

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, Lancaster, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Zurich, Switzerland*

John C. Mitchell

*Stanford University, Stanford, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

C. Pandu Rangan

*Indian Institute of Technology Madras, Chennai, India*

Bernhard Steffen

*TU Dortmund University, Dortmund, Germany*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Gerhard Weikum

*Max Planck Institute for Informatics, Saarbrücken, Germany*

More information about this series at <http://www.springer.com/series/7407>

María del Mar Gallardo · Pedro Merino (Eds.)

# Model Checking Software

25th International Symposium, SPIN 2018  
Malaga, Spain, June 20–22, 2018  
Proceedings

*Editors*

María del Mar Gallardo  
University of Málaga  
Málaga  
Spain

Pedro Merino  
University of Málaga  
Málaga  
Spain

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-319-94110-3

ISBN 978-3-319-94111-0 (eBook)

<https://doi.org/10.1007/978-3-319-94111-0>

Library of Congress Control Number: 2018947326

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

© Springer International Publishing AG, part of Springer Nature 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

This volume contains the proceedings of the 25th International Symposium on Model Checking Software, SPIN 2018, held in Málaga, Spain, June 20–22, 2018. SPIN is a well-recognized periodic event started in 1995 around the model checking tool SPIN. Since 1995, the event has evolved and has been consolidated as a reference symposium in the area of formal methods related to model checking. The previous edition of the SPIN symposium took place in Santa Barbara (USA) with a record number of submissions and participants.

The SPIN 2018 edition requested regular papers, short papers, and tool demos in the following areas: formal verification techniques for automated analysis of software; formal analysis for modeling languages, such as UML/state charts; formal specification languages, temporal logic, design-by-contract; model checking, automated theorem proving, including SAT and SMT; verifying compilers; abstraction and symbolic execution techniques; static analysis and abstract interpretation; combination of verification techniques; modular and compositional verification techniques; verification of timed and probabilistic systems; automated testing using advanced analysis techniques; combination of static and dynamic analyses; derivation of specifications, test cases, or other useful material via formal analysis; case studies of interesting systems or with interesting results; engineering and implementation of software verification and analysis tools; benchmark and comparative studies for formal verification and analysis tools; formal methods education and training; and insightful surveys or historical accounts on topics of relevance to the symposium.

The symposium attracted 28 submissions, although two of them were rejected by the chairs because they were not within the scope of the symposium. Each of the remaining submissions was carefully reviewed by three Program Committee (PC) members. The selection process included further online discussion open to all PC members. Only the papers with positive global score were considered for acceptance. In addition, within these papers, only those with no objections from the PC members were accepted. As a result, 16 papers were selected for presentation at the symposium and publication in Springer’s proceedings. The program consisted of 14 regular papers, one short paper, and a demo-tool paper.

In addition to the accepted papers, the symposium included one invited tutorial by Irina Mariuca Asavae and Markus Roggenbach entitled “Software Model Checking for Mobile Security, Collusion Detection in K,” and three invited talks: “Efficient Runtime Verification of First-Order Temporal Properties” by Klaus Havelund and Doron Peled, “Applying Formal Methods to Advanced Embedded Controllers” by Rémi Delmas, and “Program Verification with Separation Logic” by Radu Iosif.

We would like to thank all the authors that submitted papers, the Steering Committee, the PC, the additional reviewers, the invited speakers, the participants, and the

local organizers for making SPIN 2018 a successful event. We also thank all the sponsors that provided logistics and financial support to make the symposium possible.

May 2018

María del Mar Gallardo  
Pedro Merino

# Organization

## Steering Committee

Dragan Bosnacki (Chair)	Eindhoven University of Technology, The Netherlands
Susanne Graf	Verimag, France
Gerard Holzmann	Nimble Research, USA
Stefan Leue	University of Konstanz, Germany
Neha Rungta	Amazon Web Services, USA
Jaco Van de Pol	University of Twente, The Netherlands
Willem Visser	Stellenbosch University, South Africa

## Program Committee

María Alpuente	Technical University of Valencia, Spain
Irina Mariuca Asavoe	Inria, France
Dragan Bosnacki	Eindhoven University of Technology, The Netherlands
Rance Cleaveland	University of Maryland, USA
Stefan Edelkamp	King's College London, UK
Hakan Erdogmus	Carnegie Mellon, USA
María del Mar Gallardo (Chair)	University of Málaga, Spain
Stefania Gnesi	CNR, Italy
Patrice Godefroid	Microsoft Research, USA
Klaus Havelund	NASA/Caltech Jet Propulsion Laboratory, USA
Gerard Holzmann	Nimble Research, USA
Radu Iosif	Verimag, France
Frédéric Lang	Inria, France
Kim Larsen	Aalborg University, Denmark
Stefan Leue	University of Konstanz, Germany
Alberto Lluch Lafuente	Technical University of Denmark, Denmark
Pedro Merino (Chair)	University of Málaga, Spain
Alice Miller	University of Glasgow, UK
Corina Pasareanu	CMU/NASA Ames, USA
Charles Pecheur	Université catholique de Louvain, Belgium
Doron Peled	Bar-Ilan University, Israel
Neha Rungta	Amazon Web Services, USA
Antti Valmari	University of Jyväskylä, Finland
Jaco Van de Pol	University of Twente, The Netherlands
Willem Visser	Stellenbosch University, South Africa
Farn Wang	National Taiwan University, Taiwan

