Editorial

Alessandra Russo[1], Andy Schürr[2] and Heike Wehrheim[3]

[1]Department of Computing, Imperial College London, 180, Queen's Gate, London SW7 2AZ, UK
[2]Technische Universitt Darmstadt, Darmstadt, Germany
[3]Paderborn University, Paderborn, Germany

1. Introduction

Software systems are increasingly governing every aspect of our modern society, with the potential of influencing both its economical and social progress. High-quality, reliable, secure and trusted software is therefore more crucial then ever. Research in fundamental aspects of software engineering continues to deliver advancements in all these directions among both academics and practitioners. This special issue provides a sample of these contributions. Originated from the 2018 edition of the *International Conference on Fundamental Approaches to Software Engineering* (FASE) conference, this special issue contains revised and extended versions of a few of the best papers that were presented at FASE 2018, held as part of the *European Joint Conference on Theory and Practice of Software* (ETAPS) in Thessaloniki, Greece, in April 2018.

FASE is concerned with the foundations on which software engineering is built. Papers published at FASE contribute to make software engineering a more mature and soundly-based discipline, whose advancements are underpinned by formal foundations, precise definitions and rigours proofs. Topics that are of interest to FASE include:

- Software engineering as an engineering discipline, including its interaction with and impact on society and economics;
- Requirements engineering: capture, consistency, and change management of software requirements;
- Software architectures: description and analysis of the architecture of individual systems or classes of applications;
- Specification, design, and implementation of particular classes of systems: (self-)adaptive, collaborative, embedded, distributed, mobile, pervasive, cyber-physical or service-oriented applications;
- Software quality: (static or run-time) validation and verification of functional and non-functional software properties using theorem proving, model checking, testing, analysis, simulation, refinement methods, metrics or visualization techniques;
- Model-driven development and model transformation: meta-modelling, design and semantics of domain-specific languages, consistency and transformation of models, generative architectures;
- Software processes: support for iterative, agile, and open source development;
- Software evolution: refactoring, reverse and re-engineering, configuration management and architectural change, or aspect-orientation.

FASE 2018 received 77 abstract submissions from 24 different countries and only 19 papers were accepted after a through review process. The peer-reviewer papers collected in this special issue have been invited by the guest editors among 8 top papers presented at FASE'18, based on their relevance to the FAC journal. The remaining

---

Correspondence and offprint requests to: A. Russo, E-mail: a.russo@imperial.ac.uk

four of the 8 top papers were instead submitted and accepted as part an STTT special issue as more relevant to software quality tools.

1. *A verification-driven framework for iterative design of controllers*, by Claudio Menghi, Paola Spoletini, Marsha Chechik and Carlo Ghezzi, proposes a verification-driven framework that supports hierarchical decomposition of software controllers through formal specifications and incremental development and verification of the sub-components allowing prototypic implementations of components to be replaced by more sophisticated one during the development process.

2. *Read atomic transactions with prevention of lost updates: ROLA and its formal analysis*, by Si Liu, Peter Csaba lveczky, Qi Wangm Indranil and Gupta Jos Meseguer, proposes a new distributed transactions protocol that is proved to satisfy read atomicity and prevention of lost updates, and shown through statistical model checking to outperform existing distributed transactions protocols.

3. *Interactive verification of architectural design patterns in FACTum*, by Diego Marmsoler and Habtom Kashay Gidey, shows how architectural design patterns (ADP) can be verified using interactive theorem proving. The approach is instantiated by specifying ADPs over a model for dynamic architectures and verifying them using Isabelle/HOL as theorem prover.

4. *Multiple model synchronization with multiary delta lenses with amendment and K-Putput*, by Zinovy Diskin, Harald Knig and Mark Lawford proposes a algebraic model for multi-directional update propagation in the Ageneral context of non binary model-to-model transformations in order to restore consistency.

Alessandra Russo
Andy Schürr
Heike Wehrheim

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.