

*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, Zürich, 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, 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/7408>

Sanjai Rayadurgam · Oksana Tkachuk (Eds.)

# NASA Formal Methods

8th International Symposium, NFM 2016  
Minneapolis, MN, USA, June 7–9, 2016  
Proceedings

*Editors*  
Sanjai Rayadurgam  
University of Minnesota  
Minneapolis, MN  
USA

Oksana Tkachuk  
NASA Ames Research Center  
Moffett Field, CA  
USA

ISSN 0302-9743                      ISSN 1611-3349 (electronic)  
Lecture Notes in Computer Science  
ISBN 978-3-319-40647-3              ISBN 978-3-319-40648-0 (eBook)  
DOI 10.1007/978-3-319-40648-0

Library of Congress Control Number: 2016941084

LNCS Sublibrary: SL2 – Programming and Software Engineering

© Springer International Publishing Switzerland 2016

**Open Access** Chapters 3 and 8 are distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). For further details see license information in the chapters.

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.

Printed on acid-free paper

This Springer imprint is published by Springer Nature  
The registered company is Springer International Publishing AG Switzerland

# Preface

The NASA Formal Methods (NFM) Symposium is a forum for theoreticians and practitioners from academia, industry, and government, with the goals of identifying challenges and providing solutions to achieving assurance in mission- and safety-critical systems. Examples of such systems at NASA include advanced separation assurance algorithms for aircraft, Next-Generation Air Transportation (NextGen), autonomous rendezvous and docking for spacecraft, on-board software for Unmanned Aerial Systems (UAS), UAS Traffic Management (UTM), autonomous robots, and systems for fault detection, diagnosis, and prognostics. The topics covered by the NASA Formal Methods Symposia include: model checking, theorem proving, SAT and SMT solving, symbolic execution, automated testing and simulation, model-based development, static and dynamic analysis techniques, runtime verification, safety assurance, fault tolerance, compositional verification techniques, cyber security, specification formalisms, requirements analysis, certification, and applications of formal methods in systems development.

This volume contains the papers presented at NFM 2016, the 8th NASA Formal Methods Symposium, co-organized by NASA Ames Research Center and the University of Minnesota Software Engineering Center, in Minneapolis, MN, June 7–9, 2016. Previous symposia were held in Pasadena, CA (2015), Houston, TX (2014), Moffett Field, CA (2013), Norfolk, VA (2012), Pasadena, CA (2011), Washington, DC (2010), and Moffett Field, CA (2009). The series started as the Langley Formal Methods Workshop, and was held under that name in 1990, 1992, 1995, 1997, 2000, and 2008. Papers were solicited for NFM 2016 under two categories: regular papers describing fully developed work and complete results or case studies, and short papers describing tools, experience reports, and work in progress or preliminary results. The symposium received 70 submissions for review (51 regular papers and 19 short papers) out of which 29 were accepted for publication (19 as regular papers and 10 as short papers). These submissions went through a rigorous reviewing process, where each paper was first independently reviewed by three reviewers and then subsequently discussed by the Program Committee.

In addition to the refereed papers, the symposium featured three invited presentations: “Using Formal Methods to Eliminate Exploitable Bugs” by Kathleen Fisher, Professor in the Computer Science Department at Tufts University; “Where Formal Methods Might Find Application on Future NASA Missions” by Michael L. Aguilar, NASA Technical Fellow in Software Engineering and the NASA Engineering and Safety Center Discipline Expert in Software, NASA Langley Research Center; and “Murphy Was Here” by Kevin Driscoll, Engineering Fellow, Honeywell. The symposium also featured breakout sessions to explore the application of formal methods to future NASA missions and to connect the dots between capabilities that need to be matured for NASA missions and formal methods.

The organizers are grateful to the authors for submitting their work to NFM 2016 and to the invited speakers for sharing their insights. NFM 2016 would not have been possible without the collaboration of the outstanding Program Committee and additional reviewers, the support of the Steering Committee, the efforts of the staff at the University of Minnesota and NASA Ames Research Center who made this event possible, and the general support of the NASA Formal Methods community.

The NFM 2016 website can be found at: <http://nasaformalmethods.org>.

Support for the preparation of these proceedings was provided under a contract between the NASA Ames Research Center and the University of Minnesota Software Engineering Center.

