

National Aeronautics and Space Administration
and the University of Minnesota Software Engineering Center present

8th NASA Formal Methods Symposium

June 7–9, 2016, Minneapolis, MN



NFM 2016

JUNE 7-9
MCNAMARA ALUMNI CENTER
UNIVERSITY OF MINNESOTA
MINNEAPOLIS, MN

The widespread use and increasing complexity of mission-critical and safety-critical systems at NASA and the aerospace industry requires advanced techniques that address their specification, design, verification, validation, and certification requirements. The NASA Formal Methods Symposium is a forum for theoreticians and practitioners from academia, industry, and the government, with the goals of identifying challenges and providing solutions towards achieving assurance for such critical systems.

New developments and emerging applications like autonomous on-board software for Unmanned Aerial Systems (UAS), UAS Traffic Management (UTM), advanced separation assurance algorithms for aircraft, and the need for system-wide fault detection, diagnosis, and prognostics provide new challenges for system specification, development, and verification approaches.

The focus of the symposium will be on formal techniques, their theory, current capabilities and limitations, as well as their application to aerospace, robotics, and other safety-critical systems during all stages of the software life-cycle.

ORGANIZING COMMITTEE

Michael Lowry, NASA Ames Research Center

Johann Schumann, SGT, Inc./NASA Ames Research Center

Oksana Tkachuk, SGT, Inc./NASA Ames Research Center

Sanjai Rayadurgam, University of Minnesota

Mike Whalen, University of Minnesota

Mats Heimdahl, University of Minnesota

ORGANIZATION

NFM 2016 is organized by the University of Minnesota Software Engineering Center (UMSEC) in collaboration with NASA Ames Research Center.

NFM 2016 is the eighth edition of the NASA Formal Methods Symposium, steered by the NASA Formal Methods Group. The symposium grew out of a workshop series started by the NASA Langley Formal Methods Group, and is now held annually, hosted each year by one of the NASA centers or a university closely collaborating with NASA.

Modern and majestic, the McNamara Alumni Center is a bold architectural gem on the Minneapolis campus of the University of Minnesota. Its design is inspired by the Minnesota landscape: wood, water and copper.

U of M/Patrick O'Leary



SCHEDULE OF EVENTS: TUE, JUNE 7

8:00-8:15	Check-in/Registration, Welcome, and Opening Remarks	
8:15-9:15	Session: Keynote	Kathleen Fisher
	<i>Using Formal Methods to Eliminate Exploitable Bugs</i>	
9:15-9:45	Break	
9:45-11:15	Session: Requirements and Architectures	Chair: Darren Cofer
9:45-10:15	<i>Temporal Logic Framework for Performance Analysis of Architectures of Systems</i>	Ariane Piel, Jean Bourrely, Stephanie Lala, Sylvain Bertrand and Romain
10:15-10:45	<i>On Implementing Real-time Specification Patterns Using Observers</i>	John Backes, Michael Whalen, Andrew Gacek and John Komp
10:45-11:00	<i>Contract-Based Verification of Complex Time-Dependent Behaviors in Avionic Systems</i>	Devesh Bhatt, Arunabh Chattopadhyay, Wenchao Li, David Oglesby, Sam Owre and Natarajan Shankar
11:00-11:15	<i>ARSENAL: Automatic Requirements Specification Extraction from Natural Language</i>	Shalini Ghosh, Daniel Elenius, Wenchao Li, Patrick Lincoln, Natarajan Shankar and Wilfried Steiner
11:15-12:45	Lunch	
12:45-3:00	Session: Testing and Run-time Enforcement	Chair: Zvonimir Rakamaric
12:45-1:15	<i>Assisted Coverage Closure</i>	Adam Nellis, Pascal Kesseli, Philippa Ryan Conmy, Daniel Kroening, Peter Schrammel and Michael Tautschnig
1:15-1:45	<i>Synthesizing Runtime Enforcer of Safety Properties under Burst Error</i>	Meng Wu, Haibo Zeng and Chao Wang
1:45-2:00	Break	
2:00-2:30	<i>Compositional Runtime Enforcement</i>	Srinivas Pinisetty and Stavros Tripakis
2:30-2:45	<i>Improving an Industrial Test Generation Tool Using SMT Solver</i>	Hao Ren, Devesh Bhatt and Jan Hvozdic
2:45-3:00	<i>The comKorat Tool: Unified Combinatorial and Constraint-based Generation of Structurally Complex Tests</i>	Hua Zhong, Lingming Zhang and Sarfraz Khurshid
3:00-3:15	Break	
3:15-5:00	Session: Posters, Tool Demonstrations, and Reception (University Hall)	

