Lecture Notes in Computer Science **11244**

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

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

Tiziana Margaria · Bernhard Steffen (Eds.)

Leveraging Applications of Formal Methods, Verification and Validation

Modeling

8th International Symposium, ISoLA 2018
Limassol, Cyprus, November 5–9, 2018
Proceedings, Part I

Springer

Editors
Tiziana Margaria
University of Limerick
Limerick, Ireland

Bernhard Steffen
TU Dortmund
Dortmund, Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-03417-7 ISBN 978-3-030-03418-4 (eBook)
https://doi.org/10.1007/978-3-030-03418-4

Library of Congress Control Number: 2018960390

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

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

Preface

Welcome to ISoLA 2018, the *8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*, that was held in Limassol (Cyprus) during November 5–9, 2018, endorsed by EASST, the European Association of Software Science and Technology.

This year's event followed the tradition of its symposia forerunners held 2004 and 2006 in Cyprus, 2008 in Chalkidiki, 2010 and 2012 in Crete, 2014 and 2016 in Corfu, and the series of ISoLA Workshops in Greenbelt (USA) in 2005, Poitiers (France) in 2007, Potsdam (Germany) in 2009, in Vienna (Austria) in 2011, and 2013 in Palo Alto (USA).

As in the previous editions, ISoLA 2018 provided a forum for developers, users, and researchers to discuss issues related to the **adoption and use of rigorous tools and methods** for the specification, analysis, verification, certification, construction, test, and maintenance of systems from the point of view of their different application domains. Thus, since 2004 the ISoLA series of events has served the purpose of bridging the gap between designers and developers of rigorous tools on one hand, and users in engineering and in other disciplines on the other hand. It fosters and exploits synergetic relationships among scientists, engineers, software developers, decision makers, and other critical thinkers in companies and organizations. By providing a specific, dialogue-oriented venue for the discussion of common problems, requirements, algorithms, methodologies, and practices, ISoLA aims in particular at supporting researchers in their quest to improve the usefulness, reliability, flexibility, and efficiency of tools for building systems, and users in their search for adequate solutions to their problems.

The program of the symposium consisted of a collection of *special tracks* devoted to the following hot and emerging topics:

- A Broader View on Verification: From Static to Runtime and Back
  (Organizers: Wolfgang Ahrendt, Marieke Huisman, Giles Reger, Kristin Yvonne Rozier)
- Evaluating Tools for Software Verification
  (Organizers: Markus Schordan, Dirk Beyer, Stephen F. Siegel)
- Towards a Unified View of Modeling and Programming
  (Organizers: Manfred Broy, Klaus Havelund, Rahul Kumar, Bernhard Steffen)
- RV-TheToP: Runtime Verification from Theory to Industry Practice
  (Organizers: Ezio Bartocci and Ylies Falcone)
- Rigorous Engineering of Collective Adaptive Systems
  (Organizers: Rocco De Nicola, Stefan Jähnichen, Martin Wirsing)
- Reliable Smart Contracts: State of the Art, Applications, Challenges, and Future Directions
  (Organizers: Gerardo Schneider, Martin Leucker, César Sánchez)

- Formal Methods in Industrial Practice—Bridging the Gap
  (Organizers: Michael Felderer, Dilian Gurov, Marieke Huisman, Björn Lisper, Rupert Schlick)
- X-by-Construction
  (Organizers: Maurice H. ter Beek, Loek Cleophas, Ina Schaefer, and Bruce W. Watson)
- Statistical Model Checking
  (Organizers: Axel Legay and Kim Larsen)
- Verification and Validation of Distributed Systems
  (Organizer: Cristina Seceleanu)
- Cyber-Physical Systems Engineering
  (Organizers: J Paul Gibson, Marc Pantel, Peter Gorm Larsen, Jim Woodcock, John Fitzgerald)

The following events were also held:

- RERS: Challenge on Rigorous Examination of Reactive Systems (Bernhard Steffen)
- Doctoral Symposium and Poster Session (Anna-Lena Lamprecht)
- Industrial Day (Axel Hessenkämper, Falk Howar, Andreas Rausch)

Co-located with the ISoLA Symposium were:

- RV 2018: 18th International Conference on Runtime Verification (Saddek Bensalem, Christian Colombo, and Martin Leucker)
- STRESS 2018: 5th International School on Tool-based Rigorous Engineering of Software Systems (John Hatcliff, Tiziana Margaria, Robby, Bernhard Steffen)

Owing to the growth of ISoLA 2018, the proceedings of this edition are published in four volumes of LNCS: Part 1: Modeling, Part 2: Verification, Part 3: Distributed Systems, and Part 4: Industrial Practice. In addition to the contributions of the main conference, the proceedings also include contributions of the four embedded events and tutorial papers for STRESS.