## Additional Reviewers

Peter Aldous  
Mihail Asavoe  
Giovanni Bacci  
Georgiana Caltais  
Laura Carnevali  
Alessandro Fantechi  
Grigory Fedyukovich  
Martin Koelbl  
Florian Lorber  
Eric Mercer  
Marco Muniz  
Julia Sapiña  
Andrea Vandin

## Organizing Committee

Carlos Canal	University of Málaga, Spain
María del Mar Gallardo	University of Málaga, Spain
Pedro Merino	University of Málaga, Spain
Laura Panizo	University of Málaga, Spain

## Sponsors





## **Abstracts of Invited Papers**

# Software Model Checking for Mobile Security – Collusion Detection in $\mathbb{K}$

Irina Măriuca Asăvoae<sup>1</sup>, Hoang Nga Nguyen<sup>2</sup>,  
and Markus Roggenbach<sup>1</sup>

<sup>1</sup> Swansea University, UK

{I.M.Asavoe,M.Roggenbach}@swansea.ac.uk

<sup>2</sup> Coventry University, UK

Hoang.Nguyen@coventry.ac.uk

**Abstract.** Mobile devices pose a particular security risk because they hold personal details and have capabilities potentially exploitable for eavesdropping. The Android operating system is designed with a number of built-in security features such as application sandboxing and permission-based access control. Unfortunately, these restrictions can be bypassed, without the user noticing, by colluding apps whose combined permissions allow them to carry out attacks that neither app is able to execute by itself. In this paper, we develop a software model-checking approach within the  $\mathbb{K}$ -framework that is capable to detect collusion. This involves giving an abstract, formal semantics to Android applications and proving that the applied abstraction principles lead to a finite state space.

# Efficient Runtime Verification of First-Order Temporal Properties

Klaus Havelund<sup>1</sup> and Doron Peled<sup>2</sup>

<sup>1</sup> Jet Propulsion Laboratory, California Institute of Technology, USA

<sup>2</sup> Department of Computer Science, Bar Ilan University, Israel

**Abstract.** Runtime verification allows monitoring the execution of a system against a temporal property, raising an alarm if the property is violated. In this paper we present a theory and system for runtime verification of a first-order past time linear temporal logic. The first-order nature of the logic allows a monitor to reason about events with data elements. While runtime verification of propositional temporal logic requires only a fixed amount of memory, the first-order variant has to deal with a number of data values potentially growing unbounded in the length of the execution trace. This requires special compactness considerations in order to allow checking very long executions. In previous work we presented an efficient use of BDDs for such first-order runtime verification, implemented in the tool `DEJAVU`. We first summarize this previous work. Subsequently, we look at the new problem of dynamically identifying when data observed in the past are no longer needed, allowing to reclaim the data elements used to represent them. We also study the problem of adding relations over data values. Finally, we present parts of the implementation, including a new concept of user defined property macros.

---

The research performed by the first author was carried out at Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The research performed by the second author was partially funded by Israeli Science Foundation grant 2239/15: “Runtime Measuring and Checking of Cyber Physical Systems”.

# A Sample of Formal Verification Research for Embedded Control Software at ONERA

Rémi Delmas, Thomas Loquen, and Pierre Roux

ONERA Centre de Toulouse, 2 av. Édouard Belin, 31055 Toulouse, France  
{Rémi Delmas, Thomas Loquen, Pierre Roux}@onera.fr

**Abstract.** This talk presents a sample of research work conducted by the French Aerospace Lab (ONERA) on tailoring and applying formal methods to advanced embedded controllers, at various phases of the development and verification process, illustrated by industrial projects and collaborations. A first line of work<sup>1</sup>, carried out in partnership with Airbus, Dassault and LAAS-CNRS, aims at going beyond simulation for validating advanced hybrid control laws, by leveraging bounded reachability analysis and robustness analysis from the early design phases. This requires to bridge the representation gap existing between hybrid dataflow formalisms used to model control laws (e.g. Simulink, Scade-Hybrid,...), and the automata-based formalisms used by most hybrid model-checkers (e.g. SpaceEx, Flow\*, dReach,...) and robustness analysis frameworks. We discuss the steps taken to handle the complexity and size of typical industrial models. A second line of work<sup>1</sup>, carried out jointly with academic lab LRI (Paris-Sud, INRIA) and technology provider OcamlPro, addresses the sound combination of SMT-solvers and potentially unsound convex optimization engines to allow proving complex polynomial invariants on advanced control laws implementations. Such implementations are usually obtained by automatic time-discretization and code generation from a hybrid dataflow model. The proposed approach shows a notable performance improvement on controllers of interest with respect to earlier approaches based on interval arithmetic or purely symbolic methods such as cylindrical algebraic decomposition or virtual substitutions. Last, we present research conducted<sup>2</sup> in partnership with Liebherr Aerospace Toulouse and technology provider Systereel on leveraging model-checking techniques for unit-level test case generation for an air management system, taking into account the industrial setting and qualification constraints, following DO-178C and D0-333 guidelines.