SCHEDULE OF EVENTS: WED, JUNE 8

8:00-8:15	Check-in/Registration	
8:15-9:15	Session: Keynote	Michael L. Aguilar
	<i>Where Formal Methods Might Find Application on Future NASA Missions</i>	
9:15-9:45	Break	
9:45-11:30	Session: Code Generation and Synthesis	Chair: Andrew Gacek
9:45-10:15	<i>Automated Synthesis of Safe Autonomous Vehicle Control Under Perception Uncertainty</i>	Susmit Jha and Vasumathi Raman
10:15-10:45	<i>Obfuscator Synthesis for Privacy and Utility</i>	Yi-Chin Wu, Vasumathi Raman, Stephane Lafortune and Sanjit A. Seshia
10:45-11:15	<i>Code Generation Using a Formal Model of Reference Counting</i>	Gaspard Ferey and Natarajan Shankar
11:15-11:30	<i>EventB2Java: A Code Generator for Event-B</i>	Néstor Cataño and Victor Rivera
11:30-1:00	Lunch	
1:00-3:15	Session: Applications of Formal Methods	Chair: Kristin Yvonne Rozier
1:00-1:30	<i>A Formally Verified Checker of the Safe Distance Traffic Rules for Autonomous Vehicles</i>	Albert Rizaldi, Fabian Immler and Matthias Althoff
1:30-2:00	<i>Probabilistic Formal Verification of the SATS Concept of Operation</i>	Muhammad Usama Sardar, Nida Afaq, Khaza Anuarul Hoque, Taylor T. Johnson and Osman Hasan
2:00-2:15	Break	
2:15-2:45	<i>Formal Translation of IEC 61131-3 Function Block Diagrams to PVS with Nuclear Application</i>	Josh Newell, Linna Pang, David Tremaine, Alan Wassying and Mark Lawford
2:45-3:00	<i>Formal Analysis of Extended Well-Clear Boundaries for Unmanned Aircraft</i>	César Muñoz and Anthony Narkawicz
3:00-3:15	<i>Formal Validation and Verification Framework and Models for Model-Based and Adaptive Control Systems</i>	Sergio Guarro, Umit Ozguner, Tunc Aldemir, Matt Knudson, Arda Kurt, Michael Yau, Mohammad Hejase and Steve Kwon
3:15-3:30	Break	
3:30-5:00	Session: Breakouts	Chair: Michael Lowry
	<i>Connecting the dots between Formal Methods and Future NASA Missions</i>	

