Preface

This publication contains the proceedings of the Sixth NASA Formal Methods Symposium (NFM 2014), which was held April 29 - May 1, 2014 at NASA Johnson Space Center (JSC) in Houston, Texas, USA.

The widespread use and increasing complexity of mission- and safety-critical systems require advanced techniques that address their specification, verification, validation, and certification requirements.

The NASA Formal Methods 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. Within NASA such systems include autonomous robots, separation assurance algorithms for aircraft, Next Generation Air Transportation (NextGen), and autonomous rendezvous and docking for spacecraft. Moreover, emerging paradigms such as property-based design, code generation, and safety cases are bringing with them new challenges and opportunities. The focus of the symposium was on formal techniques, their theory, current capabilities, and limitations, as well as their application to aerospace, robotics, and other safety-critical systems in all design life-cycle stages. We encouraged work on cross-cutting approaches marrying formal verification techniques with advances in safety-critical system development, such as requirements generation, analysis of aerospace operational concepts, and formal methods integrated in early design stages carrying throughout system development.

The NASA Formal Methods Symposium is an annual event that was created to highlight the state of the art in formal methods, both in theory and in practice. The series is a spinoff of the original Langley Formal Methods Workshop (LFM). LFM was held six times in 1990, 1992, 1995, 1997, 2000, and 2008 near NASA Langley in Virginia, USA. In 2009 the first NASA Formal Methods Symposium was organized by NASA Ames Research Center in Moffett Field, CA. In 2010, the Symposium was organized by NASA Langley Research Center and NASA Goddard Space Flight Center, and held at NASA Headquarters in Washington, D.C. The third NFM symposium was organized by the Laboratory for Reliable Software at the NASA Jet Propulsion Laboratory/California Institute of Technology, and held in Pasadena, CA in 2011. NFM returned to NASA Langley Research Center in 2012; the Symposium was organized by the NASA Langley Formal Methods Group in nearby Norfolk, Virginia. NASA Ames Research Center organized and hosted NFM 2013, the fifth Symposium in the series. This year’s Symposium was organized via a collaboration between NASA Goddard Space Flight Center, NASA Johnson Space Center, and NASA Ames Research Center.

The topics covered by NFM 2014 include but are not limited to: model checking; theorem proving; static analysis; model-based development; runtime
monitoring; formal approaches to fault tolerance; applications of formal methods to aerospace systems; formal analysis of cyber-physical systems, including hybrid and embedded systems; formal methods in systems engineering, modeling, requirements, and specifications; requirements generation, specification debugging, formal validation of specifications; use of formal methods in safety cases; use of formal methods in human-machine interaction analysis; formal methods for parallel hardware implementations; use of formal methods in automated software engineering and testing; correct-by-design, design for verification, and property-based design techniques; techniques and algorithms for scaling formal methods, e.g., abstraction and symbolic methods, compositional techniques, parallel and distributed techniques; application of formal methods to emerging technologies.

Two types of papers were considered: regular papers describing fully developed work and complete results, and short papers describing tools, experience reports, or descriptions of work in progress with preliminary results. The Symposium received 107 abstract submissions, 83 of which resulted in full papers: 50 regular papers, and 33 short papers in total. Out of these, 20 regular papers and 9 short papers were accepted, giving an overall acceptance rate of 35% (a 40% rate for regular papers and a 27% rate for short papers). All submissions went through a rigorous reviewing process, where each paper was read by at least three reviewers.

In addition to the refereed papers, the symposium featured three invited talks and a panel feature. NASA provided a special guest talk on “NASA Future Challenges in Formal Methods,” delivered by Bill McAllister, Chief, Safety and Mission Assurance, International Space Station Safety Panels, Avionics and Software Branch, NASA Johnson Space Center. Professor Lawrence C. Paulson from the University of Cambridge gave a keynote talk on “Theorem Proving and the Real Numbers: Overview and Challenges.” Professor Moshe Y. Vardi from Rice University gave a keynote talk on “Compositional Temporal Synthesis.”