May 2016

Sanjai Rayadurgam  
Oksana Tkachuk

# Organization

## Steering Committee

Julia Badger	NASA Johnson Space Center, USA
Ben Di Vito	NASA Langley Research Center, USA
Klaus Havelund	NASA Jet Propulsion Laboratory, USA
Gerard Holzmann	NASA Jet Propulsion Laboratory, USA
Michael Lowry	NASA Ames Research Center, USA
Kristin Yvonne Rozier	University of Cincinnati, USA
Johann Schumann	SGT, Inc./NASA Ames Research Center, USA

## Organizing Committee

Michael Lowry	NASA Ames Research Center, USA (NASA Liaison)
Johann Schumann	SGT, Inc./NASA Ames Research Center, USA (General Chair)
Oksana Tkachuk	SGT, Inc./NASA Ames Research Center, USA (PC Chair)
Sanjai Rayadurgam	University of Minnesota, USA (PC Chair)
Mike Whalen	University of Minnesota, USA (Financial Chair)
Mats Heimdahl	University of Minnesota, USA (Local Arrangements Chair)

## Program Committee

Julia Badger	NASA Johnson Space Center, USA
Clark Barrett	New York University, USA
Saddek Bensalem	Verimag and University Joseph Fourier, France
Dirk Beyer	University of Passau, Germany
Borzoo Bonakdarpour	McMaster University, Canada
Alessandro Cimatti	FBK, Italy
Darren Cofer	Rockwell Collins, Inc., USA
Myra Cohen	University of Nebraska-Lincoln, USA
Misty Davies	NASA Ames Research Center, USA
Leonardo de Moura	Microsoft, USA
Ben Di Vito	NASA Langley Research Center, USA
Alexandre Duret-Lutz	LRDE/EPITA, France
Andrew Gacek	Rockwell Collins, Inc., USA
Pierre-Loic Garoche	ONERA, France
Shalini Ghosh	SRI International, USA

Susanne Graf	Universite Joseph Fourier/CNRS/VERIMAG, France
Radu Grosu	Vienna University of Technology, Austria
Arie Gurfinkel	SEI, Carnegie Mellon University, USA
Klaus Havelund	NASA Jet Propulsion Laboratory, USA
Constance Heitmeyer	Naval Research Laboratory, USA
Gerard Holzmann	NASA Jet Propulsion Laboratory, USA
Falk Howar	TU Clausthal/IPSSE, Germany
Rajeev Joshi	NASA Jet Propulsion Laboratory, USA
Dejan Jovanović	SRI International, USA
Gerwin Klein	NICTA and University of New South Wales, Australia
Daniel Kroening	University of Oxford, UK
Rahul Kumar	NASA Jet Propulsion Laboratory, USA
Michael Lowry	NASA Ames Research Center, USA
Célia Martinie	ICS-IRIT, Université Paul Sabatier, France
Eric Mercer	Brigham Young University, USA
Cesar Munoz	NASA Langley Research Center, USA
Jorge A. Navas	SGT, Inc./NASA Ames Research Center, USA
Natasha Neogi	NASA Langley Research Center, USA
Ganesh Pai	SGT, Inc./NASA Ames Research Center, USA
Charles Pecheur	Université Catholique de Louvain, Belgium
Lee Pike	Galois, Inc., USA
Andreas Podelski	University of Freiburg, Germany
Pavithra Prabhakar	Kansas State University, USA
Venkatesh Prasad Ranganath	Kansas State University, USA
Franco Raimondi	Middlesex University, UK
Sanjai Rayadurgam	University of Minnesota, USA
Kristin Yvonne Rozier	University of Cincinnati, USA
Neha Rungta	SGT, Inc./NASA Ames Research Center, USA
Oleg Sokolsky	University of Pennsylvania, USA
Oksana Tkachuk	SGT, Inc./NASA Ames Research Center, USA
Stefano Tonetta	FBK, Italy
Willem Visser	Stellenbosch University, South Africa
Virginie Wiels	ONERA/DTIM, France
Guowei Yang	Texas State University, USA

## Additional Reviewers

Archer, Myla	Dangl, Matthias
Astefanoaei, Lacramioara	David, Cristina
Backes, John	Dureja, Rohit
Brain, Martin	Dutle, Aaron
Calderon, Jose	Faghih, Fathiyeh
Cheng, Chih-Hong	Falcone, Ylies



Friedberger, Karlheinz  
Goodloe, Alwyn  
Kahsai, Temesghen  
Kalla, Priyank  
Kumar, Ramana  
Kupferman, Orna  
Lal, Ratan  
Lukina, Anna  
Mukherjee, Rajdeep  
Murray, Toby  
Pit-Claudiel, Clément  
Poplavko, Peter  
Prokesch, Daniel

Roveri, Marco  
Schrammel, Peter  
Schäf, Martin  
Selyunin, Konstantin  
Sewell, Thomas  
Siddique, Umair  
Soto, Miriam Garcia  
Svendsen, Kasper  
Tomb, Aaron  
Urban, Caterina  
Vizel, Yakir  
Wasicek, Armin

## **Abstracts of Invited Talks**

# Using Formal Methods to Eliminate Exploitable Bugs

Kathleen Fisher

Tufts University, Medford, MA 02155  
kfisher@eecs.tufts.edu

**Abstract.** For decades, formal methods have offered the promise of software that doesn't have exploitable bugs. Until recently, however, it hasn't been possible to verify software of sufficient complexity to be useful. Recently, that situation has changed. SeL4 is an open-source operating system microkernel efficient enough to be used in a wide range of practical applications. It has been proven to be fully functionally correct, ensuring the absence of buffer overflows, null pointer exceptions, use-after-free errors, etc., and to enforce integrity and confidentiality properties. The CompCert Verifying C Compiler maps source C programs to provably equivalent assembly language, ensuring the absence of exploitable bugs in the compiler.