**Keywords:** Hybrid dataflow models • Hybrid automata • Reachability analysis  
SMT solvers • Convex optimization • SAT solvers • Test case generation

---

<sup>1</sup> with funding from the French Civil Aviation Authority (DGAC) through the SEFA-IKKY program.

<sup>2</sup> with funding from the CIFRE program of the National Technological Research Agency (ANRT) and the RAPID program of the French Government Defense Procurement and Technology Agency (DGA) (project SATRUCT).

# Program Verification with Separation Logic

Radu Iosif

CNRS/VERIMAG/Université Grenoble Alpes, Grenoble, France

`Radu.Iosif@univ-grenoble-alpes.fr`

**Abstract.** Separation Logic is a framework for the development of modular program analyses for sequential, inter-procedural and concurrent programs. The first part of the paper introduces Separation Logic first from a historical, then from a program verification perspective. Because program verification eventually boils down to deciding logical queries such as the validity of verification conditions, the second part is dedicated to a survey of decision procedures for Separation Logic, that stem from either SMT, proof theory or automata theory. Incidentally we address issues related to decidability and computational complexity of such problems, in order to expose certain sources of intractability.

# Contents

## Tutorial and Invited Papers

Software Model Checking for Mobile Security – Collusion Detection in $\mathbb{K}$ . . .	3
<i>Irina Măriuca Asăvoae, Hoang Nga Nguyen, and Markus Roggenbach</i>	
Efficient Runtime Verification of First-Order Temporal Properties. . . . .	26
<i>Klaus Havelund and Doron Peled</i>	
Program Verification with Separation Logic . . . . .	48
<i>Radu Iosif</i>	

## Regular Papers

Petri Net Reductions for Counting Markings . . . . .	65
<i>Bernard Berthomieu, Didier Le Botlan, and Silvano Dal Zilio</i>	
Improving Generalization in Software IC3 . . . . .	85
<i>Tim Lange, Frederick Prinz, Martin R. Neuhäuser, Thomas Noll, and Joost-Pieter Katoen</i>	
Star-Topology Decoupling in SPIN . . . . .	103
<i>Daniel Gnad, Patrick Dubbert, Alberto Lluch Lafuente, and Jörg Hoffmann</i>	
Joint Forces for Memory Safety Checking . . . . .	115
<i>Marek Chalupa, Jan Strejček, and Martina Vitovská</i>	
Model-Checking HyperLTL for Pushdown Systems. . . . .	133
<i>Adrien Pommellet and Tayssir Touili</i>	
A Branching Time Variant of CaRet . . . . .	153
<i>Jens Oliver Gutsfeld, Markus Müller-Olm, and Benedikt Nordhoff</i>	
Control Strategies for Off-Line Testing of Timed Systems . . . . .	171
<i>Léo Henry, Thierry Jéron, and Nicolas Markey</i>	
An Extension of TRIANGLE Testbed with Model-Based Testing . . . . .	190
<i>Laura Panizo, Almudena Díaz, and Bruno García</i>	
Local Data Race Freedom with Non-multi-copy Atomicity. . . . .	196
<i>Tatsuya Abe</i>	

A Comparative Study of Decision Diagrams for Real-Time Model Checking . . . . .	216
<i>Omar Al-Bataineh, Mark Reynolds, and David Rosenblum</i>	
Lazy Reachability Checking for Timed Automata with Discrete Variables . . .	235
<i>Tamás Tóth and István Majzik</i>	
From SysML to Model Checkers via Model Transformation. . . . .	255
<i>Martin Kölbl, Stefan Leue, and Hargurbir Singh</i>	
Genetic Synthesis of Concurrent Code Using Model Checking and Statistical Model Checking. . . . .	275
<i>Lei Bu, Doron Peled, Dachuan Shen, and Yuan Zhuang</i>	
Quantitative Model Checking for a Controller Design . . . . .	292
<i>YoungMin Kwon and Eunhee Kim</i>	
Modelling Without a Modelling Language . . . . .	308
<i>Antti Valmari and Vesa Lappalainen</i>	
Context-Updates Analysis and Refinement in Chisel . . . . .	328
<i>Irina Măriuca Asăvoae, Mihail Asăvoae, and Adrián Riesco</i>	
<b>Author Index . . . . .</b>	<b>347</b>