The NFM 2014 panel feature, titled “Future Directions of Specifications for Formal Methods,” featured panelists Matt Dwyer of the University of Nebraska, Hadas Kress-Gazit of Cornell University, and Moshe Y. Vardi of Rice University. Specifications are required for all applications of formal methods yet extracting specifications for real-life safety critical systems often proves to be a huge bottleneck or even an insurmountable hurdle to the application of formal methods in practice. This is the state for safety-critical systems today and as these systems grow more complex, more pervasive, and more powerful in the future, there is not a clear path even for maintaining the bleak status quo. Therefore, NFM2014 highlighted this issue in the home of an important critical system, the Mission Control Center of NASA’s most famous critical systems, and asked our panelists where we can go from here.

The organizers are grateful to the authors for submitting their work to NFM 2014 and to the invited speakers and panelists for sharing their insights. NFM 2014 would not have been possible without the collaboration of the Steering
Committee, Program Committee, external reviewers, and the support of the NASA Formal Methods community. We are also grateful to our collaborators at the LERO the Irish Software Engineering Research Centre. The NFM 2014 website can be found at http://www.NASAFormalMethods.org.

February 2014

Julia M. Badger
Kristin Yvonne Rozier
## Program Committee

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domagoj Babic</td>
<td>Google Research, USA</td>
</tr>
<tr>
<td>Calin Belta</td>
<td>Boston University, USA</td>
</tr>
<tr>
<td>Armin Biere</td>
<td>Johannes Kepler University, Austria</td>
</tr>
<tr>
<td>Nikolaj Bjorner</td>
<td>Microsoft Research, USA</td>
</tr>
<tr>
<td>Jonathan P. Bowen</td>
<td>Museophile Limited, UK</td>
</tr>
<tr>
<td>Guillaume Brat</td>
<td>CMU/NASA Ames Research Center, USA</td>
</tr>
<tr>
<td>Gianfranco Ciardo</td>
<td>Iowa State University, USA</td>
</tr>
<tr>
<td>Frederic Dadeau</td>
<td>FEMTO-ST/Inria, France</td>
</tr>
<tr>
<td>Ewen Denney</td>
<td>SGT/NASA Ames Research Center, USA</td>
</tr>
<tr>
<td>Ben Di Vito</td>
<td>NASA Langley Research Center, USA</td>
</tr>
<tr>
<td>James Disbrow</td>
<td>NASA Dryden Flight Research Center, USA</td>
</tr>
<tr>
<td>Steven Drager</td>
<td>Air Force Research Laboratory, USA</td>
</tr>
<tr>
<td>Alexandre Duret-Lutz</td>
<td>LRDE/EPITA, France</td>
</tr>
<tr>
<td>Cindy Eisner</td>
<td>IBM Research - Haifa, Israel</td>
</tr>
<tr>
<td>Éric Férion</td>
<td>Georgia Institute of Technology, USA</td>
</tr>
<tr>
<td>Shalini Ghosh</td>
<td>SRI, USA</td>
</tr>
<tr>
<td>Alwyn Goodloe</td>
<td>NASA Langley Research Center, USA</td>
</tr>
<tr>
<td>Arie Gurfinkel</td>
<td>Software Engineering Institute, Carnegie</td>
</tr>
<tr>
<td></td>
<td>Mellon University, USA</td>
</tr>
<tr>
<td>John Harrison</td>
<td>Intel Corporation, USA</td>
</tr>
<tr>
<td>Klaus Havelund</td>
<td>NASA/Jet Propulsion Laboratory, California</td>
</tr>
<tr>
<td></td>
<td>Institute of Technology, USA</td>
</tr>
<tr>
<td>Connie Heitmeyer</td>
<td>Naval Research Laboratory, USA</td>
</tr>
<tr>
<td>Gerard Holzmann</td>
<td>NASA/Jet Propulsion Laboratory, California</td>
</tr>
<tr>
<td></td>
<td>Institute of Technology, USA</td>
</tr>
<tr>
<td>Hadas Kress-Gazit</td>
<td>Cornell University, USA</td>
</tr>
<tr>
<td>Joe Leslie-Hurd</td>
<td>Intel Corporation, USA</td>
</tr>
<tr>
<td>David Lester</td>
<td>Manchester University, UK</td>
</tr>
<tr>
<td>Kenneth McMillan</td>
<td>Microsoft Research, USA</td>
</tr>
<tr>
<td>Sheena Miller</td>
<td>Barrios Technology/NASA Johnson Space Center, USA</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Steven Miller</td>
<td>Rockwell Collins, USA</td>
</tr>
<tr>
<td>Cesar Munoz</td>
<td>NASA Langley Research Center, USA</td>
</tr>
<tr>
<td>Suzette Person</td>
<td>NASA Langley Research Center, USA</td>
</tr>
<tr>
<td>Lee Pike</td>
<td>Galois, Inc., USA</td>
</tr>
<tr>
<td>André Platzer</td>
<td>Carnegie Mellon University, USA</td>
</tr>
<tr>
<td>Neha Rungta</td>
<td>SGT/NASA Ames Research Center, USA</td>
</tr>
</tbody>
</table>
Johann Schumann  
Cristina Secceleau  
Sandeep Shukla  
Radu Siminiceanu  
Oksana Tkachuk  
Stefano Tonetta  
Helmut Veith  
Arnaud Venet  
Mike Whalen  
Nok Wongpiromsarn  
Karen Yorav