We thank the track organizers, the members of the Program Committee and their referees for their effort in selecting the papers to be presented, the local Organization Chair, Petros Stratis, the EasyConferences team for their continuous precious support during the week as well as during the entire two-year period preceding the events, and Springer for being, as usual, a very reliable partner in the proceedings production. Finally, we are grateful to Kyriakos Georgiades for his continuous support for the website and the program, and to Markus Frohme and Julia Rehder for their help with the online conference service (EquinOCS).

Special thanks are due to the following organization for their endorsement: EASST (European Association of Software Science and Technology) and Lero – The Irish Software Research Centre, and our own institutions: TU Dortmund and the University of Limerick.

November 2018

Tiziana Margaria
Bernhard Steffen

Organization

Symposium Chair

Bernhard Steffen	TU Dortmund, Germany

Program Chair

Bernhard Steffen	TU Dortmund, Germany

Program Committee

Wolfgang Ahrendt	Chalmers University of Technology, Sweden
Jesper Andersen	Deon Digital AG
Ezio Bartocci	TU Wien, Austria
Dirk Beyer	LMU Munich, Germany
Manfred Broy	Technische Universität München
Loek Cleophas	TU Eindhoven, The Netherlands
Rocco De Nicola	IMT School for Advanced Studies, Italy
Boris Düdder	University of Copenhagen, Denmark
Ylies Falcone	University of Grenoble, France
Michael Felderer	University of Innsbruck, Austria
John Fitzgerald	Newcastle University, UK
Paul Gibson	Telecom Sud Paris, France
Kim Guldstrand Larsen	Aalborg University, Denmark
Dilian Gurov	KTH Royal Institute of Technology, Sweden
John Hatcliff	Kansas State University, USA
Klaus Havelund	Jet Propulsion Laboratory, USA
Fritz Henglein	University of Copenhagen, Denmark
Axel Hessenkämper	Hottinger Baldwin Messtechnik GmbH
Falk Howar	Dortmund University of Technology and Fraunhofer ISST, Germany
Marieke Huisman	University of Twente, The Netherlands
Michael Huth	Imperial College London, UK
Stefan Jaehnichen	TU Berlin, Germany
Rahul Kumar	Microsoft Research
Anna-Lena Lamprecht	Utrecht University, The Netherlands
Peter Gorm Larsen	Aarhus University, Denmark
Axel Legay	Inria, France
Martin Leucker	University of Lübeck, Germany

Björn Lisper	Mälardalen University, Sweden
Leif-Nissen Lundæk	XAIN AG
Tiziana Margaria	Lero, Ireland
Marc Pantel	Université de Toulouse, France
Andreas Rausch	TU Clausthal, Germany
Giles Reger	University of Manchester, UK
Robby	Kansas State University, USA
Kristin Yvonne Rozier	Iowa State University, USA
Ina Schaefer	TU Braunschweig, Germany
Rupert Schlick	AIT Austrian Institute of Technology, Austria
Gerardo Schneider	University of Gothenburg, Sweden
Markus Schordan	Lawrence Livermore National Laboratory, USA
Cristina Seceleanu	Mälardalen University, Sweden
Stephen F. Siegel	University of Delaware, USA
César Sánchez	IMDEA Software Institute, Spain
Bruce W. Watson	Stellenbosch University, South Africa
Martin Wirsing	LMU München, Germany
James Woodcock	University of York, UK
Maurice ter Beek	ISTI-CNR, Italy
Jaco van de Pol	University of Twente, The Netherlands

Additional Reviewers

Yehia Abd Alrahman
Dhaminda Abeywickrama
Lenz Belzner
Saddek Bensalem
Egon Boerger
Marius Bozga
Tomas Bures
Rance Cleaveland
Giovanna Di Marzo Serugendo
Matthew Dwyer
Benedikt Eberhardinger
Rim El Ballouli
Thomas Gabor
Stephen Gilmore
Emma Hart
Arnd Hartmanns
Rolf Hennicker
Petr Hnetynka
Reiner Hähnle
Patrik Jansson
Einar Broch Johnsen
Neil Jones
Sebastiaan Joosten
Gabor Karsai
Alexander Knapp
Timothy Lethbridge
Chunhua Liao
Alberto Lluch-Lafuente
Alessandro Maggi
Dominique Méry
Birger Møller-Pedersen
Stefan Naujokat
Ayoub Nouri
Liam O'Connor
Doron Peled
Thomy Phan
Jeremy Pitt
Hella Ponsar
Andre Reichstaller
Jeff Sanders
Sean Sedwards
Christoph Seidl

Bran Selic
Steven Smyth
Josef Strnadel
Jan Sürmeli
Louis-Marie Traonouez
Mirco Tribastone
Andrea Vandin
Markus Voelter
Franco Zambonelli
Natalia Zon

(Some) Security by Construction Through a LangSec Approach (X-by-Construction)

Erik Poll

Digital Security Group, Radboud University Nijmegen, The Netherlands
erikpoll@cs.ru.nl

This talk discusses some good and bad experiences in applying formal methods to security and sketches directions for using formal methods to improve security using insights from the LangSec (language-based security) paradigm.

