Preface: Special Issue on Software Verification and Testing

Mercedes G. Merayo, Gwen Salaün

PII: S0164-1212(17)30157-7
DOI: 10.1016/j.jss.2017.07.025
Reference: JSS 10008

To appear in: *The Journal of Systems & Software*

Preface: Special Issue on Software Verification and Testing

Software is now everywhere and guiding every part of our daily life (Web applications, smartphones, video games, cars, information systems, etc.). Designing and developing software systems has always been a tedious and error-prone task, and the ever increasing system complexity is making matters even worse. Although we are still far from providing widespread techniques and tools avoiding the existence of bugs in a system under development, we know how to automatically chase and find bugs that would be very difficult, if not impossible, to detect manually.

Software Verification and Testing is a research discipline of software engineering aiming at augmenting the quality of software by looking for bugs and checking that the developed software satisfies the expected requirements (functional and non-functional). There are several techniques for debugging and obtaining high quality software such as model checking, model-based testing, theorem proving, symbolic execution, run-time verification, fault diagnosis, or static analysis.

This Special Issue (SI) is dedicated to Software Verification and Testing (SVT) and more precisely aims at contributing to the challenge of improving the usability of formal methods in software engineering. This SI is a follow-up of the SVT track that we organized at the 31th ACM Symposium on Applied Computing, held in Pisa, Italy on April 3-8, 2016. The track received 58 full paper submissions. After a careful reviewing process, the international Program Committee decided to select 13 papers for presentation during the symposium and inclusion in the SAC'16 proceedings. From these 13 papers, the five best papers were selected and invited for an extended version to this special issue. We also received new submissions since an open call circulated for this SI. All the submissions went through a rigorous peer review process; four papers were finally accepted and are included in this special issue. These papers provide key insights on different formal verification and testing approaches.

The first paper, "*Implementation relations and probabilistic schedulers in the distributed test architecture*" by Hierons and Nuñez, presents a complete framework to formally test systems with distributed ports where some choices are probabilistically quantified while other choices are non-deterministic. They define different relations that state what it means for a system to be a valid implementation of a specification. These relations are defined using probabilistic schedulers, which resolve all the possible non-determinism, and can be used to compare purely probabilistic systems.

The second paper "*A method to localize faults in concurrent C programs*" by Alves, Cordeiro and Filho, describes a new approach to localize faults in concurrent programs, which is based on bounded model checking and sequentialization techniques. The main idea is to reproduce a faulty behavior in a sequential version of a concurrent program. Faulty lines in a program are identified by analyzing counterexamples generated by a model checker and by searching for