Steering Committee

Ewen Denney  
Ben Di Vito  
Klaus Havelund  
Gerard Holzmann  
Cesar Munoz  
Corina Pasareanu  
Suzette Person  
Kristin Yvonne Rozier

Additional Reviewers

Anderson, Matthew  
Archer, Myla  
Aydin Gol, Ebru  
Bak, Stan  
Bartocci, Ezio  
Breuer, Peter  
Bushnell, David  
Dagit, Jason  
Deng, Yi  
Diatchki, Iavor  
Donzé, Alexandre  
Duggirala, Parasara Sridhar  
Elenius, Daniel  
Faber, Johannes  
Fischer, Bernd  
Fulton, Nathan  
Gario, Marco  
Gascón, Adrià  
Guralnik, Elena  
Hatvani, Leo  
Hendrix, Joe  
Jobredeaux, Romain  
Jovanović, Dejan  
Julliand, Jacques  
Kong, Zhaodan  
Koyfman, Anatoly  
Li, Wenchao  
Mallet, Frederic
Marinescu, Raluca
Mitsch, Stefan
Moran, Shiri
Mover, Sergio
Müller, Andreas
Nanjundappa, Mahesh
Orni, Avigail
Owre, Sam
Pai, Ganesh
Pan, Guoqiang
Pham, Hung
Pidan, Dmitry
Renault, Etienne

Rodriguez-Navas, Guillermo
Rozier, Eric
Saeedloï, Neda
Seidl, Martina
Swei, Sean
Thompson, Sarah
Tiwari, Ashish
Veksler, Tatyana
Völp, Marcus
Wang, Timothy
Whiteside, Iain
Zawadzki, Erik
The introduction of formal methods into a legacy software development process like the International Space Station Programs presents many of the same challenges impeding the wide acceptance of formal methods by industry. Not the least of these being budget and schedule targets and process inertia. While prior work shows that even the highly regarded Shuttle flight software development process could have benefited from the use of formal methods, the approach was never integrated into the baseline process.

Further, NASA software safety and quality assurance engineers face significant challenges of oversight and insight where NASA, rather than acquiring a software product in support of a program, contracts for a program service that happens to require software. For example, the ISS program levies no more than thirty computer based control system (software safety) and ten software quality assurance requirements on the Commercial Resupply Services and the Commercial Crew providers. While the providers compliance data is reviewed and approved by NASA, there is only the slightest opportunity to influence the software development methods employed.

