THE EIGHTEENTH ANNUAL
HIGH CONFIDENCE SOFTWARE AND SYSTEMS CONFERENCE
Annapolis, MD | May 7-9, 2018
http://cps-hcss.org
The Eighteenth Annual

HIGH CONFIDENCE SOFTWARE AND SYSTEMS CONFERENCE

http://cps-hcss.org

Annapolis, Maryland, USA | May 7-9, 2018
The HCSS steering committee and the NITRD HCSS Coordinating Group are pleased to welcome you to the 18th annual High Confidence Software and Systems (HCSS) Conference being held again this year at the Historic Inns of Annapolis in Annapolis, Maryland.

This year’s program continues a tradition of excellence over the seventeen-year history of the HCSS conference. World class research scientists from academia, industry, and government will deliver a range of experience and technical talks. These presentations will provide new scientific and technological foundations that enable new generations of engineered systems – systems essential for effective life-, safety-, security-, and mission-critical operation. New foundations in science, technology, and advanced practice are especially needed for systems that incorporate the latest advances in computing, communication, information, and control technologies.

This year’s HCSS talks focus on the themes of Complex Heterogenous Systems, Trusting Autonomy, and Blockchain Assurance. These themes and other topics will also be explored through technical posters displayed at a poster session.

We are pleased to host the Software Certification Consortium (SCC) meeting again this year. Formed in 2007, the SCC comprises industry researchers, government regulators, and academicians whose goal is to understand certification issues for systems that contain significant software components (e.g., aerospace, automotive, medical devices, nuclear, defense, etc.), and to objectively make recommendations on processes and standards that impact the certification of such systems.

We hope that you will find the 2018 Conference stimulating and informational. We greatly appreciate your attendance, and look forward to your continued participation and support of future HCSS conferences.

Perry Alexander
University of Kansas

Matthew Wilding
Rockwell Collins
# Table of Contents

1. Welcome Message .................................................. 2
2. Table of Contents .................................................. 3
3. General Information .............................................. 4
4. Conference Organization ........................................... 5
5. Program Agenda ................................................... 6
6. Conference Presentations ........................................ 10
7. Poster Presentations ............................................... 50
8. Conference Dinner ................................................ 60
9. Local Restaurants .................................................. 61
10. Notes ................................................................. 64
GENERAL INFORMATION

REGISTRATION
Registration will be in the state lobby of the Governor Calvert House and will be open:

- 8:00 a.m. to 4:30 p.m. Monday and Tuesday
- 8:00 a.m. to 1:00 p.m. Wednesday

WIRELESS INTERNET CONNECTION
A wireless internet connection will be available in the Governor Calvert Ballroom and Atrium. The network name is: Governor Calvert. The username and password are both “flag”.

POSTER PRESENTATIONS
Poster sessions will be held in the atrium of the Governor Calvert House from 2:15 p.m. to 3:00 p.m. on Monday, May 7 and Tuesday, May 8.

CONFERENCE PRESENTATIONS
Conference presentations and posters will be available online at [http://cps-hcss.org](http://cps-hcss.org).

HOTEL PARKING
Parking at the Historic Inns of Annapolis is by valet only. A reduced parking rate has been negotiated for daily conference attendees, both with in and out privileges. This reduced rate is $12/day. Daily parking for local government attendees is complimentary with approved government ID. Government attendees should visit the registration table each day to have your parking validated or the full fee may apply.

SURVEY
PROGRAM CO-CHAIRS
Perry Alexander, University of Kansas
Matthew Wilding, Rockwell Collins

STEERING COMMITTEE
Kathleen Fisher, Tufts University
John Hatcliff, Kansas State University
Nicole Hughes, National Security Agency
John Launchbury, Galois, Inc.
Stephen Magill, Galois, Inc.
Brad Martin, National Security Agency
Ray Richards, DARPA
Bill Scherlis, Carnegie Mellon University

MEETING ORGANIZERS
Katie Dey, Vanderbilt University
Anne Dyson, Cyber Pack Ventures, Inc.

GRAPHIC DESIGN
Amy Karns, Vanderbilt University

SPONSOR AGENCY
NITRD HCSS Coordinating Group

DONORS
Thank you to our donors for their generous support of the 2018 conference.
PROGRAM AGENDA
<table>
<thead>
<tr>
<th>TIME</th>
<th>TITLE</th>
<th>SPEAKER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0800 - 1630</td>
<td><strong>REGISTRATION</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0900 - 1000</td>
<td><strong>KEYNOTE PRESENTATION</strong></td>
<td>Sandeep Neema (DARPA)</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Assured Autonomy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000 - 1030</td>
<td>Data-driven Safe Control of Nonlinear Systems</td>
<td>Susmit Jha (SRI International)</td>
<td>12</td>
</tr>
<tr>
<td>1030 - 1100</td>
<td><strong>BREAK</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100 - 1130</td>
<td>Using Formal Methods to Reason About Neural Network Based Autonomous Systems</td>
<td>Stephen Magill (Galois, Inc.)</td>
<td>14</td>
</tr>
<tr>
<td>1130 - 1200</td>
<td>Towards Formally Verified Deep Neural Networks</td>
<td>Clark Barrett (Stanford University)</td>
<td>16</td>
</tr>
<tr>
<td>1200 - 1330</td>
<td><strong>LUNCH (ON YOUR OWN)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1330 - 1415</td>
<td>Growing Risk of Cyber Attacks</td>
<td>Vinh Nguyen (ODNI)</td>
<td>18</td>
</tr>
<tr>
<td>1415 - 1500</td>
<td><strong>POSTER SESSION AND BREAK</strong></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>1500 - 1530</td>
<td>Intrusion and Anomaly Detection in Autonomous Vehicle Systems</td>
<td>Sam Lauzon (University of Michigan - Transportation Research Institute)</td>
<td>19</td>
</tr>
<tr>
<td>1530 - 1600</td>
<td>The KeYmaera X Theorem Prover: Hybrid Systems Verification and Verified Runtime Validation</td>
<td>Stefan Mitsch (Carnegie Mellon University)</td>
<td>21</td>
</tr>
<tr>
<td>1600 - 1630</td>
<td>Safe Pointers and Verifiable Automatic Heap Management without Garbage Collection</td>
<td>S. Tucker Taft (AdaCore)</td>
<td>24</td>
</tr>
<tr>
<td>1630</td>
<td><strong>ADJOURN FOR THE DAY</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>TITLE</td>
<td>SPEAKER</td>
<td>PAGE</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------</td>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>0800 - 1630</td>
<td><strong>REGISTRATION</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0900 - 1000</td>
<td><strong>KEYNOTE PRESENTATION</strong></td>
<td>Peter H. Feiler</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>A Four Pillar Improvement Strategy</td>
<td>(Software Engineering Institute)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to Qualification of Embedded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000 - 1030</td>
<td>Perspectives on Applying the Safety</td>
<td>Ewen Denney</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Case Approach for the Assurance</td>
<td>(SGT / NASA Ames)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>of Complex Heterogeneous Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1030 - 1100</td>
<td><strong>BREAK</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100 - 1130</td>
<td>Principles of Layered Attestation</td>
<td>Paul Rowe</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(The MITRE Corporation)</td>
<td></td>
</tr>
<tr>
<td>1130 - 1200</td>
<td>Architecture-Driven Assurance for</td>
<td>Darren Cofer</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Model-Based Systems Engineering</td>
<td>(Rockwell Collins)</td>
<td></td>
</tr>
<tr>
<td>1200 - 1330</td>
<td><strong>LUNCH (ON YOUR OWN)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1330 - 1415</td>
<td>The Future of Cyber-autonomy</td>
<td>David Brumley</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ForAllSecure/CMU)</td>
<td></td>
</tr>
<tr>
<td>1415 - 1500</td>
<td><strong>POSTER SESSION AND BREAK</strong></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>1500 - 1530</td>
<td>Model-based System Certification – A Vision</td>
<td>Jérôme Hugues</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ISAE Supaero)</td>
<td></td>
</tr>
<tr>
<td>1530 - 1600</td>
<td>VeriCores: Cyber-Instrumenting</td>
<td>David S. Hardin</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>Devices Built from Verified</td>
<td>(Rockwell Collins)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1600 - 1630</td>
<td>High-Assurance Blockchains:</td>
<td>Joe Hendrix</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>Applications and Verification</td>
<td>(Galois, Inc)</td>
<td></td>
</tr>
<tr>
<td>1630</td>
<td><strong>ADJOURN FOR THE DAY</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1830</td>
<td><strong>CONFERENCE DINNER</strong></td>
<td>Chart House</td>
<td>60</td>
</tr>
<tr>
<td>TIME</td>
<td>TITLE</td>
<td>SPEAKER</td>
<td>PAGE</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------</td>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>0800 - 1300</td>
<td>REGISTRATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0900 - 1000</td>
<td>KEYNOTE PRESENTATION</td>
<td><strong>Emin Gün Sirer</strong></td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>Challenges Ahead for Blockchains</td>
<td>(Cornell University)</td>
<td></td>
</tr>
<tr>
<td>1000 - 1030</td>
<td>On the Emerging Trend of</td>
<td><strong>Evan Austin</strong></td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Computer-Aided Cryptology and the</td>
<td>(SPAWAR Systems Center Atlantic)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Technology Transition Gap it Presents</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1030 - 1100</td>
<td>BREAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100 - 1145</td>
<td>Applying Symbolic Execution</td>
<td><strong>Mark Mossberg</strong></td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>to Blockchain Applications</td>
<td>(Trail of Bits)</td>
<td></td>
</tr>
<tr>
<td>1145 - 1215</td>
<td>Resilient and Trustworthy</td>
<td><strong>Áron Lászka</strong></td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>Transactive Platform for Smart</td>
<td>(University of Houston)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and Connected Communities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1215 - 1245</td>
<td>Case Study: Verifying Safety</td>
<td><strong>Daniel Genin</strong></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>of a UUV Heading PID Controller</td>
<td>(Johns Hopkins University</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Applied Physics Laboratory)</td>
<td></td>
</tr>
<tr>
<td>1245</td>
<td>CONFERENCE ADJOURNED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CONFERENCE PRESENTATIONS

*bold name denotes presenter*
Abstract:
Tremendous advances have been made in the last decade in constructing autonomy systems, as evidenced by the proliferation of a variety of unmanned vehicles. In spite of these advances, deployment and broader adoption of such systems in safety-critical DoD applications remains challenging and controversial. The Defense Science Board Report on Autonomy, notes that assuring systems operate safely and perform as expected, is integral to trust, especially in a military context. Systems must also be designed so that operators can determine whether, once it has been deployed, it is operating reliably, and, if not, that appropriate action can be taken. The talk will present an overview of DARPA’s Assured Autonomy program, that aims to establish trustworthiness at the design stage and incorporate sufficient capabilities so that inevitable variations in operational trustworthiness can be measured and addressed appropriately. In the course of Assured Autonomy program, researchers will aim to develop tools that provide foundational evidence that a system can satisfy explicitly stated functional and safety goals, resulting in a measure of assurance that can also evolve with the system.