A number of factors have enabled this revolution in the formal methods community, including increased processor speed, better infrastructure like the Isabelle/HOL and Coq theorem provers, specialized logics for reasoning about low-level code, increasing levels of automation afforded by tactic languages and SAT/SMT solvers, and the decision to move away from trying to verify existing artifacts and instead focus on co-developing the code and the correctness proof.

In this talk, I will explore the promise and limitations of current formal methods techniques for producing useful software that provably does not contain exploitable bugs. I will discuss these issues in the context of DARPA's HACMS program, which has as its goal the creation of high-assurance software for vehicles, including quad-copters, helicopters, and automobiles.

# Where Formal Methods Might Find Application on Future NASA Missions

Michael L. Aguilar

NASA Langley Research Center, Hampton, VA 23681  
Michael.L.Aguilar@nasa.gov

**Abstract.** In many cases, formal methods are a solution looking for a problem. NASA recently released the 2015 NASA Technology Roadmaps that describe numerous possible future missions. Within these descriptions are capabilities that need to be matured in order for mission success. Many of these future capabilities could be accomplished through the use of formal methods. The future capabilities identified by NASA in these roadmaps may just be the problems formal methods have been seeking. Think of these roadmaps as “on-ramps” for engineering using formal methods.

These missions include joint robotic and human exploration of Mars, robotic probes of the icy moons of the outer planets where there is evidence of organic chemistry. Sophisticated earth-orbiting satellites to advance earth science, and possible robotic refueling and maintenance missions of these satellites.

One of the predominant cross-cutting challenges is autonomy and its verification: the capability of automation to make and execute decisions in-situ; necessitated in part by the long light-time delays from Earth for deep space spacecraft. Another challenge is the high expense of achieving high assurance for software intensive systems.

And then there are the overarching issues of budget, schedule, and design. It is highly unlikely these system-of-systems will be implemented and interfaced, tested and verified, before deployment. How could formal methods define the requirements for these systems such that the protocols and interfaces, functions and fault management execute as intended for integration that may occur for the first time off-planet?

In my experience, NASA can accept new techniques where it can be demonstrated that current practices are not sufficient. For these future system-of-systems, formal methods may prove to be not only sufficient but necessary.

# Murphy Was Here

Kevin Driscoll

Honeywell, Golden Valley, Minnesota 55422  
kevin.driscoll@honeywell.com

**Abstract.** My boss once said that “All system failures are caused by design faults.” This is because, regardless of the requirements, critical systems should be designed to never fail. It is extremely rare for a critical system to fail in a way that was anticipated by the designers (e.g., redundancy exhaustion). This keynote will explore the factors that lead to designers underestimating the possibility/probabilities of certain failures. Examples of rare, but actually occurring, failures will be given. These will include Byzantine faults, component transmutation, “evaporating” software, and exhaustively tested software that still failed. Problems that Formal Methods could have found before actual occurrence will be identified as well as problems that are still intractable with the current state of the art. The well known Murphy’s Law states that: “If anything can go wrong, it will go wrong.” For critical systems, the following should be added: “And, if anything can’t go wrong, it will go wrong anyway.”

# Contents

## Requirements and Architectures

Temporal Logic Framework for Performance Analysis of Architectures of Systems . . . . .	3
<i>Ariane Piel, Jean Bourrely, Stéphanie Lala, Sylvain Bertrand, and Romain Kervarc</i>	
On Implementing Real-Time Specification Patterns Using Observers . . . . .	19
<i>John D. Backes, Michael W. Whalen, Andrew Gacek, and John Komp</i>	
Contract-Based Verification of Complex Time-Dependent Behaviors in Avionic Systems . . . . .	34
<i>Devesh Bhatt, Arunabh Chattopadhyay, Wenchao Li, David Oglesby, Sam Owre, and Natarajan Shankar</i>	
ARSENAL: Automatic Requirements Specification Extraction from Natural Language . . . . .	41
<i>Shalini Ghosh, Daniel Elenius, Wenchao Li, Patrick Lincoln, Natarajan Shankar, and Wilfried Steiner</i>	