Regardless, any mature development process includes corrective actions to eliminate the recurrence of escapes. Because single corrective actions routinely identify multiple escapes in the development cycle, these investigations provide an opportunity to examine the utility of formal methods.
One of the first achievements in automated theorem proving was Jutting’s construction of the real numbers using AUTOMATH [14]. But for years afterwards, formal proofs focused on problems from functional programming and elementary number theory. In the early 90s, John Harrison revived work on the reals by formalising their construction using HOL [8] and by undertaking an extensive programme of research into verifying floating point arithmetic, including the exponential and trigonometric functions [9–11].

MetiTarski represents a different approach to theorem proving about the reals. Reducing everything to first principles is rigorous, but makes proofs of the simplest statements extremely time-consuming. Many other automatic theorem provers are confined to linear arithmetic, or at best, polynomial comparisons. MetiTarski can prove complicated assertions involving transcendental functions. It takes many of their properties as axioms, and reasons from these properties using sophisticated decision procedures. MetiTarski has recently been integrated with other powerful reasoning tools, including KeYmaera [19] and PVS [17]. With this power, proofs involving such things as aircraft manoeuvres and the stability of hybrid systems can be undertaken, even when the dynamics are described by complicated formulas involving many special functions. Examples of this research can be found in these proceedings, for example, Denman’s work on qualitative abstraction of hybrid systems [6].

This very success raises the question of how to recover the rigour of LCF-style theorem proving without losing the power of MetiTarski. The standard answer to this question (used by Isabelle’s Sledgehammer for example [18]) is for the external prover to generate some sort of certificate that can be checked rigorously. The point is that the expensive proof search does not need to be checked, but only the proof that was actually found.

Checking a certificate using a separate theorem prover, such as Isabelle, requires machine formalisations of all the underlying mathematics. Since Harrison’s work mentioned above, researchers worldwide have formalised substantial chunks of real analysis, including measure theory and probability theory [12,16]. Independently, from the 1960s onwards, computer algebra systems enjoyed rapid development, as did decision procedures for real arithmetic. Much recent work has focused on formalising computer algebra algorithms within theorem provers, especially Coq [2,15]. Investigations into special function inequalities have been conducted using PVS [5].
Nevertheless, the mathematics needed to certify the sort of proofs found by MetiTarski does not appear to have been formalised as yet. MetiTarski relies on an external decision procedure for real-closed fields (RCF) [7] to test the satisfiability of first-order formulas involving polynomials. The underlying algorithm is called CAD (Cylindrical Algebraic Decomposition) and QEPCAD [3] is a well-known implementation, although it has also been implemented in Mathematica and Z3 [13]. Each of these implementations is very complicated, and there is no obvious way to verify their results.

The underlying mathematics is real algebraic geometry [1]. MetiTarski also relies upon upper and lower bounds for the fractions it reasons about, given in the form of truncated power series or rational functions derived from continued fractions [4]. The necessary mathematics here belongs to approximation theory, and unusually, we are not concerned with the closeness of the approximations; the soundness of MetiTarski relies only upon the property that they are indeed upper or lower bounds. Proving these properties formally appears to require a substantial effort. And although we are only concerned with the real numbers, the necessary theory is most easily reached via complex analysis. That branch of mathematics remains largely unformalised at the moment, so we have much to do.

Acknowledgements. The Edinburgh members of the project team are Paul Jackson, Grant Passmore and Andrew Sogokon. The Cambridge team includes James Bridge, William Denman and Zongyan Huang. We are grateful to our outside collaborators such as César Muñoz, Eva Navarro-López, André Platzer, and others not listed here.

The research was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/I011005/1, EP/I010335/1].

References


Synthesis is the automated construction of a system from its specification. In standard temporal-synthesis algorithms, it is assumed the system is constructed from scratch. This, of course, rarely happens in real life. In real life, almost every non-trivial system, either in hardware or in software, relies heavily on using libraries of reusable components. Furthermore, other contexts, such as web-service orchestration and choreography, can also be modeled as synthesis of a system from a library of components.

In this talk we describe and study the problem of compositional temporal synthesis, in which we synthesize systems from libraries of reusable components. We define two notions of composition: data-flow composition, which we show is undecidable, and control-flow composition, which we show is decidable. We then explore a variation of control-flow compositional synthesis, in which we construct reliable systems from libraries of unreliable components.