Bio:
Dr. Sandeep Neema joined DARPA in July 2016. His research interests include cyber physical systems, model-based design methodologies, distributed real-time systems, and mobile software technologies.

Prior to joining DARPA, Dr. Neema was a research associate professor of electrical engineering and computer science at Vanderbilt University, and a senior research scientist at the Institute for Software Integrated Systems, also at Vanderbilt University. Dr. Neema participated in numerous DARPA initiatives through his career including the Transformative Apps, Adaptive Vehicle Make, and Model-based Integration of Embedded Systems programs.

Dr. Neema has authored and co-authored more than 100 peer-reviewed conference, journal publications, and book chapters. Dr. Neema holds a doctorate in electrical engineering and computer science from Vanderbilt University, and a master’s in electrical engineering from Utah State University. He earned a bachelor of technology degree in electrical engineering from the Indian Institute of Technology, New Delhi, India.
DATA-DRIVEN SAFE CONTROL OF NONLINEAR SYSTEMS

Susmit Jha

SRI International

Abstract:
Data-driven methods based on machine learning have been adopted to runtime design and optimization of the control of autonomous cyber-physical systems. These methods exploit previous experience of a system to achieve high performance but they lack safety guarantees. Thus, their adoption in safety-critical systems often requires a simplex architecture with a backup model-based safety controller that guarantees the system's safety while data-driven controller optimizes performance within the safety envelope. But the absence of reasonably accurate but yet tractable models poses a major challenge. In this paper, we present a novel data-driven approach for the synthesis of this backup safety controller. Our method requires only an approximate linear model of the system and Lipschitz continuity of the unknown nonlinear dynamics. The nonlinear dynamics is conservatively modeled as a quadratic state-dependent disturbance of a base linear system and this quadratic disturbance is learned from data. The synthesis of the safety controller and the safe set is then performed by restricting the safe set to be elliptical and using a quadratic Lyapunov function as safety certificate. We experimentally demonstrate the effectiveness of the proposed approach.
Bio:

Susmit Jha is a Senior Scientist at SRI International. The focus of his research is in combining formal methods and machine learning to build trusted artificial intelligence and correct-by-construction adaptive systems. He is the Principal Investigator for NSF projects on CPS:Small: Self-Improving Cyber-Physical Systems and EAGER: Duality-Based Algorithm Synthesis. He is a Principal Investigator in the IOBT REIGN CRA with Army Research Lab and leads the task on trusted short time-scale learning for developing reliable IoBT reflexes. Before joining SRI International, he was a Staff Scientist in Embedded Intelligence, Cyber-Physical Systems Group at United Technology Research Center, Berkeley where he performed on Contract-Based Development and Adaptive Deployment for Communication Under Contested Environments (DARPA C2E), Formal Knowledge Modeling and Inference for Aircrew Labor In-Cockpit Automation System (DARPA ALIAS) and Modeling, and Verification of Human-Machine Systems (ONR). Susmit also led projects on Trusted Artificial Intelligence and Automated Mode Switching of Systems at UTRC. Before joining United Technology Research Center, Susmit was a Research Scientist at Intel where he worked on the AI-guided automated synthesis of network-on-chip, adaptive runtime management of platforms and IOT (wireless sensor network) security. Susmit received his Ph.D. in Electrical Engineering and Computer Science from UC Berkeley in 2011 and his B.Tech. in Computer Science and Engineering from Indian Institute of Technology, Kharagpur in 2006. His Ph.D. thesis work on Structurally Constrained Induction and Deduction helped pioneer the area of automatic program synthesis and influenced the development of the FlashFill feature in Excel. His thesis developed a novel synthesis paradigm of oracle-guided inductive synthesis (OGIS) that has been applied for controller synthesis, reverse engineering machine learning models, microarchitecture design, and even automatic discovery of magic card tricks.
USING FORMAL METHODS TO REASON ABOUT NEURAL NETWORK BASED AUTONOMOUS SYSTEMS

Stephen Magill
Galois, Inc.

Abstract:
Neural networks trained via reinforcement learning have shown great promise in increasing system performance and robustness, and are increasingly being incorporated into both software systems and cyber-physical systems. A key question, however, is how to establish a sufficient level of trust in these autonomous systems, particularly for use in mission critical roles. Formal methods based testing has become an important component of verification and validation for non-autonomous systems, but these techniques do not apply directly to systems that include AI components. In this work, we take a step toward systematic and high-coverage testing of autonomous systems by combining recent results on formally describing neural networks with existing techniques for exploring the behavior of traditional software. Our technique combines SMT-based local models of neural network behavior with concolic execution of code to enable system-level reasoning about the behavior of systems that incorporate neural network components.

To demonstrate the approach, we consider the representative problem of using reinforcement learning to train a neural network to play a video game. This basic setup, where a neural network is embedded in traditional discrete logic, and system execution is interleaved with environmental actions, occurs in many applications of machine learning, including AI-based control algorithms and sensor systems for autonomous vehicles.

We focus on two key questions regarding systems involving neural networks: 1) Can we use formal methods to determine system-level inputs that will lead to incorrect behavior? 2) By combining this ability to identify problematic inputs with the reinforcement learning training loop, can we improve learning performance?

We find that the answer is yes to both questions and show that incorporating formal reasoning into training of our example system accelerates convergence of reinforcement learning and enables us to learn controllers for versions of the game rules that traditional reinforcement learning struggles to cope with. From a system analysis perspective, we achieve a capability
that is very similar to what concolic execution provides for non-AI software: we are able to take a set of concrete system test cases and use our technique to generate new test cases that explore novel system behavior, improving our ability to find system errors.

Our technique is based on existing constraint-based encodings of neural networks, but is novel in that we analyze the network together with the rest of the system and then use these results to improve reinforcement learning. In the talk we will 1) describe our techniques and experimental results, 2) present the general theory underlying our work, and 3) expand on the connection between our sample problem and other autonomous systems problems.

Bio:

Dr. Stephen Magill is a Principal Researcher at Galois, where he focuses on applying formal methods to establish safety, security, and privacy. Stephen's work spans domains from security services to cyber-physical systems and embedded control. Over the past four years, he has led teams that have: 1) delivered formal verification technology into the hands of programmers via integration with DevOps, 2) leveraged parallelism to improve security, and 3) developed scalable analyses to measure data leakage and privacy risk. Prior to joining Galois, Stephen was a Research Scientist at the Institute for Defense Analyses Center for Computing Sciences (IDA/CCS) and a post-doctoral researcher at the University of Maryland, College Park. Stephen earned his Ph.D. in Computer Science from Carnegie Mellon University and his B.S. from the University of Tulsa.
TOWARDS FORMALLY VERIFIED DEEP NEURAL NETWORKS

Clark Barrett and Guy Katz
Stanford University

Abstract:

Designing software controllers for autonomous systems is a difficult and error-prone task. On one hand, these systems are required to handle a wide variety of situations and corner cases; but on the other hand, they must meet a very high bar of capability, efficiency, responsiveness and reliability. In recent years, machine-learning (ML) has proven remarkably useful in addressing this challenge, and ML-generated artifacts are expected to become widespread in, e.g., autonomous vehicles. However, this trend has given rise to a new challenge: ML artifacts are, by nature, opaque, and are not amenable to traditional verification and certification. The present inability to automatically reason about these artifacts is likely to severely hinder their applicability to real-world situations.

In this talk we will survey our recent work on verifying deep neural networks, which are a state-of-the-art ML artifacts. Modern neural networks are known to perform well in general, but may present undesirable behavior in some situations. Neural network verification is the systematic exploration of the neural network in order to ensure that certain properties hold for any possible input that the network might encounter when deployed. This problem is computationally difficult (NP-complete); however, by bringing together techniques from the fields of linear programming and satisfiability modulo theories (SMT), we show how it is possible to greatly simplify the problem it in practice. Intuitively, this is done by deferring treatment of the non-linear elements of the problem (which are the main source of complexity) for as long as possible, and only considering them if/when needed. Using an initial implementation of this technique we were able to verify properties of a family of neural networks that are part of the ACAS Xu system: the next-generation airborne collision avoidance system, currently being developed by the FAA. We will also discuss how verification can be applied to ensure that a neural network is robust to small perturbations of their inputs (“adversarial robustness”).

This talk is based on joint work with David Dill, Kyle Julian and Mykel Kochenderfer [1, 2]. This work was supported by Intel and by the FAA.
References:


Bio:

Clark Barrett joined Stanford University in September 2016 after 14 years at NYU. His PhD dissertation (Stanford, 2003) introduced a novel approach to constraint solving called Satisfiability Modulo Theories (SMT). His current work focuses on the application of SMT solvers to improve reliability and security of software, hardware, and machine learning systems. He was also a pioneer in industrial applications of formal hardware verification, contributing to efforts at Intel and 0-in Design Automation. He has received best paper awards from DAC, ITC, and FMCAD, and is an ACM Distinguished Scientist.
GROWING RISK OF CYBER ATTACKS

Vinh Nguyen

National Intelligence Officer for Cyber Issues National Intelligence Council
Office of the Director of National Intelligence

Abstract:
The National Intelligence Officer for Cyber Issues will present the current IC assessment on the threats of disruptive cyber attacks against US critical infrastructure. The presenter will also discuss how the international security and technological development, such as the cyber norms of appropriate state behaviors and the growth of autonomous capabilities through IoT and AI, are increasing this cyber risk in the next five years.

Bio:

Vinh Nguyen leads the Intelligence Community mid- and long-term strategic analysis to support and advance the cyber mission. He serves as the subject matter expert and advises the Director of National Intelligence (DNI) on cyber issues in support of the DNI’s role as the principal intelligence adviser to the President.

Recruited by NSA through the Stokes Program, Mr. Nguyen received the dual degrees in psychology and computer science from the University of Pennsylvania. He earned his MA in International Science and Technology Policy at the George Washington University’s Elliott School of International Affairs, where he focused his work on defense innovation policies and processes. He was trained on positive executive leadership at the University of Michigan’s Ross School of Business.

Mr. Nguyen was responsible for leading and integrating the enterprise- wide cryptologic mission to detect and assess the cybersecurity threats from the Asia- Pacific areas at NSA. He has served in many key analytic and technical positions in the network security, information operations, and counterterrorism areas at NSA. He was instrumental in developing and prototyping cyber attribution methods, leading joint operations against key adversaries, and analyzing terrorist use of the Internet.

Mr. Nguyen was a recipient of the DNI’s Exceptional Accomplishment Award (2017), NSA’s Senior Special Achievement Award (2015), the DNI’s Presidential Daily Briefing Pin (2013), the Meritorious Civilian Service Award (2010), the Sparky Baird Award for the best contributed article published in the SIGNAL magazine (2007), and the Distinguished Young AFCEAN of the Year Award (2006-07). He joined the Defense Intelligence Senior Level ranks in 2014.
INTRUSION AND ANOMALY DETECTION IN AUTONOMOUS VEHICLE SYSTEMS

Sam Lauzon

University of Michigan (Transportation Research Institute)

Abstract:

In today’s automotive research environment, considerable focus has been centered on achieving full and complete automation (i.e. SAE-Level 5), whereby all aspects of driving are completely under autonomous, computer control.