On the face of it, security looks like a promising application area for formal methods. Cyber security is a huge and still growing concern. It is widely recognized that security should be addressed *throughout* the software development life cycle, ideally by practising so-called Security-by-Design, and not bolted on later as an afterthought; this means that formal methods for security could be applied at any stage of the software development life cycle, from the earliest stages of requirements engineering to the final stages such as pen-testing or patching.

Still, all this is easier said than done. Security requirements can be tricky to formalise – or even to spot at all – and it can be difficult to say what it means for an application to be secure. It is often easier to say what may make an application insecure, as is done by lists of standard security flaws such as the OWASP Top Ten[1] or the CWE/SANS Top 25[2]. Such lists are very useful, but always incomplete, and lend themselves more naturally to testing for certain types of security flaws post-hoc than to guaranteeing their absence by construction.

A more constructive approach to security can be taken by realising that security problems typically arise in interactions and exploit the *languages* used in these interactions. The most obvious example is the interaction between an attacker and a system, where the attacker tries to abuse the interface the system exposes. This interface can be a network protocol, but it may also involve a file format, say JPEG, or a language such as HTML. Security problems can also arise in the interaction between two applications (or an application and an external service) even if neither of them is malicious. Classic examples here are the interaction between a web application and its back-end database, where SQL injection becomes a worry, or the interaction between a web application and the browser, where XSS becomes a worry.

The LangSec paradigm[3] highlights the central role played by the languages used for these interactions – e.g. file formats, protocols, or query languages – in causing security problems. Root causes of security problems identified are: the large number of these

[1] https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

[2] https://cwe.mitre.org/top25/.

[3] See http://langsec.org, esp. http://langsec.org/bof-handout.pdf, or [5].

languages, their complexity, their expressivity, the lack of clear specifications, and finally the fact that parsers to process these languages are hand-written, and often mix parsing and processing of inputs.

This also provides a clear way forward in using formal methods to improve security, namely by providing formal descriptions of the input languages involved and using these descriptions to generate parser code, thus getting at least some security by construction. Ironically, formalisms for describing languages are some of the best-known and most basic formal methods around, and parsing is one of the oldest and best understood parts of computer science, with plenty of tools for generating code. So it is a bit of an embarrassment to the computer science community that this is where modern IT screws up so badly, with so many security flaws. In addition to parsers, one would also like to generate unparsers (aka pretty-printers or serialisers), as interactions between systems typically involve an unparser at one end and a parser at the other end. Recent initiatives here include Hammer [2] and Nail [1]. Formal descriptions of input languages can also be used for testing, in test generation or as test oracles.

Even if we get rid of all (un)parser bugs, there remains the risk of *unintentionally* parsing some inputs [7], especially inputs coming from sources that an attacker can control. Here formal methods can also help, with data flow analysis to trace where data comes from and/or where it might end up. Ideally, such data flows can then be controlled by a type system, where different types explicitly distinguish the various languages that the application handles (e.g. to avoid the chance of accidentally processing a user name or a fragment of HTML as an SQL statement), the various trust levels associated with different input channels (e.g. to distinguish tainted inputs from untainted data), or both. As these types can be application-specific, it is natural to use extensible type systems for this, e.g. using type qualifiers [4] or type annotations [3], or to turn to domain-specific languages [6].

References

1. Bangert, J., Zeldovich, N.: Nail: a practical interface generator for data formats. In: Security and Privacy Workshops (SPW), 2014, pp. 158–166. IEEE (2014)
2. Bratus, S., Crain, A.J., Hallberg, S.M., Hirsch, D.P., Patterson, M.L., Koo, M., Smith, S.W.: Implementing a vertically hardened DNP3 control stack for power applications. In: Industrial Control System Security Workshop (ICSS'16), pp. 45–53. ACM (2016)
3. Dietl, W., Dietzel, S., Ernst, M.D., Muşlu, K., Schiller, T.W.: Building and using pluggable type-checkers. In: ICSE'11, pp. 681–690. ACM (2011)
4. Foster, J.S., Terauchi, T., Aiken, A.: Flow-sensitive type qualifiers. In: PLDI'02, SIGPLAN Notices, vol. 37, pp. 1–12. ACM (2002)
5. Momot, F., Bratus, S., Hallberg, S.M., Patterson, M.L.: The seven turrets of Babel: a taxonomy of LangSec errors and how to expunge them. In: Cybersecurity Development (SecDev), pp. 45–52. IEEE (2016)
6. Omar, C., Kurilova, D., Nistor, L., Chung, B., Potanin, A., Aldrich, J.: Safely composable type-specific languages. In: Jones, R. (ed.) ECOOP 2014 – Object-Oriented Programming, ECOOP 2014. LNCS, vol, 8586, pp. 105–130. Springer, Heidelber (2014)
7. Poll, E.: LangSec revisited: input security flaws of the second kind. In: Symposium on Security and Privacy Workshops (SPW). IEEE (2018)

Contents – Part I

Towards a Unified View of Modeling and Programming