Acknowledgements. Joint work with Yoad Lustig and Sumit Nain.

References

Panel: Future Directions of Specifications for Formal Methods

Julia M. Badger\textsuperscript{1} and Kristin Yvonne Rozier\textsuperscript{2,⋆}

\textsuperscript{1}NASA Johnson Space Center, Houston, Texas, USA  
Julia.M.Badger@nasa.gov
\textsuperscript{2}NASA Ames Research Center, Moffett Field, California, USA  
Kristin.Y.Rozier@nasa.gov

Specifications are required for all applications of formal methods, yet extracting specifications for real-life safety critical systems often proves to be a huge bottleneck or even an insurmountable hurdle to the application of formal methods in practice. This is the state for safety-critical systems today and as these systems grow more complex, more pervasive, and more powerful in the future, there is not a clear path even for maintaining the bleak status quo. Therefore, we propose highlighting this issue in the home of an important critical system, the Mission Control Center of NASA’s most famous critical systems, and asking our panelists where we can go from here.

Panelists

– Matt Dwyer, University of Nebraska, USA
– Hadas Kress-Gazit, Cornell University, USA
– Moshe Y. Vardi, Rice University, USA

Panel Questions

1. **Where are we now?** Please outline your background and answer the question “Where are we now?” with regards to specifications.

2. **Where will we get specifications from?** At NASA in particular, extracting specifications needed for any formal analysis is a huge challenge. Some critical systems are designed without ever having what this community would consider to be a formal set of requirements. Some design processes don’t formally define requirements until the testing phase, far too late to use them for design or design-time analysis, or other key periods in the system development life-cycle where formal methods are applicable. Even for critical systems where specifications are defined early in the system development life-cycle, they often mix many different objectives, mixing many different levels of detail and describing things like how the system is defined, how the system should behave, legal-speak on why the system satisfies rules, and

⋆ Panel Moderator
more – sometimes all in the same sentence! As safety-critical systems become increasingly complex and the budgetary and other constraints tighten, where can we look in the future to hope to extract the specifications we need for formal analysis?

3. **How should we measure specification quality?** How can we know when we’re “done” extracting specifications or have some idea of how well we’ve done? As critical systems continue to grow in complexity, how will we measure the completeness, coverage, or general quality of a specification or a set of specifications?

4. **How do we best use specifications?** How should formal specifications (both those we are given and those we must extract) fit into the design lifecycle for different kinds of critical systems? How can we indoctrinate formal specifications into diverse teams of system designers without hitting barriers to adoption such as huge costs in terms of time and learning curves? What should our roadmap look like for a future full of well-specified (formally analyzable) critical systems?

5. **We are now open for questions from the audience.**
# Table of Contents

DO-333 Certification Case Studies .................................................. 1  
*Darren Cofer and Steven Miller*

A Compositional Monitoring Framework for Hard Real-Time Systems ........................................ 16  
*André de Matos Pedro, David Pereira, Luís Miguel Pinho, and Jorge Sousa Pinto*

Leadership Election: An Industrial SoS Application of Compositional Deadlock Verification ...................... 31  
*Pedro R.G. Antonino, Marcel Medeiros Oliveira, Augusto C.A. Sampaio, Klaus E. Kristensen, and Jeremy W. Bryans*

Verification of Certifying Computations through AutoCorres and Simpl .............................................. 46  
*Lars Noschinski, Christine Rizkallah, and Kurt Mehlhorn*

Distinguishing Sequences for Partially Specified FSMs ........................................................................... 62  
*Robert M. Hierons and Uraz Cengiz Türker*

On Proving Recoverability of Smart Electrical Grids ................................................................................. 77  
*Seppo Horsmanheimo, Maryam Kamali, Mikko Kolehmainen, Mats Neovius, Luigia Petre, Mauno Rönkkö, and Petter Sandvik*

Providing Early Warnings of Specification Problems .................................................................................. 92  
*Dustin Hoffman, Aditi Tagore, Diego Zaccai, and Bruce W. Weide*