Additional efforts have been put into the research and development of new systems that will be added to complex, heterogeneous networks in order to identify anomalous behavior as a result of compromise or failure in one or more of the connected devices. These systems are being referred to as “IDS”, or Intrusion detection systems.

Currently, there are many types of anomaly and intrusion detection related technologies being investigated and tested. However, it is still not yet fully understood weather or not these technologies will be able to autonomously identify critical issues in such heterogeneous networks. As a result, The University of Michigan Transportation Research Institute (UMTRI) has been contracted by the National Highway Transportation Safety Administration (NHTSA), numerous vehicle manufacturers, and equipment suppliers to develop methods of testing and investigating the efficacy of these systems.

During our investigation over the last two years, we have found none of these systems have been developed to a sufficient degree of accuracy in which all true-positive and false-positive events are conclusively identified.

This research has led to the conclusion that a moderate failure or malicious attack occurring in a vehicle equipped with an intrusion detection system may or may not be positively identified, and thereby, mitigation procedures may not be confidently executed to prevent further compromise of the system. In fact, we have found that when mitigation procedures are deployed, an attacker may be able to use alternate techniques to create an attack vector using the mitigation system itself – having the intrusion detection system mitigate critical safety messages on behalf of the attacker.
These results have shown there may be extreme difficulty in an autonomous, heterogeneous system’s ability to self-identify complex issues pertaining to anomalous or malicious behavior, which leads to the concern relating to the safety and security implications of the use of such vehicles (or related systems) in a possibly hostile environment.

Bio:

**Sam Lauzon** is a Sr. Engineer in Research at the University of Michigan Transportation Research Institute. His experience with vehicle cybersecurity, automotive software, system design, embedded electronics, and expertise in applied cybersecurity - as well as his leadership of the multi-million dollar Department of Homeland Security (DHS) S&T sponsored Uptane (https://uptane.org) project gives him a unique combination of hands-on expertise and an ability to rally the industry while creating a consensus about the implementation and use of new tools to further cybersecurity practice in the industry. He previously held a leading role in the development of automotive infotainment systems, specializing in vehicle interfaces, over-the-air update implementations and low level system support.
THE KEYMAERA X THEOREM PROVER: HYBRID SYSTEMS VERIFICATION AND VERIFIED RUNTIME VALIDATION

Stefan Mitsch, Brandon Bohrer, Nathan Fulton, Yong Kiam Tan, and André Platzer
Carnegie Mellon University

Abstract:
Formal verification plays a crucial role in making cyber-physical systems safe. Formal methods make strong guarantees about the system behavior if accurate models of the system can be obtained. Models including both controller and physical dynamics are essential; but any model we could possibly build, no matter how detailed, necessarily deviates from the real world. If the real system fits to the model, its behavior is guaranteed to satisfy the correctness properties verified with respect to the model. Otherwise, all bets are off. For true safety, we need to combine formal verification of models with methods to ensure that these verification results obtained offline provably apply to implementations at runtime.

Formal Verification in KeYmaera X. KeYmaera X [1] is a theorem prover for hybrid systems, i. e., systems with interacting discrete and continuous dynamics, which arise in virtually all mathematical models of cyber-physical systems.¹ It implements differential dynamic logic (dL [2,3,4]) for hybrid programs, a program notation for hybrid systems. It also provides the necessary foundations for adversarial settings using hybrid games [5]. KeYmaera X was used in several case studies, e.g., for analyzing safety of vertical collision avoidance maneuvers in the next-generation airborne collision avoidance protocol ACAS X [6] and for ground robot collision avoidance [7].

Verification support in KeYmaera X for complex heterogeneous systems features a small-kernel architecture [1] to increase trust in proofs, tactical theorem proving [8] to automate new proofs, adapt existing proofs, and compose proofs about sub-systems to system proofs [9,10].

Verified Runtime Validation in KeYmaera X. Runtime verification and validation methods are lightweight formal methods that monitor a system for violation of properties during the system’s normal operation. With our verified runtime validation method ModelPlex [11,12], implemented in KeYmaera X, we use theorem proving to approach runtime monitoring from a complementary perspective: what are the right properties to monitor in order to ensure safety?
Due to the inevitability of modeling challenges in CPS, this makes ModelPlex, so far, the only technique that can transfer offline guarantees about CPS models to online guarantees about the running CPS.

ModelPlex can help increase trust in autonomy by sandboxing unverified implementations and by monitoring for unexpected behavior in the environment. For example, ModelPlex-generated monitors provided a basis for addressing the safety of reinforcement learning [13]. The underlying idea of ModelPlex is that correct monitoring properties must detect non-compliance between a model and reality based on sensor measurements. Compliance amounts to comparing observations at different times with the predictions of a model. Instead of running models, which can be computationally expensive due to differential equations, a major insight of ModelPlex is that much of the expensive computation can be done offline with a formal proof, resulting in purely arithmetical conditions on the relationship between sensor values, control output, and their history. These conditions, if shown to be true at system runtime from verifiably correct implementations [14], provably witness compliance between the model and reality. The resulting monitor and the observed system execution are, thus, accompanied by a correctness proof.

References:


---

1KeYmaera X is available at http://keymaeraX

Bio:

Stefan Mitsch is a System Scientist in the Logical Systems Lab of Carnegie Mellon’s Computer Science Department. He received his PhD in computer science from Johannes Kepler University in 2012. His research focuses on modeling, formal verification, and verified runtime safety methods for cyber-physical systems. He is particularly interested in applying software engineering methods to support system design and verification.
SAFE POINTERS AND VERIFIABLE AUTOMATIC HEAP MANAGEMENT WITHOUT GARBAGE COLLECTION

S. Tucker Taft
AdaCore

Abstract:
SPARK 2014 is a language intended for formal verification of software. It is based on a large subset of Ada 2012, a language that itself includes standard syntax for specifying pre and postconditions, subtype predicates, and type invariants. SPARK augments Ada 2012 with annotations for specifying global variable usage, dependencies between inputs and outputs, responsibilities for initialization and synchronization, loop variants and invariants, etc. These annotations enable various levels of proof, from simple absence of run-time errors, all the way to full functional correctness.

SPARK 2014 supports most features of Ada 2012, including Ada’s object-oriented features, a subset of Ada’s multitasking features including Ada’s data-oriented synchronization mechanism known as protected types, Ada’s generic templates, etc. One significant feature not currently supported in SPARK 2014 is the ability to create heap-based data structures linked together using pointers (called “access types” in Ada). This is in large part due to the challenges of formally verifying pointer-heavy code and manual heap management. Garbage collection eliminates the complexity of verifying manual heap management, but introduces the complexity of verifying the garbage collector, while introducing real-time and resource limitation concerns. Even without the complexities of verifying the heap management, the "aliasing" inherent in most use of pointers makes proofs that much harder.

Program verification in general depends heavily on having full understanding of any potential "aliasing" between distinct names that might refer to the same underlying memory, so that when an update is performed, appropriate (sound) conclusions can be drawn. SPARK enforces strict non-aliasing between two parameters passed to the same subprogram, or between a parameter and a global variable used by the subprogram, if updating is permitted of either. This means that when inside a subprogram, program proofs need not consider the possibility of aliasing between parameters and/or global variables.
When considering adding pointers to SPARK, we felt it was essential to preserve these restrictions on aliasing. The notion of "pointer ownership," which restricts the number of pointers through which an object can be updated, provides the key ingredient to reduce aliasing, and to simplify heap management.

This presentation will describe the approach taken to adding "owning" pointers and automatic heap management to SPARK, and the proof rules necessary to allow the formal verification of pointer-heavy SPARK programs.

When considering adding pointers to SPARK, we felt it was essential to preserve these restrictions on aliasing. The notion of "pointer ownership," which restricts the number of pointers through which an object can be updated, provides the key ingredient to reduce aliasing, and to simplify heap management.

This presentation will describe the approach taken to adding "owning" pointers and automatic heap management to SPARK, and the proof rules necessary to allow the formal verification of pointer-heavy SPARK programs.

For full information on the SPARK language, see:


**Bio:**

**S. Tucker Taft** is VP and Director of Language Research at AdaCore, and Product Architect for AdaCore's new "QGen" Model-Based Development toolsuite. Tucker led the Ada 9X language design team, culminating in the February 1995 approval of Ada 95 as the first ISO standardized object-oriented programming language. His specialties include programming language design, compiler implementation, high-integrity software, integrated development environments, real-time systems, parallel programming, and model-based development. Since 2001, Tucker has been a member of the ISO Rapporteur Group that developed Ada 2005 and Ada 2012. Most recently Tucker has been designing and implementing the parallel programming language ParaSail, and working on parallel programming extensions for Ada.

Prior to joining AdaCore, Tucker was Founder and CTO of SofCheck, Inc., providing tools and technology to enhance software development quality and productivity. Prior to that Mr. Taft was a Chief Scientist at Intermetrics, Inc. and its follow-ons for 22 years. Tucker received an A.B. Summa Cum Laude degree from Harvard University, where he has more recently taught compiler construction and programming language design.
KEYNOTE PRESENTATION

A FOUR PILLAR IMPROVEMENT STRATEGY TO QUALIFICATION OF EMBEDDED SOFTWARE SYSTEMS

Peter H. Feiler
Software Engineering Institute

Abstract:
This presentation will first summarize challenges in qualifying safety-critical embedded software systems and identify problem areas that contribute to these challenges. This leads to a four pillar strategy for improving the qualification of such systems. The first pillar focuses on improving requirement specification as basis for verification. The second pillar focuses on an architecture centric virtual system integration approach to support incremental verification. The third pillar focuses on analysis of functional and non-functional properties to complement testing. The fourth pillar focuses on automated incremental assurance throughout the development life cycle.

Bio:
Peter Feiler is a 33 year veteran and SEI Fellow at the Software Engineering Institute (SEI). His current research interest is in improving the quality of safety-critical software-reliant systems through architecture-centric virtual system integration and incremental life cycle assurance to reduce rework and qualification costs. Peter Feiler has been the technical lead and main author of the SAE Architecture Analysis & Design Language (AADL) standard. He has a Ph.D. in Computer Science from Carnegie Mellon.
PERSPECTIVES ON APPLYING THE SAFETY CASE APPROACH FOR THE ASSURANCE OF COMPLEX HETEROGENEOUS SYSTEMS

Ewen Denney and Ganesh Pai

SGT / NASA Ames Research Center

Abstract:

The Safety Case approach is being adopted in a number of safety- and mission-critical application domains in the U.S., e.g., medical devices, defense aviation, automotive systems, and, lately, civil aviation. This paradigm refocuses traditional, process-based approaches to assurance on demonstrating explicitly stated safety assurance goals, emphasizing the use of structured rationale, and concrete, product-based evidence as the means to provide justified confidence that systems and software are fit for purpose to safely achieve mission objectives.

