Lecture Notes in Computer Science 11294

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

More information about this series at http://www.springer.com/series/7408

Ruzica Piskac · Philipp Rümmer (Eds.)

Verified Software

Theories, Tools, and Experiments

10th International Conference, VSTTE 2018
Oxford, UK, July 18–19, 2018
Revised Selected Papers

Springer

Editors
Ruzica Piskac
Yale University
New Haven, CT, USA

Philipp Rümmer
Uppsala University
Uppsala, Sweden

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-03591-4 ISBN 978-3-030-03592-1 (eBook)
https://doi.org/10.1007/978-3-030-03592-1

Library of Congress Control Number: 2018960421

LNCS Sublibrary: SL2 – Programming and Software Engineering

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the proceedings of the 10th International Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE 2018), held during July 18–19, 2018, as part of the Federated Logic Conference (FLoC) in Oxford, UK, and affiliated with the 30th International Conference on Computer-Aided Verification (CAV).

The goal of the VSTTE conference series is to advance the state of the art in the science and technology of software verification, through the interaction of theory development, tool evolution, and experimental validation. We solicited contributions describing significant advances in the production of verified software, i.e., software that has been proven to meet its functional specifications. Submissions of theoretical, practical, and experimental contributions were equally encouraged, including those that focus on specific problems or problem domains. We were especially interested in submissions describing large-scale verification efforts that involve collaboration, theory unification, tool integration, and formalized domain knowledge. We also welcomed papers describing novel experiments and case studies evaluating verification techniques and technologies. The topics of interest included education, requirements modeling, specification languages, specification/verification/certification case studies, formal calculi, software design methods, automatic code generation, refinement methodologies, compositional analysis, verification tools (e.g., static analysis, dynamic analysis, model checking, theorem proving, satisfiability), tool integration, benchmarks, challenges, and integrated verification environments.

The inaugural VSTTE conference was held at ETH Zurich in October 2005, and the following editions took place in Toronto (2008 and 2016), Edinburgh (2010), Philadelphia (2012), Atherton (2013), Vienna (2014), San Francisco (2015), and Heidelberg (2017).

This year there were 24 submissions. Each submission was reviewed by at least three Program Committee members. The committee decided to accept 19 papers for presentation at the conference. The program also included three invited talks, given by Cesare Tinelli (University of Iowa, USA), Stuart Matthews (Altran UK), and Rayna Dimitrova (University of Leicester, UK).

We would like to thank the invited speakers and the authors for their excellent contributions to the program this year, the Program Committee and external reviewers for diligently reviewing the submissions, and the organizers of FLoC and CAV 2018 for their help in organizing this event. We also thank Natarajan Shankar for his tireless stewardship of the VSTTE conference series over the years.

The VSTTE 2018 conference and the present volume were prepared with the help of EasyChair.

August 2018

Ruzica Piskac
Philipp Rümmer

Organization

Program Committee

June Andronick	CSIRO—Data61 and UNSW, Australia
Martin Brain	University of Oxford, UK
Michael Butler	University of Southampton, UK
Supratik Chakraborty	IIT Bombay, India
Roderick Chapman	Protean Code Limited, UK
Cristina David	University of Cambridge, UK
Dino Distefano	Facebook and Queen Mary University of London, UK
Mike Dodds	University of York, UK
Patrice Godefroid	Microsoft Research, USA
Arie Gurfinkel	University of Waterloo, Canada
Liana Hadarean	Synopsys, USA
Bart Jacobs	KU Leuven, Belgium
Swen Jacobs	CISPA and Saarland University, Germany
Cezary Kaliszyk	University of Innsbruck, Austria
Andy King	University of Kent, UK
Tim King	Google, USA
Vladimir Klebanov	SAP, Germany
Akash Lal	Microsoft Research, India
Nuno Lopes	Microsoft Research, UK
Alexander Malkis	Technical University of Munich, Germany
Yannick Moy	AdaCore, France
Gennaro Parlato	University of Southampton, UK
Andrei Paskevich	Université Paris-Sud, LRI, France
Ruzica Piskac	Yale University, USA
Markus Rabe	University of California, Berkeley, USA
Philipp Rümmer	Uppsala University, Sweden
Peter Schrammel	University of Sussex, UK
Natarajan Shankar	SRI International, USA
Tachio Terauchi	Waseda University, Japan
Mattias Ulbrich	Karlsruhe Institute of Technology, Germany
Philipp Wendler	LMU Munich, Germany
Thomas Wies	New York University, USA
Greta Yorsh	Queen Mary University of London, UK
Aleksandar Zeljić	Uppsala University, Sweden
Damien Zufferey	MPI-SWS, Germany