SCHEDULE OF EVENTS: THU, JUNE 9

8:00-8:15	Check-in/Registration	
8:15-9:15	Session: Keynote	Kevin Driscoll
	<i>Murphy Was Here</i>	
9:15-9:45	Session: Report-back from Breakouts	
	<i>Connecting the Dots between Formal Methods and Future NASA Missions</i>	Led by Michael Lowry
9:45-10:00	Break	
10:00-11:30	Session: Techniques for Automated Verification	Chair: Temesghen Kahsai
10:00-10:30	<i>Verifying Relative Safety, Accuracy, and Termination for Program Approximations</i>	Shaobo He, Shuvendu Lahiri and Zvonimir Rakamaric
10:30-11:00	<i>Bandwidth and Wavefront Reduction for Static Variable Ordering in Symbolic Reachability Analysis</i>	Jeroen Meijer and Jaco van de Pol
11:00-11:30	<i>Gray-box Learning of Serial Compositions of Mealy Machines</i>	Andreas Abel and Jan Reineke
11:30 -1:00	Lunch	
1:00-3:15	Session: Theorem Proving and Proofs	Chair: César Muñoz
1:00-1:30	<i>Specification and Proof of High-Level Functional Properties of Bit-Level Programs</i>	Clément Fumex, Claire Dross, Jens Gerlach and Claude Marché
1:30-2:00	<i>Formal Verification of an Executable LTL Model Checker with Partial Order Reduction</i>	Julian Brunner and Peter Lammich
2:00-2:15	Break	
2:15-2:45	<i>A Modular Way to Reason About Iteration</i>	Jean-Christophe Filliâtre and Mário Pereira
2:45-3:00	<i>A Proof Infrastructure for Binary Programs</i>	Ashlie B. Hocking, Benjamin D. Rodes, John C. Knight, Jack W. Davidson and Clark L. Coleman
3:00-3:15	<i>Hierarchical Verification of Quantum Circuits</i>	Sidi Mohamed Beillahi, Mohamed Yousri Mahmoud and Sofiene Tahar
3:15-3:30	Break	
3:30-4:45	Session: Correctness and Certification	Chair: Mike Whalen
3:30-4:00	<i>Semantics for Locking Specifications</i>	Michael D. Ernst, Damiano Macedonio, Massimo Merro and Fausto Spoto
4:00-4:30	<i>From Design Contracts to Component Requirements Verification</i>	Jing Janet Liu, John D. Backes, Darren Cofer and Andrew Gacek
4:30-4:45	<i>A Hybrid Architecture for Correct-by-Construction Hybrid Planning and Control</i>	Robert P. Goldman, Daniel Bryce, Michael Pelican, David Musliner and Kyungmin Bae
4:45-5:00	Closing Remarks	

KEYNOTE SPEAKERS

KATHLEEN FISHER | TUE, JUNE 7
Using Formal Methods to Eliminate Exploitable Bugs

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.

BIOGRAPHY

Kathleen Fisher is Professor in the Computer Science Department at Tufts University. Previously, she was a Principal Member of the Technical Staff at AT&T Labs Research, a Consulting Faculty Member in the Computer Science Department at Stanford University, and a program manager at DARPA where she started and managed the HACMS and PPAML programs. Kathleen's research focuses on advancing the theory and practice of programming languages and on applying ideas from the programming language community to the problem of ad hoc data management. The main thrust of her work has been in domain-specific languages to facilitate programming with massive amounts of ad hoc data. Kathleen is an ACM Fellow. She has served as program chair for FOOL, ICFP, CUFP, and OOPSLA and as general chair for ICFP. Kathleen is past Chair of the ACM Special Interest Group in Programming Languages (SIGPLAN), past Co-Chair of CRA's Committee on the Status of Women (CRA-W), a former editor of the Journal of Functional Programming, and an associated editor of TOPLAS. Kathleen is a recipient of SIGPLAN's Distinguished Service Award.

One of the flagship applications for technologies developed under the HACMS program uses advanced technology from the U of M, Rockwell Collins, DARPA, Boeing, Galois Inc., Draper Labs, and Data61 to make unmanned aerial vehicles much more secure.

Photo released by DARPA



MICHAEL L. AGUILAR | WED, JUNE 8

Where Formal Methods Might Find Application on Future NASA Missions

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



During cryogenic testing, the JWST mirrors were subjected to temperatures dipping to 24 Kelvins, permitting engineers to measure in extreme detail how the shape of each mirror changed as it cooled.

NASAMSFC/David Higginbotham/Erinnett Given

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.

BIOGRAPHY

Currently, Mr. Aguilar works at the NASA Langley Research Center (LaRC) as the NASA Technical Fellow in Software Engineering, and the NASA Engineering and Safety Center (NESC) Discipline Expert in Software.