In aviation, safety cases are core engineering artifacts that detail the efforts undertaken for safety risk management. They are required as part of the regulatory process that approves access to the National Airspace System (NAS) when conducting flight operations with Unmanned Aircraft Systems (UAS) in certain scenarios, e.g., when using alternative means of compliance to the relevant regulations. UAS operations can be viewed as a complex but loosely-coupled heterogeneous system in which safety is achieved through layers of risk mitigation mechanisms that collectively comprise a safety system, and where the constituent subsystems each contribute to overall safety by providing risk reduction functionality. Thus, the system safety case must provide assurance that individual subsystems do not compromise safety, that each mitigation layer is effective, that the composition/combination of mitigations is also effective, and moreover that subsystem interactions and emergent behavior with potential safety impacts are well managed.

Over the past few years, we have developed safety cases for a number of real UAS operations supporting diverse NASA projects in airborne Earth science, aeronautics, and airspace systems. Some of those safety cases underwent scrutiny by the aviation regulator, the applicable NASA boards for airworthiness and flight safety, flight readiness, and mission readiness, and were successfully approved resulting in operational flight approval for the relevant projects. In the corresponding engineering effort, we leveraged our ongoing research in tools and...
technologies for developing aviation safety cases, through our assurance case automation toolset, AdvoCATE. Our approach *composes* diverse evidence using *structured arguments*—to capture assurance rationale and provide a qualitative safety justification—and a *safety architecture*—which provides a quantitative basis for risk assessment and update. Our tool implements a unified model of safety assurance, integrating hazard analyses, requirements, safety architectures, structured assurance arguments, and heterogeneous evidence. AdvoCATE is being engineered atop formal foundations to provide unique capabilities for: i) automated argument creation and assembly, ii) integration of formal methods into wider assurance arguments, iii) automated pattern instantiation, iv) hierarchical and modular abstraction, and v) queries and views.

In this talk, we elaborate our practical experience and insights gained creating UAS safety cases to enable so-called *beyond visual line of sight* (BVLOS) flight. As an example, we use a recently developed safety case, created within the context of the UAS traffic management (UTM) project—which is being developed to enable safe, low-altitude UAS operations within the NAS. The safety case assembled a diversity of engineering artifacts and analyses as evidence, and modeled the safety architecture to: a) present an overall picture of safety risk, b) elaborate the measures undertaken to ensure operational safety, and c) justify risk reduction. Moreover, structured arguments were used to provide assurance of functional safety for the required surveillance and avoidance capabilities.

**Bio:**

**Dr. Ewen Denney** is a senior computer scientist with SGT, Inc. and the Technical Area Liaison (TAL) for the Robust Software Engineering Group at the NASA Ames Research Center. Within RSE, he currently leads a research group that is seeking to establish a rigorous basis for safety cases, in particular software safety cases, develop a tool to support this, and apply this work to NASA problems. He has worked on automated code generation, formal certification, and safety analysis in the aerospace domain, developing substantial AI-based systems for the automated generation of code for scientific computation, and the certification of autocode. He is the author of more than 70 publications on formal methods, program synthesis, and safety cases, and has chaired several international conferences.
PRINCIPLES OF LAYERED ATTESTATION

*Paul Rowe, **Perry Alexander, †Pete Loscocco, ††Sarah Helble, ††Aaron Pendergrass, and *John D. Ramsdell

*The MITRE Corporation, **University of Kansas, †Department of Defense, ††John Hopkins University Applied Science Laboratory

Abstract:
Layered attestations gather heterogeneous pieces of evidence from different parts of a target system, building a structured case for the trustworthiness of the target for a given interaction. Just as with evidence in criminal cases, the manner in which evidence is gathered, processed, and presented affects the conclusions one is warranted to draw. Of course the attestation mechanisms themselves may form part of the target system. A skeptical appraiser may therefore wish to request evidence of the state of the attestation mechanisms, as well as evidence of a “chain of custody” for measurement data.

This work has three main contributions. First, we identify the logical structure induced by dependencies that exist among heterogeneous system components. We show that by measuring components “bottom up,” any corruption not detected by the attestation must either have occurred recently, or else have affect a component deep in the dependency structure. Second, we demonstrate how to establish a robust chain of custody for the evidence in an optimistic setting in which virtualized TPMs are available. We show that such a chain of custody preserves the “recent or deep” conclusions warranted by a bottom-up order of measurement. Finally, we introduce Attestation Protocol Description Terms (APDTs), a domain-specific language for specifying how to gather and process evidence. We give a formal semantics for the execution of PDTs that allow them to serve as a semantically explicit representation of a layered attestation for the purpose of negotiation between the target and the appraiser. This semantics has been implemented in Maat, a flexible platform for negotiating and performing layered attestations.

Bio:
Dr. Paul D. Rowe is a Lead Cybersecurity Researcher at The MITRE Corporation. His research interests include cryptographic protocol analysis, Trusted Computing, cyber resiliency, and formal methods for modeling and verification. He received his PhD in mathematics from the University of Pennsylvania.
ARCHITECTURE-DRIVEN ASSURANCE FOR MODEL-BASED SYSTEMS ENGINEERING

Darren Cofer
Rockwell Collins

Abstract:
Continued advances in aircraft performance and safety will be based on software-intensive systems with frightening levels of complexity. As the growth in complexity of aircraft systems continues, development costs and schedules have increased dramatically. Systems engineering tools and methods have not sufficiently adapted to the demands of today’s complex systems and as a result aircraft systems are typically late, over budget, and often have reduced capabilities compared to original goals. New model-based systems engineering (MBSE) tools that leverage advanced automation and analysis capabilities are needed to manage this complexity.

This talk describes our Architecture-Driven Assurance approach to MBSE. Our approach is based on comprehensive use of formal methods throughout the development process to ensure that vulnerabilities and design defects are eliminated from critical systems. We have developed integrated tools for architectural modeling, analysis, and synthesis to make this approach practical and effective. It is based on the following foundations:

System Architecture modeled in AADL - We create analyzable models of the system architecture to support “virtual integration” throughout the development and sustainment life cycle.

Architecture model is correct - We apply a variety of tools to verify that the architecture implements system-level requirements. This includes: compositional reasoning about system behavior based on formal contracts added to the AADL model; a logic and supporting tool for generating assurance cases from the AADL model structure; model-based safety analysis to evaluate behavior in the presence of faults; and simulation and automatic test case generation for validation of the model and, eventually, the implementation.

Software components are correct - Component contracts are checked for consistency and realizability. Component contracts are exported to component development environments (e.g., Simulink) for implementation and verification.
System does what the model says - We target the formally verified kernel (seL4) to guarantee isolation between components, and to ensure that there are no information flows other than those explicitly defined in the architecture.

Software implementation corresponds to model - Our Trusted Build process automatically generates implementation code from architecture model, component specifications, and kernel/OS build system.

These tools and methods are being developed and refined through a combination of technology development projects and application programs. Technology development programs create or enhance our MBSE capabilities while technology application programs demonstrate the practicality and effectiveness of the approach. In several ongoing programs we are applying the Architecture-Driven Assurance approach to next generation rotorcraft, including Future Vertical Lift (FVL).

**Bio:**

**Darren Cofer** is a Fellow in the Rockwell Collins Advanced Technology Center. He earned his PhD in Electrical and Computer Engineering from The University of Texas at Austin. His principal area of expertise is developing and applying advanced analysis methods and tools for verification and certification of high-integrity avionics systems. His background includes work with formal methods for system and software analysis, the design of real-time embedded systems for safety-critical applications, and the development of nuclear propulsion systems in the U.S. Navy. He has served as the principal investigator on numerous sponsored research projects for NASA, AFRL, and DARPA.
THE FUTURE OF CYBER-AUTONOMY

David Brumley
Carnegie Mellon University

Abstract:
My vision is to automatically check and defend the world's software from exploitable bugs. In order to achieve this vision, I am building technology that shifts the attack/defend game away from the current manual approaches for finding and fixing software security vulnerabilities to fully autonomous cyber reasoning systems.

In this talk, I will describe the DARPA Cyber Grand Challenge, the first effort to create a fully autonomous cyber security system. I'll focus on Mayhem, built by ForAllSecure. Mayhem can find new vulnerabilities, generate exploits, and self-heal off-the-shelf software. Mayhem is the result of 10 years of academic research and 2 years of commercial development. Mayhem competed and won a $2 million dollar prize in the US Cyber Grand Challenge competition co-hosted at DEFCON 2016. I will describe how Mayhem works, the Cyber Grand Challenge competition, and how Mayhem fared against the world's best hacking teams. I will also describe how Mayhem, and other autonomous systems like it, will change the security landscape in the next decade.

Bio:
David Brumley is the CEO and co-founder of ForAllSecure, and a Professor at Carnegie Mellon University in ECE and CS. ForAllSecure's mission is to make the world's software safe, and they develop automated techniques to find and repair exploitable bugs to make this happen. Prof. Brumley previously was the Director of CyLab, the CMU Security and Privacy Institute.

Brumley's honors include a United States Presidential Early Career Award for Scientists and Engineers (PECASE) from President Obama, a 2013 Sloan Foundation award and numerous best paper awards. Prof. Brumley is also advisor and a founding member of PPP, one of the world's most elite competitive hacking teams.
MODEL-BASED SYSTEM CERTIFICATION – A VISION

*Jérôme Hugues and **Eric Feron

*ISAE Supaero, **Georgia Tech

Abstract:

The engineering of safety-critical systems is a triad made of engineers building safer and more resilient systems, end-users asking for more functions and autonomy, and regulatory organizations ensuring a sufficient level of safety is achieved when operating the system. The latter is often perceived as a drag, or even a show-stopper, asking for evidence of safety in the form of document-based plans. In the last few years, Model-Based techniques have become recognized solutions to manage complexity, and to tackle multiple analysis and optimization problems.

Recently, all major standards pertaining to safety-critical systems or sub-systems have been updated to consider model-based techniques, such as RTCA’s DO-178C/DO-331 for avionics software, SAE’s ARP4761A for avionics systems, and ISO 26262 for automotive systems. As a consequence, the community is now equipped with model-based toolsets to perform the engineering of systems (e.g. SysML, STAMP, AADL), to implement their functions (SCADE, Simulink, etc.) or to conduct analyses (e.g. AltaRica for safety). These are also combined with code generation capabilities. Usually, each toolset adheres to one (sub)set of standards, and are neither open nor interoperable outside of relatively small communities.

Even though these standards are tightly coupled to cover the whole systems engineering process, we note they cooperate only through the exchange of documents that interconnect the results of model-based tool-produced steps. These documents are later passed to authorities for reviews. Hence, we perform Model-Based Systems/Software Engineering (MBSE), Model-Based Safety Assessment (MBSA), but critically miss a Model-Based System Certification approach that would seamlessly combine models and analysis artefacts to consolidate the certification artefacts.