Additional Reviewers

Amani, Sidney
Ekici, Burak
Kirsten, Michael
Lewis, Corey
Margheri, Andrea
Paul, Lucas
Wang, Qingxiang
Winkler, Sarah

Abstracts of Invited Talks

Contract-based Compositional Verification of Infinite-State Reactive Systems

Cesare Tinelli

Department of Computer Science, The University of Iowa, USA
cesare-tinelli@uiowa.edu

Abstract. Model-based software development is a leading methodology for the construction of safety- and mission-critical embedded systems. Formal models of such systems can be validated, via formal verification or testing, against system-level requirements and modified as needed before the actual system is built. In many cases, source code can be even produced automatically from the model once the system designer is satisfied with it. As embedded systems become increasingly large and sophisticated, the size and complexity of models grows correspondingly, making the verification of top-level requirements harder, especially in the case of infinite-state systems. We argue that, as with conventional software, contracts are an effective mechanism to establish boundaries between components in a system model, and can be used to aid the verification of system-level properties by using compositional reasoning techniques. Component-level contracts also enable formal analyses that provide more accurate feedback to identify sources of errors or the parts of a system that contribute to the satisfaction of a given requirement. This talk discusses our experience in designing an assume-guarantee-based contract language on top of the Lustre modeling language and leveraging it to extend the Kind 2 model checker with contract-based compositional reasoning techniques.

Verified Software: Theories, Tools, … and Engineering

Stuart Matthews

Altran Technologies, SA
stuart.matthews@altran.com

Abstract. Continual innovation of software verification theories and tools is essential in order to meet the challenges of ever-more complex software-intensive systems. But achieving impact ultimately requires an understanding of the engineering context in which the tools will be deployed. Based on our tried-and-trusted methods of high-integrity software development at Altran, I will identify key features of the industrial landscape in which software verification tools have to operate, and some of the pitfalls that can stop them being adopted, including regulation, qualification, scalability, cost justification, and the overall tool ecosystem. Within this context I will present Altran's own on-going research and development activities in verified software technologies. The talk will conclude by drawing some key lessons that can be applied to avoid the traps and pitfalls that tools encounter on their journey to succesful deployment.

Synthesis of Surveillance Strategies for Mobile Sensors

Rayna Dimitrova

Department of Informatics, University of Leicester, UK
rd307@leicester.ac.uk

Abstract. The increasing application of formal methods to the design of autonomous systems often requires extending the existing specification and modeling formalisms, and addressing new challenges for formal verification and synthesis. In this talk, I will focus on the application of reactive synthesis to the problem of automatically deriving strategies for autonomous mobile sensors conducting surveillance, that is, maintaining knowledge of the location of a moving, possibly adversarial target. By extending linear temporal logic with atomic surveillance predicates, complex temporal surveillance objectives can be formally specified in a way that allows for seamless combination with other task specifications. I will discuss two key challenges for applying state-of-the-art methods for reactive synthesis to temporal surveillance specifications. First, naively keeping track of the knowledge of the surveillance agent leads to a state-space explosion. Second, while sensor networks with a large number of dynamic sensors can achieve better coverage, synthesizing coordinated surveillance strategies is challenging computationally. I will outline how abstraction, refinement, and compositional synthesis techniques can be used to address these challenges.

The talk is based on joint work with Suda Bharadwaj and Ufuk Topcu.

Contents