HAL
open science

Preface: Special issue on software verification and testing

Mercedes Merayo, Gwen Salaün

To cite this version:

Mercedes Merayo, Gwen Salaün. Preface: Special issue on software verification and testing. Journal of Systems and Software, 132, pp.317 - 318, 2017, 10.1016/j.jss.2017.07.025 . hal-01869106

HAL Id: hal-01869106

https://inria.hal.science/hal-01869106v1

Submitted on 6 Sep 2018

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: 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

a diagnostic value. This approach is useful to improve debugging of concurrent programs, since it indicates the line that should be corrected and the values leading to a successful execution.

The third paper "*Predictive runtime verification of timed properties*" by Pinisetty, Jéron, Tripakis, Falcone, Marchand and Preoteasa, presents a runtime verification approach of timed properties where the system is not entirely a black-box but something about its behaviour is known a priori. A priori knowledge about the behaviour of the system allows the verification monitor to foresee the satisfaction (or violation) of the monitored property. In addition to providing a conclusive verdict earlier, the verification monitor also provides additional information such as the minimum (maximum) time when the property can be violated (satisfied) in the future.

The fourth paper "*A feature matching and transfer approach for cross-company defect prediction*" by Yu, Jiang and Zhang, focuses on software defect prediction and presents a feature matching and transfer (FMT) approach to address the heterogeneous features of cross-company defect prediction (CCDP). This paper presents first a feature matching algorithm to convert the heterogeneous features into the matched features, and second a feature transfer approach to transfer the matched features from the source project to the target project for CCDP. Experiments were conducted on 16 datasets from NASA and PROMISE, and the results show that FMT is effective for CCDP.

Many people have contributed to the success of this special issue. Besides the authors of the papers, we would like to thank both the members of the Program Committee of the symposium (SAC-SVT'16) and the additional reviewers who kindly agreed to help us with the reviewing of the papers in this special issue. All carried out an excellent job during this demanding process: Bestoun S. Ahmed, Christian Attiogbe, Kyungmin Bae, Jongmoon Baik, Gianluca Barbon, Luciano Baresi, Sylvie Boldo, Mario Bravetti, Michael Butler, Javier Camara, Francisco Chicano, Peter Csaba Olveczky, Cristina David, Josh Dehlinger, Yuxin Deng, Anna Derezinska, Yliès Falcone, Huixing Fang, Lu Feng, Mamoun Filali, Mario Gleirscher, Ana Loreto González Hernández, Lindsay Groves, Henri Hansen, Hossein Hojjat, Claudia Iacob, Laura Inozemtseva, Marijn Jongerden, Marinos Kintis, Kais Klai, Nikolai Kosmatov, Hongjin Liang, Luis Llana, David Lo, Malte Lochau, Delphine Longuet, Benjamin Monmege, Shin Nakajima, Brian Nielsen, Meriem Ouederni, Francisco Palomo-Lozano, Marc Pantel, Mike Papadakis, Fayola Peters, Cosmin Radoi, Veselin Raychev, Arend Rensink, Antonella Santone, Olcay Taner Yildiz, Chouki Tibermacine, Stavros Tripakis, Antti Valmari, Anton Wijs, Xusheng Xiao, Lina Ye, Xin Ye, Eugen Zalinescu.

We hope that these articles will bring the reader novel insights and ideas.

Mercedes G. Merayo
Universidad Complutense de Madrid, Spain

Gwen Salaün
Université Grenoble Alpes, France

The Journal of Systems and Software (JSS) Guest Editors