes the solution we implemented by means of a centralized access point constituted by an LMS, an architecture based on Tuple Spaces, and the use of eLearning IMS-LTI specification to allow communication and information exchange among the different services and components of a system whose aim is to help students develop programming skills.

Keywords: IMS-LTI; moodle; eclipse; blackboard architecture

Mario Manso Vázquez, Manuel Caeiro Rodríguez and Martín Llamas Nistal 1015–1023 Are Learning Software Systems Well-Prepared to Support Self-Regulated Learning Strategies?

There are several self-regulated learning strategies that can be followed by students in order to attain their learning goals. They involve different issues, from motivation to cognitive and metacognitive beliefs and procedures. In conjunction, they involve learners taking full awareness and responsibility of their own learning. Nowadays, many software tools are used by teachers and by students to carry out different learning activities. In this research piece we wonder if these tools are well prepared to support the development of self-regulated learning strategies by students. In order to know to what extent these software tools offer this support we defined several criteria for designing software tools with proper support for these self-regulated learning processes, and mainly for the implementation of self-regulated learning strategies. From the criteria we created a questionnaire to evaluate the level of support offered by a software tool for these strategies. This questionnaire allows the evaluation of software tools in this respect, enabling its selection and providing useful information for software developers so they can improve their tools to offer better support. Here we present an example of some results obtained from the first refinement stage of this questionnaire, and also the calibration process of the scores to get coherent results.

Keywords: self-regulated learning; learning strategy; software tool evaluation; questionnaire

María Luisa Sein-Echaluze, Angel Fidalgo-Blanco and Francisco J. García-Peñalvo 1024–1035 Students’ Knowledge Sharing to Improve Learning in Academic Engineering Courses

This paper presents an example of scaffolding during the development of an engineering course, in which students are supported by teachers and other students. This proposal covers the benefits of the use of shared knowledge repositories in which content was created by students. Teamwork is the transversal competence that is considered to be the central knowledge topic. The cooperation among students through teamwork methodology has generated more than 500 learning resources and a knowledge management system, BRACO, which has been created with these resources to manage information and conduct searches according to each student’s profile and needs. The generated knowledge spiral is composed of knowledge circles that increase during each iteration of the action-research implementation. The reflection phase of this research consists of the evaluation of the impact on learning for students in the experimental group after using the knowledge resources generated by students in relation with teamwork competence, in contrast with the control group that does not experience this intervention. With regard to the assessments, several surveys and a learning analytics system, this paper explains the underlying methodological foundations and the empirical study. In comparison to the control group, the experimental group obtained better results in relation to indicators of positive learning results, such as student-student interaction, teamwork development and final grades during the teamwork process.

Keywords: knowledge spirals; knowledge sharing; teamwork competence; cooperative learning; learning content management system; repository

Camino Fernández, Gonzalo Esteban, Francisco J. Rodríguez-Lera, Francisco Rodríguez-Sedano and David Díez 1036–1052 Design Patterns Combination for Agile Development of Teaching/Learning Haptic Simulators

Software engineering courses usually face a situation where a reasonable size application has to be developed within a semester period. But, on the one hand, if students begin from scratch and start with the knowledge acquisition and requirements phases, they usually do not get to the deployment one. And, on the other hand, developing quality software should go on being the main focus of a software engineering course. Our proposal is the development of a haptic simulator as a teaching/learning tool for this purpose using the SHULE framework and following a Scrum development methodology. The core of this framework includes a combination of design patterns that also guide the development of the whole simulator. This approach has been used in the development of a cataract surgery simulator as a teaching/learning tool, and the experience is shown as an example of the general Scrum development that is presented.

Keywords: design patterns; agile development; teaching/learning environment; haptic simulator

Félix J. Pascual-Miguel, Emiliano Acquila-Natale, Ángel Hernández-García and Miguel A. Conde 1053–1062 Design and Implementation of a Business Simulation Game Tool for Services and Digital Economy Courses in Engineering Degrees

Engineering education is increasingly introducing business administration and management courses in order to adapt engineering curricula to the needs of the labor market and 21st century competencies. Traditionally, business simulation games have proved their efficacy in enhancing learning of business-related subjects in graduate and post-graduate courses. However, business simulation games focus on training on operation of traditional markets, characterized by a strong emphasis on manufacturing, promotion and distribution of physical goods, and thereby they might not be suitable to apply concepts specific to the digital and services economy. This research covers this gap by presenting a new business game simulation tool adapted to this new economy. The paper outlines the design and implementation of such a tool, and it proposes an instructional plan that uses the tool in order to fulfill the new training needs of engineers in business and management courses.

Keywords: business simulation games; engineering teaching; business administration; digital economy; SCRUM