Preface for the special issue of the 10th international symposium on theoretical aspects of software engineering (TASE 2016)

Bonsangue, M.M.; Deng, Y.X.

Science of Computer Programming 162 (2018) 1–2

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Preface for the special issue of the 10th International Symposium on Theoretical Aspects of Software Engineering (TASE 2016)

The 10th International Symposium on Theoretical Aspects of Software Engineering, TASE 2016, was held in Shanghai during July 17–19, 2016.

The symposium brings together researchers and developers from academia and industry with interest in the theoretical aspects of software engineering. Since modern software systems are becoming larger and more complex, it is challenging to make them trustworthy and software engineering methodologies need to be enhanced using latest results from theoretical computer science. Theoretical computer science has been most effective on developing abstract models of computation and associated analysis methods, and studying the limitation of such models. But more emphasis is necessary for models and techniques supporting modern software engineering concepts. On the other hand, software engineering methods require complex software solutions that are only possible by applying specification, modeling and validation techniques developed in theoretical computer science.

The authors of the most prominent papers presented at TASE 2016 were invited to submit an extended version to this special issue. In order to guarantee the quality of the selected papers, each submission was reviewed by at least two referees. The review process has ensured that the seven accepted articles significantly extend the original conference versions and meet the journal standard. The articles reflect the scope of the symposium and their topics range from algebraic semantics, model transformations, and conformance testing, to formal verification of memory models, analysis of android applications, cloud applications and resource bugs.

The paper*A New Roadmap for Linking Theories of Programming and its Applications on GCL and CSP* written by *Jifeng He* and *Qin Li* proposes a new roadmap for linking theories of programming. It takes an algebra of programs as its foundation, and generates both denotational and operational representations from the algebraic refinement relation. This new approach is applied to GCL (Guarded Command Language) and CSP (Communicating Sequential Processes) to link their various semantical representations based on their algebraic semantics.

The paper *Agile Validation of Model Transformations Using Compound F-Alloy Specifications* written by *Loïc Gammaitoni, Pierre Kelsen*, and *Qin Ma* formalizes the notion of “hybrid analysis” and extends it to support efficient validation of compound transformations. To enable the effective involvement of domain experts in the validation process, the authors propose a new approach to model transformation validation, called Visualization-Based Validation (VBV). Following VBV, representative instances of a to-be-validated model transformation are automatically generated by hybrid analysis and shown to domain experts for feedback in a visual notation that they are familiar with. The paper prescribes a process to guide the application of VBV to model transformations and illustrates it with a benchmark model transformation.

The paper *Sound Conformance Testing for Cyber-Physical Systems: Theory and Implementation* written by *Hugo Araujo, Gustavo Carvalho, Morteza Mohaqeqi, Mohammad Reza Mousavi*, and *Augusto Sampaio* proposes a conformance testing algorithm for cyber-physical systems. The authors show how the dynamics of system specification and the sampling rate play an essential role in making sound verdicts. They specify and prove error bounds that lead to sound test-suites for a given specification and a given sampling rate. In addition, they use reachability analysis to find such bounds and implement the proposed approach using the CORA toolbox in Matlab.

The paper *Formal Proof of Dynamic Memory Isolation Based on MMU* written by *Narjes Jomaa, David Nowak, Gilles Grimaud*, and *Samuel Hym* demonstrates how a proof assistant such as Coq can be used to model a hardware architecture with a memory management unit, and an abstract model of microkernel supporting preemptive scheduling and memory management. The authors formalize the consistency properties that must be preserved in order for memory isolation to be preserved.

https://doi.org/10.1016/j.scico.2018.03.006

The paper *Lightweight Energy Consumption Analysis and Prediction for Android Applications* written by *Yan Hu, Jiwei Yan, Dong Yan, Qiong Lu*, and *Jun Yan* proposes a lightweight and automatic approach to analyze and predict the energy consumption for Android apps. The authors proceed from two perspectives: the method-level analysis gives developers facts about the energy consumption of the user methods in their apps, while the API-level analysis shows the energy consumption of Android APIs. They construct a statistical model from a set of energy values obtained by Dalvik bytecode based instrumentation and software-based measurement, to predict the energy consumption of method sequences or API sequences.

The paper *Simulation-Based Matching of Cloud Applications* written by *Filippo Bonchi, Antonio Brogi, Andrea Canciani*, and *Jacopo Soldani* presents two types of behaviour-aware matching of applications (exact and plug-in) both based on a notion of simulation. The authors extend the notion of plug-in matching by relaxing the notion of simulation to permit matching an operation with a sequence of operations. They also present a coinductive procedure to compute such relaxed simulation, and formally prove the termination, soundness, and completeness of such procedure.

The paper *State-Taint Analysis for Detecting Resource Bugs* written by *Zhiwu Xu, Cheng Wen*, and *Shengchao Qin* proposes a technique state-taint analysis to detect resource bugs. The authors specify the appropriate usage of resources in terms of resource protocols and then presents a taint-like analysis which employs resource protocols to guide resource bug detection. They also refine the protocols to guide the analysis for energy leak detection. A prototype tool called statedroid is developed and shown to be suitable in practice to detect energy leak patterns.

We would like to thank the authors of the seven articles. We gratefully acknowledge the referees for the high quality of their reviews that helped the authors to improve their contributions in various ways. We would also like to thank the editors of Science of Computer Programming and the support staff at Elsevier, for their support during the whole editorial process.

Marcello Bonsangue *
Leiden University, Netherlands
E-mail address: m.m.bonsangue@liacs.leidenuniv.nl

Yuxin Deng
East China Normal University, China
E-mail address: yxdeng@sei.ecnu.edu.cn

Available online 27 March 2018

* Corresponding author.