## Testing and Run-Time Enforcement

Assisted Coverage Closure . . . . .	49
<i>Adam Nellis, Pascal Kesseli, Philippa Ryan Conmy, Daniel Kroening, Peter Schrammel, and Michael Tautschnig</i>	
Synthesizing Runtime Enforcer of Safety Properties Under Burst Error . . . . .	65
<i>Meng Wu, Haibo Zeng, and Chao Wang</i>	
Compositional Runtime Enforcement . . . . .	82
<i>Srinivas Pinisetty and Stavros Tripakis</i>	
Improving an Industrial Test Generation Tool Using SMT Solver . . . . .	100
<i>Hao Ren, Devesh Bhatt, and Jan Hvozdic</i>	
The comKorat Tool: Unified Combinatorial and Constraint-Based Generation of Structurally Complex Tests . . . . .	107
<i>Hua Zhong, Lingming Zhang, and Sarfraz Khurshid</i>	

**Code Generation and Synthesis**

Automated Synthesis of Safe Autonomous Vehicle Control Under Perception Uncertainty . . . . . 117  
*Susmit Jha and Vasumathi Raman*

Obfuscator Synthesis for Privacy and Utility. . . . . 133  
*Yi-Chin Wu, Vasumathi Raman, Stéphane Lafortune, and Sanjit A. Seshia*

Code Generation Using a Formal Model of Reference Counting . . . . . 150  
*Gaspard Férey and Natarajan Shankar*

EventB2Java: A Code Generator for Event-B . . . . . 166  
*Néstor Cataño and Victor Rivera*

**Applications of Formal Methods**

A Formally Verified Checker of the Safe Distance Traffic Rules for Autonomous Vehicles. . . . . 175  
*Albert Rizaldi, Fabian Immler, and Matthias Althoff*

Probabilistic Formal Verification of the SATS Concept of Operation . . . . . 191  
*Muhammad Usama Sardar, Nida Afaq, Khaza Anuarul Hoque, Taylor T. Johnson, and Osman Hasan*

Formal Translation of IEC 61131-3 Function Block Diagrams to PVS with Nuclear Application . . . . . 206  
*Josh Newell, Linna Pang, David Tremaine, Alan Wassying, and Mark Lawford*

Formal Analysis of Extended Well-Clear Boundaries for Unmanned Aircraft . . . . . 221  
*César Muñoz and Anthony Narkawicz*

Formal Validation and Verification Framework for Model-Based and Adaptive Control Systems . . . . . 227  
*Sergio Guarro, Umit Ozguner, Tunc Aldemir, Matt Knudson, Arda Kurt, Michael Yau, Mohammad Hejase, and Steve Kwon*

**Techniques for Automated Verification**

Verifying Relative Safety, Accuracy, and Termination for Program Approximations . . . . . 237  
*Shaobo He, Shuvendu K. Lahiri, and Zvonimir Rakamarić*

Bandwidth and Wavefront Reduction for Static Variable Ordering  
in Symbolic Reachability Analysis . . . . . 255  
*Jeroen Meijer and Jaco van de Pol*

Gray-Box Learning of Serial Compositions of Mealy Machines . . . . . 272  
*Andreas Abel and Jan Reineke*

**Theorem Proving and Proofs**

Specification and Proof of High-Level Functional Properties of Bit-Level  
Programs . . . . . 291  
*Clément Fumex, Claire Dross, Jens Gerlach, and Claude Marché*

Formal Verification of an Executable LTL Model Checker with Partial  
Order Reduction . . . . . 307  
*Julian Brunner and Peter Lammich*

A Modular Way to Reason About Iteration . . . . . 322  
*Jean-Christophe Filliâtre and Mário Pereira*

A Proof Infrastructure for Binary Programs . . . . . 337  
*Ashlie B. Hocking, Benjamin D. Rodes, John C. Knight,  
Jack W. Davidson, and Clark L. Coleman*

Hierarchical Verification of Quantum Circuits. . . . . 344  
*Sidi Mohamed Beillahi, Mohamed Yousri Mahmoud, and Sofiène Tahar*

**Correctness and Certification**

Semantics for Locking Specifications. . . . . 355  
*Michael D. Ernst, Damiano Macedonio, Massimo Merro,  
and Fausto Spoto*

From Design Contracts to Component Requirements Verification . . . . . 373  
*Jing Liu, John D. Backes, Darren Cofer, and Andrew Gacek*

A Hybrid Architecture for Correct-by-Construction Hybrid Planning  
and Control . . . . . 388  
*Robert P. Goldman, Daniel Bryce, Michael J.S. Pelican,  
David J. Musliner, and Kyungmin Bae*

**Author Index** . . . . . 395