

Special issue on realizing artificial intelligence synergies in software engineering

Tim Menzies · Marjan Mernik

Published online: 5 February 2014
© Springer Science+Business Media New York 2014

This special issue focuses on issues arising from the RAISE'12 Workshop on Realizing Artificial Intelligence Synergies in Software Engineering. Our objective is to provide a forum for researchers and industrial practitioners to exchange and discuss the latest innovative synergistic AI and SE techniques/practices.

Why explore this combination of SE and AI? We think there are many answers to that question. As SE is asked to answer dynamic automated, adaptive, and/or large-scale demands, other computer science disciplines come to play. AI is one of them that may bring SE to further heights. Conversely, SE can also play a role to alleviate development costs and the development effort associated with AI tools. Such mutually beneficial characteristics have appeared in the past few decades and still evolve due to new challenges. That is, this special issue explores not only the application of AI techniques to software engineering problems but also the application of software engineering techniques to AI problems.

This special issue is the result of much work that is still ongoing. As to future work, the RAISE series is also on-going. At the time of this writing, the RAISE'13 workshop has just completed (that event was sponsored by the United States National Science Foundation, and we thank them for their generous support). At that meeting, we saw much continued interest in this union of AI and SE. The reader should expect much novel and exciting work from this combination of ideas, in the very near future.

And regarding the work seen to date, we hope that all the aforementioned papers will provide readers with some glimpse of the kind of work discussed at RAISE'12. Also, we would like to sincerely thank the reviewers for their assistance in the reviewing process.

T. Menzies (✉)

Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown,
WV 26506-610, USA
e-mail: tim@menzies.us

M. Mernik

Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17,
2000 Maribor, Slovenia
e-mail: marjan.mernik@uni-mb.si

Finally, we include below some notes on the papers accepted to this special issue. Please enjoy.

Wasif Afzal, Richard Torkar, Robert Feldt, Tony Gorschek. *Prediction of faults-slipping through in large software projects: an empirical evaluation*. This paper presents an extensive empirical evaluation of a number of techniques for predicting the number of faults slipping through to unit, function, integration, and system test phases of a large industrial project. Statistical techniques (multiple regression, pace regression), machine learning techniques (support vector machines, back-propagation artificial neural networks), tree-structured techniques, nearest neighbor techniques, ensemble techniques, and search-based techniques (genetic programming, gene expression programming, artificial immune recognition system, and particle swarm optimization-based artificial neural network) have been contrasted to the human predictions. It is turned out that a combination of human and a search-based technique has the potential to provide improved prediction results.

Tuong Huan Nguyen, Bao Quoc Vo, Markus Lumpe, John Grundy. *KBRE: A Framework for knowledge-based Requirements Engineering*. This paper discusses an important topic: detecting inconsistencies during Requirements Engineering (RE). It explores a somewhat under-explored issue in this field: the use of domain knowledge and semantics of requirements. The paper presents a knowledge-based RE framework (KBRE) that handles domain knowledge and semantics of the requirements. The authors show how this knowledge facilitates the identification of requirements inconsistencies and other related problems. The approach is well grounded in formal theory and uses description logics in the form of Manchester OWL Syntax. They describe a tool they have built, using this method, and its application to an industrial use case.

Mathias Landhäuser, Sven J. Körner, Walter F. Tichy. *From Requirements to UML Models and Back: How Automatic Processing of Text Can Support Requirements Engineering*. This paper presents a Requirements Engineering Feedback System (REFS), which automates the process of keeping requirement specifications, written in a natural language, and models consistent. At first, UML models are created from the textual specifications, and the UML representation is tightly linked with the underlying text. To reflect model changes to the specification, the initial model and the changed model need to be compared and matched. Identified differences are then integrated into the original specification. If requirement specifications changed after the modeling and development of software have already started, the bidirectional connections of text and model are able to detect the results of textual changes and making appropriate updates in the corresponding model. A novel approach has been presented on several case studies.