



ELSEVIER

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico


Preface

Preface of Special issue on Automated Verification of Critical Systems (AVoCS'14)



AVoCS 2014, the *14th International Conference on Automated Verification of Critical Systems* has been hosted by the University of Twente, and has taken place in Enschede, Netherlands, on 24–26 September, 2014.

The aim of the AVoCS series is to contribute to the interaction and exchange of ideas among members of the international research community on tools and techniques for the verification of critical systems. The subject is to be interpreted broadly and inclusively. It covers all aspects of automated verification, including model checking, theorem proving, abstract interpretation, and refinement pertaining to various types of critical systems (safety-critical, security-critical, business-critical, performance-critical, etc.).

This special issue presents a selection of papers in this area of the AVoCS series. After the conference, there was an open call to contribute to this special issue. Authors of accepted papers were especially encouraged to submit an extended version of their paper. All submitted papers underwent a thorough reviewing process, with several iterations, where each paper was reviewed by several external domain experts. As a result of this process, out of seven interesting submissions, four high-quality papers presenting different aspects of automated verification of critical systems were selected to be published in this SCP special issue. The selected papers all give a different contribution to the automated verification of critical systems.

Andrea Bonacchi, Alessandro Fantechi, Stefano Bacherini, and Matteo Tempestini consider a highly relevant application domain for the AVoCS series, namely railway interlocking systems, which monitor the status of objects in a railway yard. They show how model extraction is used to validate an implementation of such a system, according to standard safety guidelines. In particular, the extracted model is used for testing, and also to verify several formal properties.

Ali Jafari, Ehsan Khamespanah, Marjan Sirjani, Holger Hermanns, and Matteo Cimini introduce PTRebeca, an actor-based language for asynchronous message passing with a notion of probabilistic time. They provide a semantics, and a tool set to automatically generate Markov Automaton models, which can be model checked for (probabilistic) reachability properties.

Jan Friso Groote, Remco van der Hofstad, and Matthias Raffelsieper propose a technique to estimate the state space size of a program. They model the program as a Cartesian product of a number of random graphs. The paper presents several computational studies. The estimations obtained are more accurate than the standard estimation technique based on a single random graph. This technique could be applied to estimate the remaining error probability after checking a part of the state space.

Petr Ročkai, Jiří Barnat, and Lubos Brim present an extension of their software model checker DIVINE for LLVM programs, to support the verification of programs with exceptions, which is an important step towards verification of real-world C++ programs.

Finally, we would like to thank all authors and reviewers. All reviewers carefully read the papers and provided constructive feedback to improve the papers. This feedback was carefully taken into account by the authors. This special issue would not have been possible without their efforts.

Guest Editors
 Marieke Huisman
 Jaco van de Pol
*Formal Methods and Tools, CTIT,
 University of Twente,
 The Netherlands*

E-mail addresses: M.Huisman@utwente.nl (M. Huisman), J.C.vandePol@utwente.nl (J. van de Pol)

29 April 2016
 Available online 6 May 2016