Preface to the Special Section Issue on Improving Software Quality through Formal Methods

Yliès Falcone, Leonardo Mariani

To cite this version:
hal-02548907

HAL Id: hal-02548907
https://inria.hal.science/hal-02548907
Submitted on 21 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Preface to the Special Issue on Improving Software Quality through Formal Methods

Yliès Falcone · Leonardo Mariani

Received: date / Accepted: date

Abstract This special issue is dedicated to the presentation of novel results in the scope of formal methods for the verification and testing of software. The papers included in the special issue showcase approaches that successfully combine a rigorous and formal background with the capability to address real software applications in realistic settings.

1 Introducing the Special Issue

The special issue is devoted to the use of formal methods for improving the quality of software systems. The included articles focus on the use of runtime techniques, such as runtime verification and trace analysis; see Falcone et al. [2013]; Bartocci and Falcone [2018]; Bartocci et al. [2019]. Following an open call, this issue results in 3 papers that were selected out of 8 papers after a careful reviewing process. The papers are briefly discussed as follows.

In "Verifying Temporal Specifications of Java Programs", Spegni et al. combine satisfiability modulo theory and static analysis to approximate the behavior of Java threads. The behavior approximation is represented as a network of timed automata and the authors show that their abstraction preserves the truthfulness of two real-time temporal logics.
In "Runtime Verification of Real-Time Event Streams Under Non-synchronized Arrival", Scheffel et al. present an approach to monitor concurrent systems with a shared clock abstracted as real-time event streams. The authors present a specification language and an online monitoring algorithm for the evaluation of the specifications.

In "Model Generation of Component-based Systems", Salva and Blot present an approach to learn the behavior of component-based systems from their execution traces using machine learning techniques. The resulting tool has been successfully experimented with network protocol traces.

The special issue was announced during the track on Software Verification and Testing (SVT) at the 33rd ACM/SIGAPP Symposium On Applied Computing. One of the papers accepted in this special issue is an extended version of a paper that originally appeared at the SVT track. Collectively, these papers tackle cutting-edge topics thereby appealing to the experts in the field.

Finally, the guest editors would like to thank the authors and the reviewers who contributed to the special issue.

References