Mechanized, Compositional Verification of Low-Level Code .................................................................... 98  
*Björn Bartels and Nils Jähnig*

Formally Verified Computation of Enclosures of Solutions of Ordinary Differential Equations .................. 113  
*Fabian Immler*

On the Quantum Formalization of Coherent Light in HOL ........................................................................ 128  
*Mohamed Yousri Mahmoud and Sofiène Tahar*

Refinement Types for TLA+ ...................................................................................................................... 143  
*Stephan Merz and Hernán Vanzetto*
Using Lightweight Theorem Proving in an Asynchronous Systems Context ......................................................... 158
  Matthew Danish and Hongwei Xi

JKelloy: A Proof Assistant for Relational Specifications of Java Programs .......................................................... 173
  Aboubakr Achraf El Ghazi, Mattias Ulbrich, Christoph Gladisch, Shmuel Tyszberowicz, and Mana Taghdiri

Verifying Hybrid Systems Involving Transcendental Functions ............. 188
  Paul Jackson, Andrew Sogokon, James Bridge, and Lawrence Paulson

Verifying Nonpolynomial Hybrid Systems by Qualitative Abstraction and Automated Theorem Proving ......................... 203
  William Denman

Combining PVSio with Stateflow .................................................. 209
  Paolo Masci, Yi Zhang, Paul Jones, Patrick Oladimeji, Enrico D’Urso, Cinzia Bernardeschi, Paul Curzon, and Harold Thimbleby

Qed. Computing What Remains to Be Proved ................................. 215
  Loïc Correnson

Warps and Atomics: Beyond Barrier Synchronization in the Verification of GPU Kernels ........................................... 230
  Ethel Bardsley and Alastair F. Donaldson

Testing-Based Compiler Validation for Synchronous Languages .......... 246
  Pierre-Loïc Garoche, Falk Howar, Temesghen Kahsai, and Xavier Thirioux

Automated Testcase Generation for Numerical Support Functions in Embedded Systems ............................................. 252
  Johann Schumann and Stefan-Alexander Schneider

REFINER: Towards Formal Verification of Model Transformations ....... 258
  Anton Wijs and Luc Engelen

Designing a Deadlock-Free Train Scheduler: A Model Checking Approach ................................................................. 264
  Franco Mazzanti, Giorgio Oronzo Spagnolo, and Alessio Ferrari

A Synthesized Algorithm for Interactive Consistency ...................... 270
  Adrià Gascón and Ashish Tiwari

Energy-Utility Quantiles ............................................................. 285
  Christel Baier, Marcus Daum, Clemens Dubslaff, Joachim Klein, and Sascha Klüppelholz
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incremental Verification of Compiler Optimizations</td>
<td>300</td>
</tr>
<tr>
<td>Grigory Fedyukovich, Arie Gurfinkel, and Natasha Sharygina</td>
<td></td>
</tr>
<tr>
<td>Memory Efficient Data Structures for Explicit Verification of Timed Systems</td>
<td>307</td>
</tr>
<tr>
<td>Peter Gjøl Jensen, Kim Guldstrand Larsen, Jiří Srba, Mathias Grund Sørensen, and Jakob Haar Taankvist</td>
<td></td>
</tr>
<tr>
<td>The Gradual Verifier</td>
<td>313</td>
</tr>
<tr>
<td>Stephan Arlt, Cindy Rubio-González, Philipp Rümmer, Martin Schäf, and Natarajan Shankar</td>
<td></td>
</tr>
<tr>
<td>Synthesizing Predicates from Abstract Domain Losses</td>
<td>328</td>
</tr>
<tr>
<td>Bogdan Mihaila and Axel Simon</td>
<td></td>
</tr>
<tr>
<td>Formal Verification of kLIBC with the WP Frama-C Plug-in</td>
<td>343</td>
</tr>
<tr>
<td>Nuno Carvalho, Cristiano da Silva Sousa, Jorge Sousa Pinto, and Aaron Tomb</td>
<td></td>
</tr>
<tr>
<td>Author Index</td>
<td>359</td>
</tr>
</tbody>
</table>