The authors/speakers have been involved in research and educational activities to define and spread better engineering practice. In this talk, we report on current efforts to ease the production of certification credits and manage the complexity of the system, not to mention its cost. We will illustrate our demonstration with past examples from the control/command, real-time embedded domains, applied to space and avionics systems.
Our effort is driven by the following considerations: certification can be greatly improved provided it is equipped with a Model-Based process that

- traces requirements to implementation artefacts, but more importantly to mathematical proofs supporting claims of conformance;
- combines analysis results from heterogeneous models or their produced artefacts, e.g. from analysis, simulation, generated code, and execution on test bench; and propagate them to the corresponding stakeholders;
- enables autocoding of functions that are supported by basic blocks that are formally proved (e.g. optimization, linear operators, ...) that are amenable to theorem proving;
- enables autocoding from architectural descriptions that are correct-by-construction, e.g. connected to dimensioning considerations (for schedulability, energy, cyber-security, ...) but also correctly implemented on top of avionics platform.

Here, Model-Based is contemplated mostly as a computer-aided workbench, as the process would first and foremost enable the designers to concentrate on its systems as a valid combination of mathematically sound artefacts, not as a collection of documents.

Bio:

Jérôme Hugues is associate professor at the Department of Complex Systems Engineering of the Institute for Space and Aeronautics Engineering (ISAE) in Toulouse. He holds an HDR (2017), a PhD (2005) and engineering degree (2002) from the Telecom ParisTech. He is responsible of several distributed and real-time systems courses. He is the head of the "Embedded Systems" Master Program at ISAE, and responsible for the management of the Embedded Systems Engineering curriculum.

His research interests are focused on software architecture to support the design of safety-critical software-based real-time and embedded systems; and programming languages and artifacts to support them.

He is also a member of the SAE AS-2C committee working on the definition of the AADL architecture description language to assist the designer in various stages of its design: formal verification, dimensioning down to code generation. He is the main author of two annexes document for AADLv2, and a reviewer of the AADLv2 core document, and associated annexes. He leads the Ocarina project, an AADL model processor.
VERICORES: CYBER-INSTRUMENTING DEVICES BUILT FROM VERIFIED COMPONENTS

*David S. Hardin, *Konrad Slind, and **Perry Alexander

*Rockwell Collins, **University of Kansas

Abstract:
Experience on cyber-assurance research and development programs such as DARPA HACMS has taught us that the safety and security of systems constructed using legacy components can be significantly improved if low-overhead monitoring/attestation/cross-domain elements are introduced at critical interfaces. Such elements establish that system components achieve acceptable performance and isolate bad behavior. These subsystems must be designed to a high level of assurance and operate in a trustworthy fashion, lest their introduction increases, rather than decreases, the system's attack surface.

We are currently developing VeriCores, standardized cyber-instrumenting devices for high-assurance systems built using verified components. A VeriCore is intended to minimize "consumables", such as Size, Weight, Power, and Cost (SWaP-C). This is both a security, as well as an engineering concern, as a "bad actor" can't attack a resource that isn't present. In some designs, a VeriCore may be instantiated as a "bump-in-the-wire", drawing its power from the monitored interface, acting as a pass-through for valid transactions over that interface, and establishing execution characteristics for appraisal. In the case where a high-assurance separation kernel is already present, the VeriCore logic may be implemented solely in software, and execute in a partition of that operating system.

The basic software stack for a VeriCore consists of the seL4 verified microkernel, verified drivers for supported I/O interfaces, a verified networking stack for seL4 currently in development by Kestrel Institute under DARPA funding, and a minimal verified runtime and measurement capability derived from the CakeML verified ML compiler. The VeriCore tool suite includes tools supporting verified development and synthesis of application-specific components. The goal of VeriCore development is to produce verified applications to execute on the verified runtime environment, employing Verification-Enhanced Development principles.
Once realized, VeriCore instances can be "dropped into" critical systems designs as needed, wherever critical, provably correct basic input validation, lexing, parsing, attestation protocol handling, data transformation, cross-domain filtering, majority voting, basic secure hash, etc. functions are required in order to meet high-assurance design requirements.

In order to provide efficient implementations of high-level data structures with the high assurance needed for VeriCore deployment, we have developed a verifying compiler approach that supports the "natural" functional proof style, but yet applies to more efficient data structure implementations. Algorithms are expressed in a system-level language called DASL, employing a mostly-functional style (similar to modern programming languages, such as Swift and Rust), with additional annotations indicating maximum array sizes, etc. We base the Intermediate Verification Language for the DASL toolchain upon higher-order logic; thus, claims about program behavior, as well as source-to-source transformations of the intermediate form, can be mathematically proven. On that basis, proofs about high-level programs over high-level data structures can be carried out while a formal, proved connection to the low-level efficient implementation is automatically ensured.

DASL compiler output is source code for a choice of conventional programming languages, such as Ada, Java, VHDL, C or ML. This source is then further compiled to machine code or an FPGA bit file using conventional tools and a small set of verified runtime functions. Since we compile to an array-based implementation, supporting hardware implementations is particularly straightforward. If the user selects ML code generation, we are able to offer a complete verified toolchain and verified execution environment down to the machine code level, using technologies from the CakeML project. CakeML provides mechanisms to convert HOL4 logic formulas to a subset of Standard ML, as well as a verified compiler from that ML subset to machine code for a number of common Instruction Set Architectures. We thus can readily perform a verified translation of a DASL program to machine code, and execute that verified machine code on VeriCore hardware.

In this talk we will describe the VeriCore infrastructure, detail VeriCore application development, and discuss various applications of the technology to large, heterogeneous systems as well as autonomous systems. We will conclude with a demonstration of the VeriCore technology on representative hardware.

**Bios:**

**Dr. David S. Hardin** has made contributions in the areas of formal methods, computer architecture for High Assurance systems, as well as real-time and embedded Java. He is a Principal Research Engineer in the Advanced Technology Center at Rockwell Collins, and currently serves as a co-Principal Investigator for the DARPA CASE (Cyber Assured Systems Engineering) program. He is the editor of the book Design and Verification of Microprocessor Systems for High-Assurance Applications (Springer 2010), and a co-author of The Real-Time Specification for Java, as well as the Java 2 Micro Edition Connected Device Configuration/Foundation Profile
standards. He is author or co-author of more than 60 peer-reviewed publications, and is a co-inventor on 12 U.S. patents. In 1999, Dr. Hardin co-founded aJile Systems, a startup company focused on real-time and embedded Java technology, and served as aJile’s Chief Technical Officer from 1999 to 2003. Dr. Hardin was selected as a Rockwell Engineer of the Year for 1997 for his contributions to the development of the world’s first Java microprocessor. His academic career includes BSEE and MSEE degrees from the University of Kentucky, and a Ph.D. in Electrical and Computer Engineering from Kansas State University, in 1989. Dr. Hardin is a proud native of the Commonwealth of Kentucky, and is a Kentucky Colonel.

Dr. Konrad Slind has been a member of the Rockwell Collins Advanced Technology Center since 2009. He previously worked at Cambridge University Computer Lab (post-doc), and University of Utah School of Computing (faculty). His research focuses on the theory and application of formal methods. He has published in a variety of areas: implementation of higher order logic theorem provers, verifying compilation, specification of multiprocessor memory models, verification of cryptographic block ciphers, verification of functional programs, and introduction of formal methods into certification processes. Recent work has explored formal approaches to fast verified network guards. Slind’s current research is on runtime systems for graph-based algorithms intended to execute in autonomous vehicles.

Dr. Perry Alexander is the AT&T Distinguished Professor of Electrical Engineering and Computer Science and Director of the Information and Telecommunication Technology Center at The University of Kansas. His research interests include system-level modeling, design languages, heterogeneous specification, language semantics, and trusted computing. He received the BSEE and BSCS in 1986, the MSEE in 1988, and the PhD in 1992 all from The University of Kansas. From September 1992 through July 1999 he was a faculty member and Director of The Knowledge-Based Software Engineering Laboratory in the Electrical and Computer Engineering and Computer Science department at The University of Cincinnati. Dr. Alexander has been involved in numerous projects funded by DARPA, AFRL, NSF, NASA, NavAir, Battelle, and US Department of Defense. He currently leads the ACHILLES and Armored Software efforts at ITTC. He is the chief architect of the Rosetta system specification language and is author of System-Level Design using Rosetta. Dr. Alexander has published over 100 refereed research papers. He has won 22 teaching awards and was named a Kemper Teaching Fellow and the ASEE’s Midwest Region Teacher of the Year in 2003, and received the Sharp Teaching Professorship in 2009. He is a member of Sigma Xi and a Senior Member of ACM and IEEE.
HIGH-ASSURANCE BLOCKCHAINS: APPLICATIONS AND VERIFICATION

Joe Hendrix and Dave Archer

Galois, Inc

Abstract:

In this presentation, we will describe work at Galois aimed at providing and maintaining data integrity guarantees via blockchains even when systems are compromised, and verifying a blockchain implementation satisfies security and functionality requirements. Blockchain developers have shown an interest in extending blockchains to be useful in context outside crypto-currencies. Making this transition requires both finding suitable use cases that address needs, and in ensuring the proposed blockchain technology can actually deliver on its promises.

In the talk, we will focus on two use cases that we can demonstrate: (1) A user-level filesystem that used a blockchain and Intel SGX to provide tamper resistant event logging; (2) a secure messaging application that records and allows verification of each step of a high throughput message delivery protocol. The latter demonstration takes as input telemetry from a UAV flight simulator and distributes that data through a decentralized messaging middleware to multiple analytic packages that graphically display flight data. Those analytics packages then verify each transport protocol step as the data is displayed.

In our verification efforts, we have focused on an existing blockchain system, namely GuardTime KSI blockchain system, which maintains a hash-based ledger to provide secure timestamping and providence guarantees with a fast transaction time. We used the applications above to derive security requirements on the system, and then used automated reasoning tools to show selected aspects of those requirements.

In particular, we formalized a model of the core Guardtime signing service in the Lean interactive theorem prover. We then proved the temporal property that under specific configuration limits and assumptions of network delivery the system would respond to each signature request with a certificate. From a formal methods standpoint, the work made heavy use of Lean’s reflection capabilities to automate proofs within an interactive theorem proving environment even though the proofs were fundamentally about model-checking of infinite state systems.
The main goals of this talk are to provide an overview of the main components in a blockchain system, the guarantees they should provide, and how different techniques and tools developed by the formal methods community can ensure a given system satisfies its requirements. In addition to our work, we would also like to discuss areas where we feel there are gaps in FM tools and cryptographic proofs, and propose ways in which the two communities can work together to address those gaps. In particular, we think there is additional work to be done in model-code correspondence so that proofs about the model can be shown to be true about the implementation, and in developing tools to extend cryptographic proofs, as found in an academic cryptographic paper, to proofs about the underlying distributed network semantics of an implementation.