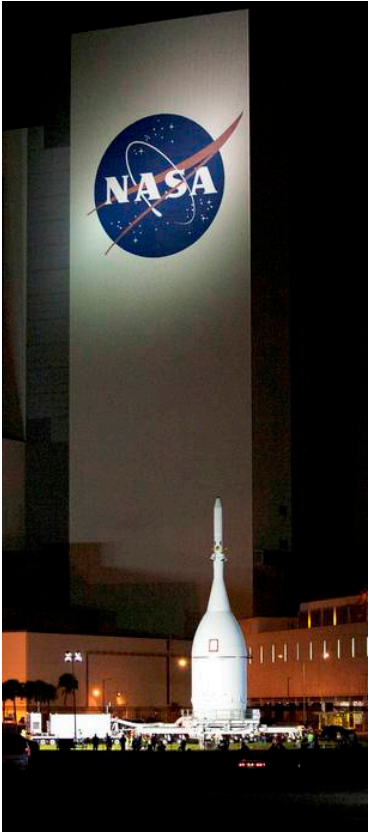
He received his Master’s Degree in Software Engineering from Carnegie Mellon, specializing in real-time systems, and Bachelor’s Degree in Computer Science from CSUN, specializing in robotic automation.

Mr. Aguilar has developed embedded systems his entire career, acting as Project Manager, Configuration Manager, Real-time Performance Engineer, HW API Developer, Test Driver Developer, HW Installation and Integration Engineer, and SW Installation and Integration Engineer. In addition to spacecraft command and control systems, he worked software development on flight simulators, NATO communications systems, RADAR Command and Control

Centers, and Marine Special Forces robotics. As the Safety Engineer for the Interim Control Module (ICM), a reboost and attitude control system for the International Space Station (ISS), Mr. Aguilar performed the analysis and assessment, both in hardware and software, of the ICM quad-processor autonomous Fault, Detection, Isolation, and Recovery (FDIR) capabilities.

After joining NASA in 2003, Mr. Aguilar acted as the James Webb Space Telescope (JWST) Science Instrument Flight Software Manager, managing the interface and integration of the JWST C&DH Core Flight Software development at the Goddard Space Flight Center (GSFC) and the Science Instrument flight software applications developed externally by the European Space Agency, the Canadian Space Agency and EMS Technologies, the University of Arizona and Lockheed Martin ATC, and the JPL/European Consortium.

Mr. Aguilar is currently the NESC Discipline Expert in Software, leading assessments in software architectures, interface modeling, software defects, software static analysis, abort systems, fault tolerance, flight termination systems, software reuse, and flight software code generation. He is currently very involved in the verification of the software being developed for the future NASA crewed missions.



At NASA's Kennedy Space Center in Florida, the agency's Orion spacecraft passes the spaceport's iconic Vehicle Assembly Building as it is transported to Launch Complex 37 at Cape Canaveral Air Force Station.

KEVIN DRISCOLL | THU, JUNE 9

Murphy Was Here

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 transmogrification, “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.”

BIOGRAPHY

Mr. Driscoll is a Honeywell Engineer Fellow with over 45 years of experience in the design of safety-critical and security-critical systems, including the aspects of hardware, software, and systems design. He has nearly 50 patents issued or pending and over 50 papers published in these areas. He was instrumental in creating several network standards, including ARINC 659 SAFEbus, SAE AS4710 PI-bus, and IEEE 1149 JTAG. He led the effort to create the “Handbook for Data Network Evaluation Criteria” for the FAA. Mr. Driscoll created the concept and terminology for “time and space partitioning”. He has been the electronic system architect for space vehicles (e.g., NASA’s Orion Crew Exploration Vehicle), aircraft (e.g., Boeing 777 AIMS), ground and unmanned underwater vehicles (classified). Prior to joining Honeywell, he worked in the areas of voice and data cryptography for the U.S. Army Security Agency and has developed cryptography specifically for real-time systems.

SPONSORED BY THE UNIVERSITY OF MINNESOTA SOFTWARE ENGINEERING CENTER

MASTER OF SCIENCE

UNIVERSITY OF MINNESOTA

Driven to DiscoverSM

SOFTWARE ENGINEERING

UNIVERSITY OF MINNESOTA

- Earn a Master of Science degree in 2 years
- Schedule designed for working professionals:
1 day/week on alternating Fridays and Saturdays
- Offered by the Department of Computer Science and Engineering
- Competitive comprehensive cost includes tuition, fees, textbooks, parking and more
Cost guaranteed for two years



JOIN THE CLASS OF 2018 THIS FALL

msse.umn.edu