Additional contributors to our efforts include Ledah Casburn, Joey Dodds, Tom DuBuisson, Simon Hudon, Ben Jones, Alex Malozemoff, and Ben Sherman.

Bio:

**Dr. Joe Hendrix** is a Principal Researcher at Galois. His research interests are in the application of automated reasoning techniques to hard computer science and engineering problems. While at Galois, he has contributed to Galois’ SAW Verification tools, the Crucible symbolic simulator, and the Macaw binary analysis library.
KEYNOTE PRESENTATION

CHALLENGES AHEAD FOR BLOCKCHAINS

Emin Gün Sirer

Cornell University

Abstract:
While there is much hype surrounding blockchain technologies, they offer undeniable scientific advantages compared to the state of the art, and remain poised to disrupt finance if they can overcome the technical hurdles they face. This talk will provide an introduction to the technology and outline the challenges ahead, namely, performance, scalability and correctness. The talk will describe how the unique adversarial environment and architecture of these systems require a re-thinking of the way we approach these problems.

Bio:
Emin Gün Sirer is an associate professor at Cornell University and co-director of the Initiative for Cryptocurrencies and Smart Contracts (IC3). His current research focuses on advancing the science and engineering of blockchain technologies. He has co-discovered how to beat Bitcoin, as well as proposing many protocol improvements to improve its performance and scale. He has also been prescient in calling many attacks and system failures before they happened. He has been accused of being the DAO hacker as well as Satoshi Nakamoto, though he’s neither.
ON THE EMERGING TREND OF COMPUTER-AIDED CRYPTOLOGY AND THE TECHNOLOGY TRANSITION GAP IT PRESENTS

Evan Austin
SPAWAR Systems Center Atlantic

Abstract:
Modern cryptography has greatly benefited from calls for open standardization. The push for "one size fits all" solutions like the Advanced Encryption Standard (AES) and the Secure Hash Algorithms family has promoted interoperability and allowed researchers to focus on improving the security and performance of the most widely adopted cryptosystems, maximizing the impact of their efforts. However, the looming quantum computing threat and desires to utilize cryptography in previously unidentified domains and applications have precipitated a shift towards alternative cryptographic primitives. To date, these new algorithms and parameter sets have not received the same level of scrutiny and review that a system like AES has enjoyed over its last twenty years.

There is a shared belief that the lack of analysis, especially in the form of formal verification, is due to the increasingly complex constructions that new cryptoschemes are founded on. Borrowing the words of Halevi, Bellare, and Rogaway - "[cryptographers] generate more proofs than [they] carefully verify" because "many proofs in cryptography have become essentially unverifiable". Inspired by the more general software and hardware verification communities, cryptologists have leveraged advances in formal methods and language theory research in the development of tools and techniques to address this issue giving rise to the comparatively young field of computer-aided cryptology.

The purpose of this talk is to communicate our experiences investigating and working with a subset of cryptologic reasoning tools. The specific focus will be on the challenges we faced trying to transition research software to a more operationally focused environment staffed with subject matter, but not necessarily formal methods and programming language, experts. We will also highlight areas where we think the application of these tools would be of immediate benefit, including in blockchain assurance, one of this year's highlighted themes of the High Confidence Software and Systems Conference.
References:


Bio:
Dr. Evan Austin is a scientist in the Research and Applied Science competency of Space and Naval Warfare (SPAWAR) Systems Center Atlantic in Charleston, South Carolina. Since joining SPAWAR Systems Center Atlantic in 2015, Dr. Austin has served as the principal investigator on a number of internally and externally funded basic and applied research projects with the shared theme of high-assurance software and systems development. He also serves as the technical lead for SPAWAR Systems Center Atlantic's recently formed Automated Cryptology research and development team and as a member of their Science and Technology Advisory Council.

Dr. Austin holds Bachelor of Science, Master of Science, and Doctor of Philosophy degrees in computer science, all from the University of Kansas. His primary research interests are applied formal methods and programming language theory.
APPLYING SYMBOLIC EXECUTION TO BLOCKCHAIN APPLICATIONS

Mark Mossberg
Trail of Bits

Abstract:
Ethereum is a novel, decentralized computation platform that has quickly risen in popularity since it was introduced in 2014, and currently controls the equivalent of one hundred ten billion dollars. At its foundation is a virtual machine which executes "smart contracts": programs that ultimately control the majority of the value transfer within the network. As with most other types of programs, correctness is very important for smart contracts. However, somewhat uniquely to Ethereum, incorrectness can have a direct financial cost, as evidenced by a variety of high profile attacks involving the loss of hundreds of millions of dollars. The error-prone nature of developing smart contracts and the increasing amounts of capital processed by them motivates the development of analysis tools to assist in automated error and vulnerability discovery.

In this talk, we describe our work towards smart contract analysis tooling for Ethereum, which focuses on a modern technique called symbolic execution. We provide context around both Ethereum and symbolic execution, and then discuss the unique technical challenges involved with combining the two, touching on topics including blockchains, constraint solvers, and virtual machine internals. Lastly, we present Manticore: an open source symbolic execution tool which we have used to enhance smart contract security audits

Bio:
Mark Mossberg is a researcher and engineer at Trail of Bits, and the lead developer for the Manticore project. His interests include automatic binary analysis, systems software development, and interface design.
RESILIENT AND TRUSTWORTHY TRANSACTIVE PLATFORM FOR SMART AND CONNECTED COMMUNITIES

*Áron Lászka, **Anastasia Mavridou, and **Abhishek Dubey
*University of Houston, **Vanderbilt University

Abstract:
Internet of Things, Cyber-Physical Systems, and Data Sciences are fueling the development of innovative solutions for various applications in Smart and Connected Communities (SCC). These solutions are often data-driven, which makes them vulnerable to data integrity attacks. Additionally, the increasing dependence on dynamic data-driven support systems implies that any operational breach in the underlying SCC networks or services due to attacks or failures will have cascading effects. Lastly, such attacks will not only cripple the SCC operations but also escalate customers’ privacy concerns, influencing the extent to which they are willing to share data.

These challenges have led to increasing focus on SCC platforms that provide participants the capability to not only exchange data and services in a decentralized and perhaps anonymous manner, but also provide them with the capability to preserve an immutable and auditable record of all transactions in the system. Such transactive platforms are actively being suggested for use in Healthcare, Smart Energy Systems, and Smart Transportation Systems. These platforms can provide support for privacy-preserving and anonymizing techniques, such as differential privacy, fully homomorphic encryption, and mixing [2,3]. Further, the immutable nature of records and event chronology in these platforms provides high rigor and auditability. Lastly, the decentralized nature of these platforms ensures that any adversary needs to compromise a large number of node to take control of the system.

Blockchains form a key component of these platforms because they enable participants to reach a consensus on any state variable in the system, without relying on a trusted third party or trusting each other. Distributed consensus not only solves the trust issue, but also provides fault-tolerance since consensus is always reached on the correct state as long as the number
of faulty nodes is below a threshold. Further, blockchains also enable performing computation in a distributed and trustworthy manner in the form of smart contracts. However, while the distributed integrity of a blockchain ledger presents unique opportunities, it also introduces new assurance challenges that must be addressed before protocols and implementations can live up to their potential. For instance, smart contracts deployed in practice are riddled with bugs and security vulnerabilities. A recent automated analysis of 19,336 smart contracts deployed in practice found that 8,333 of them suffered from at least one security issue. Although this study was based on smart contracts deployed on the public Ethereum blockchain, the analyzed security issues were largely platform agnostic. Security vulnerabilities in smart contracts present a serious issue for two main reasons. Firstly, smart-contract bugs cannot be patched. By design, once a contract is deployed, its functionality cannot be altered even by its creator. Secondly, once a faulty or malicious transaction is recorded, it cannot be removed from the blockchain (“code is law” principle). The only way to roll back a transaction is by performing a hard fork of the blockchain, which requires consensus among the stakeholders and undermines the trustworthiness of the platform.

In practice, these vulnerabilities often arise due to the semantic gap between the assumptions that contract writers make about the underlying execution semantics and the actual semantics of smart contracts. To tackle these security risks and vulnerabilities, we introduce a formal, finite-state machine (FSM) based language for designing smart contacts, and we provide a tool [4] for generating code from the high-level, graphical, and FSM-based language to low-level smart contract code (Solidity [1]). The advantages of our approach, which helps developers to create secure contracts rather than to fix existing ones, are threefold. First, we provide a formal model with clear semantics and an easy-to-use graphical editor, thereby decreasing the semantic gap and eliminating the issues arising from it. Second, rooting the whole process in rigorous semantics allows the integration of formal analysis tools, which can be used to verify safety and security properties, thereby enabling the development of correct-by-design smart contracts. Third, we model and analyze the interactions between the decentralized smart contract and external IoT actors. Specifically, we provide and enforce rigorous interaction semantics that enables us to provide end-to-end assurance guarantees on the whole system.


Bios:

Áron Lászka is an Assistant Professor in the Department of Computer Science at the University of Houston. His research interests broadly revolve around the security and resilience of cyber-physical systems and Internet of Things, the economics of cyber-security, and game-theoretic modeling of security problems. Previously, he was a Research Assistant Professor at Vanderbilt University from 2016 to 2017, and a Postdoctoral Scholar at the University of California, Berkeley from 2015 to 2016. He graduated summa cum laude with a Ph.D. in Computer Science from the Budapest University of Technology and Economics in 2014. In 2013, he was a Visiting Research Scholar at Pennsylvania State University.

Anastasia Mavridou is a Postdoc at the Institute for Software Integrated Systems, Vanderbilt University, where she works with Prof. Janos Sztipanovits. Her research interests lie in the area of component-based design, modeling and analysis of concurrent systems with a focus on correct-by-construction techniques. She received her PhD in Computer Science from Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland in 2016 under the supervision of Prof. Joseph Sifakis and Dr. Simon Bliudze and her MSc in Computer Science from University of Tulsa, USA in 2012.

Abhishek Dubey is a Senior Research Scientist at the Institute for Software Integrated Systems and an Assistant Professor in the Computer Science and Computer Engineering department at Vanderbilt University. His research interests include tools, platforms and analytical techniques required for dynamic and resilient cyber-physical platforms. Abhishek completed his PhD in Electrical Engineering from Vanderbilt University in 2009. He received his M.S. in Electrical Engineering from Vanderbilt University in August 2005 and completed his undergraduate studies in electrical engineering from the Indian Institute of Technology, Banaras Hindu University, India in May 2001. He is a senior member of IEEE.
CASE STUDY: VERIFYING SAFETY OF A UUV HEADING PID CONTROLLER

Daniel Genin, Durward McDonnell, Raymond McDowell, Ian Blumenfeld, Adam W. Cushman, and Jay Guthrie

Johns Hopkins University Applied Physics Laboratory

Abstract:

Proportional Integral Derivative (PID) controllers are ubiquitous in cyber-physical systems, from industrial control to drones. Their simplicity and well-understood dynamics make them perfect candidates for safety critical applications. However, simplicity of design is inevitably compromised to address departures of the real-world systems from the idealized linear dynamics. For example, low-pass filtering may be added to reduce noise in the feedback signal, gain scheduling may be added to account for non-linearity in response, and output clamping may be added to constrain the control signal to an acceptable range. As the complexity of the controller design grows, ensuring correct and safe operation of the system becomes a real challenge.

We explore an approach based on hybrid systems analysis and formal methods that has the potential to establish safety and correctness of a PID controller down to the source code level implementation. To test the approach, we apply it to a heading controller of an Unmanned Underwater Vehicle (UUV). First, we discuss how control theory stability analysis, based on Lyapunov function theory, can be adapted for PID specification verification in KeYmaera X, and what gaps remain to be filled to make the analysis sound. Next, we consider how the type constraint mechanisms and proof capabilities of SPARK / GNATprove may be used to formally verify that the source-level SPARK implementation of the controller logic satisfies the assumptions of the KeYmaera X specification proofs. In particular, we discuss progress made toward verifying that floating point calculations preserve the desired PID recurrences.
Bios:

Dr. C. Durward McDonell is a Senior Computer Scientist at JHU/APL. He has a Ph.D. in mathematics from the University of Minnesota. Durward joined APL in 2005. His research interests include formal analysis of computer systems, particularly theorem proving and type theory, and quantum computer science.

Dr. Raymond McDowell is a software assurance analyst with the Software Assurance Research and Applications lab at JHU/APL. He earned bachelor’s and master’s degrees in computer science and engineering from the Massachusetts Institute of Technology and a Ph.D. in computer and information science from the University of Pennsylvania. He has more than 15 years’ experience in industry and academia in the areas of automated support for software engineering and formal specification and analysis of programs and programming languages.

Dr. Daniel Genin is a Senior Computer Scientist at JHU/APL. He received his bachelor’s and doctoral degrees in mathematics from the Pennsylvania State University. After joining APL in 2012 Daniel contributed to a range of different projects from kernel integrity measurement to formal software verification. His current interests are in the area of formal software analysis and verification, as well as reverse engineering and embedded systems.

Ian Blumenfeld received his MA in mathematics from the University of Pennsylvania. He is interested in using formal reasoning in a variety of critical domains. His previous work has used formal methods tools in areas from malware analysis to cryptography. Prior to joining JHU/APL, Ian worked at CyberPoint, Galois and in the High Confidence Software and Systems Group at NSA.

Adam W. Cushman is a member of JHU/APL’s Senior Professional Staff and is Assistant Group Supervisor of the Special Concepts and Engineering Group. He has served as project or technical lead on a wide variety of projects and his areas of interest are in physics-based modeling and simulation, autonomous systems, navigation, guidance, and control, and software development.

Jay Guthrie is a controls engineer at JHU/APL. He received his master’s degree in engineering from Purdue University. He is currently working towards a Ph.D. in electrical engineering at Johns Hopkins University. He is interested in optimization-based control methods with applications in power systems, aerospace vehicles, and autonomous systems.
Poster sessions will be held in the atrium of the Governor Calvert House at the following times:

- 2:15 p.m. – 3:00 p.m. on Monday, May 7
- 2:15 p.m. – 3:00 p.m. on Tuesday, May 8

*bold name denotes presenter*
Bringing CPS Verification Tools to the CPS Community through the Cyber-Physical Systems Virtual Organization (CPS-VO)
Marcus Lucas
University of California, Los Angeles

Ground Impact and Hazard Mitigation for Safer UAV Operation
Andrew Poissant, Lina Castano, and Huan Xu
University of Maryland

Integrated Instruction Set Randomization and Control Reconfiguration for Securing Cyber-Physical Systems
Bradley Potteiger, Zhenkai Zhang, and Xenofon Koutsoukos
Vanderbilt University

Reusability of Modeling and Verification Components Among the DesignBIP and FSolidM Design Studios
*Anastasia Mavridou, **Áron Lászka, and *Janos Sztipanovits
*Vanderbilt University, **University of Houston

Static Analysis of Programmatically Generated Network Software: Challenges and Synergies
Jonathan Myers and Christopher Rouff
Johns Hopkins University Applied Physics Laboratory
BRINGING CPS VERIFICATION TOOLS TO THE CPS COMMUNITY THROUGH THE CYBER-PHYSICAL SYSTEMS VIRTUAL ORGANIZATION (CPS-VO)

Marcus Lucas
University of California, Los Angeles

Abstract:
One of the aims of the Cyber-Physical Systems Virtual Organization (CPS-VO) is to increase the awareness of the many verification tools available to the CPS community. With that in mind, the verification tools group in the CPS-VO actively maintains a searchable library of existing tools. In addition, the group has created introductory tutorials for several of the most popular tools. These allow users to jump in and test their systems, without having to wade through what can be extensive and sometimes opaque documentation. Near term, we intend to embed one or more such tools in the VO website itself, our goal being to make them accessible and easy to use, particularly for students.

Bio:
Marcus Lucas is a PhD student at the University of California Los Angeles under Professor Paulo Tabuada. His research interests include the modeling, analysis, control, and security of cyber-physical systems. Currently, he is working to develop model-free controllers that merge data-driven control techniques with control barrier functions.
INTEGRATED INSTRUCTION SET RANDOMIZATION AND CONTROL RECONFIGURATION FOR SECURING CYBER-PHYSICAL SYSTEMS

Bradley Potteiger, Zhenkai Zhang, and Xenofon Koutsoukos
Vanderbilt University

Abstract:
Cyber-Physical Systems (CPS) have been increasingly subject to cyber-attacks including code injection attacks. Zero day attacks further exasperate the threat landscape by requiring a shift to defense in depth approaches. With the tightly coupled nature of cyber components with the physical domain, these attacks have the potential to cause significant damage if safety-critical applications such as automobiles are compromised. Moving target defense techniques such as instruction set randomization (ISR) have been commonly proposed to address these types of attacks. However, under current implementations an attack can result in system crashing which is unacceptable in CPS. As such, CPS necessitate proper control reconfiguration mechanisms to prevent a loss of availability in system operation. This paper addresses the problem of maintaining system and security properties of a CPS under attack by integrating ISR, detection, and recovery capabilities that ensure safe, reliable, and predictable system operation. Specifically, we consider the problem of detecting code injection attacks and reconfiguring the controller in real-time. The developed framework is demonstrated with an autonomous vehicle case study.

Bio:
Bradley Potteiger is a PhD student in the Department of Electrical Engineering at Vanderbilt University with a research affiliation at the Institute of Software Integrated Systems. He received his MS. degree from Vanderbilt University in Electrical Engineering and his BS. degree in Computer Engineering from the University of Maryland, Baltimore County. His research at Vanderbilt is focused on Cyber Physical System (CPS) security with respect to protecting safety critical systems. Through his research he has worked with various research organizations within industry and government.
GROUND IMPACT AND HAZARD MITIGATION FOR SAFER UAV OPERATION

Andrew Poissant, Lina Castano, and Huan Xu
University of Maryland

Abstract:
As Unmanned Aircraft Vehicles (UAV) become more commonplace, there is a growing need for safer flight software that allows the UAV to understand and autonomously react to numerous flight anomalies. Decision-making software must allow the aircraft to perform tasks such as detect and avoid obstacles, correct non-critical failures mid-flight, or select the safest response during the presence of a critical flight anomaly. This research, performed at the University of Maryland, develops a ground impact and hazard mitigation (GIHM) module that interfaces with nominal UAV flight control software, which ultimately allows for a UAV to choose the safest landing option in case of a critical flight anomaly. The module incorporates generating a feasible ground impact footprint in real-time with LandScan USA data, a dataset that contains accurate spatial population data with a resolution of 30 m. All model development and simulation was performed in Matlab and Simulink.

Other works have used ground impact models for determining the reachable footprint for a UAV, and some have integrated census or tax data in a decision-making model that chooses the lowest hazard landing point. However, this work integrates a high-resolution population dataset and a decision-making engine with a model used to determine a UAV’s reachable footprint for many hazardous flight conditions. These conditions include loss of thrust and stuck ailerons, rudder, or elevator. Additionally, previous work did not integrate their ground impact models with flight control software, which is a fundamental feature of this work. The model we developed shows a preliminary reduction in simulated casualty expectation (fatalities per flight hour) by 97% when this new safety module is incorporated into a nominal UAV control software. This nominal flight software includes a mission plan, a path planning module, 6-DOF aircraft model, and flight controller. The path planning algorithm takes in the mission waypoints and real-time aircraft system states and updates the UAV’s flight plan. The updated flight plan is sent to the control system, where actuator commands for the aircraft are outputted and sent to the 6-DOF aircraft model. Finally, the 6-DOF aircraft states model outputs the real-time aircraft dynamics and sends those back to the path planning and control system blocks. The new ground impact hazard mitigation module takes in the aircraft dynamics, outputted by the 6-DOF aircraft states model, and uses those dynamics to determine whether the aircraft is experiencing a critical flight anomaly, calculates its reachable ground footprint.
based on the aircraft’s capabilities and flight dynamics, and determines the safest place to land based on expected population density in its potential landing areas.

This work aligns with the “Trusting Autonomy” theme for HCSS because of the significant advances that have been made relating to safer autonomous control of UAVs. The poster details the architecture and development of the new safety module that reduces the casualty expectation of any UAV. Furthermore, this work aligns with the goals of HCSS because of its capability to ensure more dependable and safer UAV emergency landings.

Bio:

Andrew Poissant is a graduate student at the University of Maryland getting his M.S. degree in systems engineering, with a focus on reliability and control. He will receive his M.S. degree in May 2018. He received a B.S. degree in chemical and biomolecular engineering from the University of Maryland in May 2017. His graduate research is in developing reachable footprint and ground risk models for safer descent of Unmanned Aerial Vehicles experiencing critical flight anomalies.
REUSABILITY OF MODELING AND VERIFICATION COMPONENTS AMONG THE DESIGNBIP AND FSOLIDM DESIGN STUDIOS

*Anastasia Mavridou, **Áron Lászka, and *Janos Sztipanovits

*Vanderbilt University, **University of Houston

Abstract:

Design studios facilitate system development by providing system architects with all necessary tools and services for modeling, verifying, and generating systems. Design studio components can be organized in three main categories: 1) semantic integration, 2) service integration, and 3) tool integration components. Semantic integration components comprise the domain of the modeling language, i.e., its metamodel, which explicitly specifies the building blocks of the language and their relations. Service integration components include dedicated model editors, code editors, and components for visualization of simulation and verification results. Additionally, service integration components include model transformation and code generation services, consistency and type checking mechanisms, model repositories, and version control services. Finally, tool integration components consist of interfaces and integration services towards integrated tools, such as simulation and verification tools.

We have developed two design studios: 1) the DesignBIP studio for modeling, verifying, and generating systems using the Behavior-Interaction-Priority (BIP) component-based framework and 2) the FSolidM studio for designing, verifying, and generating secure Solidity smart contracts for Ethereum blockchains. We have reused several components between the two design studios, such as model editors and integrated verification tools. In particular, the modeling languages that describe the behavior of DesignBIP models and FSolidM smart contracts are both based on Finite State Machines (FSM), which allowed us to reuse the FSM-based model editor. To verify the correctness of smart contracts, we have used BIP. To do that, we have defined an equivalent transformation from FSolidM FSMs to BIP FSMs and implemented it as a plugin in FSolidM. Additionally, we have integrated a tool in both design studios that transforms BIP models into observationally equivalent SMV models, which is the input language of the NuSMV model checker. To verify the correctness of BIP and FSolidM models, we have integrated into both design studios the NuSMV model checker and simulation tools from the BIP toolset.
We present the design flow of both studios, which is as follows. Initially, models are designed using dedicated model editors. Optionally, safety and security patterns can be used to simplify the modeling process by adding useful functionality or by enforcing important safety and security properties by construction. Next, the checking loop starts in which the models are checked for conformance. Additionally, the models can be checked for safety and liveness properties, as well as deadlock freedom using the integrated NuSMV model checker and BIP simulation tools. If the required conformance conditions are not satisfied by the model, then the design studio informs the user of the problematic nodes of the model and of the inconsistency causes to facilitate model refinement. Similarly, if the required safety/liveness properties are not satisfied by the model, the design studio returns counterexamples based on faulty execution paths. Finally, when the conformance conditions and safety/liveness properties are satisfied, the code (Java for DesignBIP and Solidity for FSolidM) may be generated.

Bios:

Anastasia Mavridou is a Postdoc at the Institute for Software Integrated Systems, Vanderbilt University, where she works with Prof. Janos Sztipanovits. Her research interests lie in the area of component-based design, modeling and analysis of concurrent systems with a focus on correct-by-construction techniques. She received her PhD in Computer Science from Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland in 2016 under the supervision of Prof. Joseph Sifakis and Dr. Simon Bliudze and her MSc in Computer Science from University of Tulsa, USA in 2012.

Áron Lászka is an Assistant Professor in the Department of Computer Science at the University of Houston. His research interests broadly revolve around the security and resilience of cyber-physical systems and Internet of Things, the economics of cyber-security, and game-theoretic modeling of security problems. Previously, he was a Research Assistant Professor at Vanderbilt University from 2016 to 2017, and a Postdoctoral Scholar at the University of California, Berkeley from 2015 to 2016. He graduated summa cum laude with a Ph.D. in Computer Science from the Budapest University of Technology and Economics in 2014. In 2013, he was a Visiting Research Scholar at Pennsylvania State University.

Dr. Janos Sztipanovits is currently the E. Bronson Ingram Distinguished Professor of Engineering at Vanderbilt University. He is founding director of the Institute for Software Integrated Systems (ISIS). His current research interest includes the foundation and applications of Model-Integrated Computing for the design of Cyber Physical Systems. His other research contributions include structurally adaptive systems, autonomous systems, design space exploration and systems-security co-design technology. He served as program manager and acting deputy director of DARPA/ITO between 1999 and 2002 and he was member of the US Air Force Scientific Advisory Board between 2006-2010. He was founding chair of the ACM Special Interest Group on Embedded Software (SIGBED). Dr. Sztipanovits was elected Fellow of the IEEE in 2000 and external member of the Hungarian Academy of Sciences in 2010. He graduated (Summa Cum Laude) from the Technical University of Budapest in 1970 and received his doctorate from the Hungarian Academy of Sciences in 1980.
STATIC ANALYSIS OF PROGRAMMATICALLY GENERATED NETWORK SOFTWARE: CHALLENGES AND SYNERGIES

Jonathan Myers and Christopher Rouff
Johns Hopkins University Applied Physics Laboratory

Abstract:

Code generated programmatically should be better-suited to analysis and verification than hand-written code. Instead, we find that existing static analysis tools are ill-suited to working with programmatically-generated code; paradigms used by the code generators may be misunderstood by static analyzers, leading to numerous false positives. More importantly, the types of code errors introduced through the use of programmatically-generated code may not be identified by current static analysis tools, leading to false negatives. To address this, we have developed a code generation system in tandem with development of an analysis system. The generated code is created with static analysis in mind, and the analyzers are tuned to the code generation system. This allows rigorous analysis of generated code and high confidence in its quality.

We demonstrate that rigorous analysis of programmatically-generated code is possible through our work on AGNES: Automatic Generation of Network Essential Software. In the AGNES system, a network protocol implementation is generated in the C language through a custom-made code generator. As network software is a common target of malicious attack, we seek to generate C code which is demonstrably free of common weaknesses taken from the CWE database. We present an initial set of 13 CWEs, the strategies used to ensure their absence in our generated code, and our experience extending the Clang Static Analyzer, and using KLEE and Frama-C to perform verification.

By carefully choosing and restricting the coding patterns used in code generation and tuning them to our software analysis system, we generate code which is designed for rigorous analysis. We examine the trade-offs between complexity of verification and complexity of code generation.

We demonstrate the use of programmatically generated code contracts as a means for the communication of intent between the code generator and the code verification system. We
find that many properties of C code are difficult to infer from the source code alone; code contracts provide a powerful tool for informing analysis, and greatly simplify the process of verification. When writing C code by hand, the addition of code contracts can be burdensome, but programmatically-generated code contracts can be added into our system with minimal additional complexity.

We demonstrate that rigorous validation of programmatically-generated source code is possible. Through concurrent and integrated development of code generation and static analysis systems, it is possible to achieve deep synergy leading to extremely high-quality analysis and high confidence in the generated code.

**Bio:**

Jonathan Myers is a computer scientist employed by the Johns Hopkins University Applied Physics Laboratory. He conducts research provides engineering in the field of computer security, focusing primarily on platform integrity and defensive technology as well as static analysis and formal verification. He has a MS in computer science from the University of Arizona, and has in the past worked on fields ranging from software obfuscation to asteroid discovery.
The conference dinner will be held at Chart House on Tuesday, May 9 at 6:30 p.m. Within walking distance of historic downtown Annapolis, Chart House offers fantastic waterfront views of City Dock, the state capital, and the U.S. Naval Academy. Located in the Eastport section of Annapolis, the restaurant has ample parking and is accessible by water taxi. For persons attending the dinner, tickets can be purchased (cash only) at the registration desk.

300 Second St | Annapolis, MD 21403 | Phone: 410.268.7166

DIRECTIONS FROM THE GOVERNOR CALVERT HOUSE

Driving (.9 mi)
- Head northwest toward Maryland Ave
- Exit the traffic circle onto School St
- Turn right onto Church Circle
- Turn right onto Duke of Gloucester St
- Slight right onto Compromise St
- Head over the Eastport Bridge and continue onto 6th St
- Turn left onto Severn Ave
- Turn left onto 2nd St
- Destination will be on the left

Walking (0.8 mi)
- Head south toward East St
- Exit the traffic circle onto Francis St
- Turn left onto Main St
- At the traffic circle, continue straight to stay on Main St
- Continue onto Compromise St
- Continue onto 6th St
- Turn left onto Severn Ave
- Turn left onto 2nd St
- Destination will be on the left
LOCAL RESTAURANTS
Armadillo’s Bar & Grill – 132 Dock Street, Annapolis, MD 21401
Veteran American grill offering burgers & beers along with dock views & occasional live music.

Baroak – 126 West St, Annapolis, MD 21401
Teddy Folkman’s Belgian-American eatery spotlights mussels & frites & a global array of craft beers.

Brown Mustache Coffee – 35 Maryland Ave, Annapolis, MD 21401
Super charming and cozy coffee place nestled in Old Fox Books run by a local food activist and his wife.

Cafe Normandie – 185 Main St, Annapolis, MD 21401
Quaint, long-running (since 1986) cafe for country French fare & wines, plus weekend breakfast.

Chart House – 300 2nd St, Annapolis, MD 21403
High-end chain eatery serving seafood & steakhouse fare in a classy setting with view.

Chick and Ruths Delly – 165 Main Street, Annapolis, MD 21401
Lively landmark diner featuring greasy-spoon breakfasts & piled-high sandwiches in kitschy environs.

Cracker Barrel – 115 Blue Jay Court, Stevensville, MD 21666
Brad’s favorite. Ask for the secret wine list!

Dock Street Bar & Grill – 136 Dock Street, Annapolis, MD 21401
Chesapeake Bay cuisine served daily until 1 a.m.

Dry 85 – 200 Main Street, 193 B Main Street, Annapolis, MD 21401
A modern industrial take on a Prohibition-era speakeasy.

Federal House – 22 Market Space, Annapolis, MD 21401
Casual eatery in historic digs offering New American tavern fare.

Galway Bay Irish Pub – 63 Maryland Ave, Annapolis, MD 21401
Convivial pub featuring classic Gaelic grub & American standards in warm, brick-walled environs.

Harry Browne’s – 66 State Circle, Annapolis, MD 21401
A captivating historic restaurant/lounge. Lavish lunches, divine dinners and sumptuous Sunday brunch.
LOCAL
RESTAURANTS

Harvest – 26 Market Space, Annapolis, MD 21401
A casual dining and tap room in downtown Annapolis.

Iron Rooster – 12 Market Space, Annapolis, MD 21401
Creative all-day breakfast menu and American comfort food.

Joss Café & Sushi Bar – 1959 Main Street, Annapolis, MD 21401
Voted ‘Best Sushi Restaurant’ in Annapolis for 8 years running by the readers of What’s Up? Magazine.

Lemongrass – 167 West St., Annapolis, MD 21401
Fresh, authentic Thai Cuisine in a warm contemporary environment.

Level – A Small Plates Lounge – 69 West St., Annapolis, MD 21401

Mason’s Famous Lobster Rolls – 188 Main Street, Annapolis, MD 21401
Lobster roll sandwiches, salads, and soups round out the menu at this pint-sized, simple counter serve.

Metropolitan Kitchen and Lounge – 169 West St., Annapolis, MD 21401
Casual restaurant with a broad American menu & full bar plus rooftop deck & local live music.

O’Brien’s Oyster Bar – 113 Main St., Annapolis, MD 21401
Imaginative seafood dishes and nouveau American cuisine. Dancing and live entertainment nightly.

Preserve – 164 Main Street, Annapolis, MD 21401
Preserve is a casual American restaurant using sustainable and local products.

Rams Head Tavern – 33 West St., Annapolis, MD 21401
An Annapolis landmark since 1989!

Rice Workshop – 138 Main St, Annapolis, MD 21401
Vietnamese street style food.

Sofi’s Crepes – 1 Craig Street, Annapolis, MD 21401
Brunch crepes with sweet and savory fillings.

Vin 909 – 909 Bay Ridge Ave., Annapolis, MD 21403
Great food. Great wine. Excellent service. It’s worth the walk, even in the rain.