HCSS

HIGH CONFIDENCE
SOFTWARE & SYSTEMS

The 25th Annual

HIGH CONFIDENCE SOFTWARE & SYSTEMS CONFERENCE

MAY 12-14, 2025

Welcome to HCSS 2025

The HCSS steering committee is pleased to welcome you to the 25th 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 the HCSS tradition of excellence. World-class research scientists from academia, industry, and government will draw on a broad range of experiences to deliver three days of compelling 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 operations. Systems of systems that incorporate the latest advances in computing, communication, information, machine learning, and control technologies present extremely challenging problems for assurance and resilience. HCSS presenters and participants will discuss advances toward mitigations or solutions to those problems and explore pathways to achieving high confidence in future software and systems.

We are pleased to once again co-locate the High Confidence Software and Systems (HCSS) Conference with two vibrant communities whose work complements the goals of this gathering.

The Software Certification Consortium (SCC), formed in 2007, brings together industry researchers, government regulators, and academics to explore certification challenges for systems with significant software components—such as those in aerospace, automotive, medical devices, nuclear energy, and defense. The group aims to objectively recommend processes and standards that impact the certification of these critical systems.

The Trusted Computing Center of Excellence™ (TCCoE) Summit also returns this year. The TCCoE supports the principled development and deployment of trustworthy systems built on the seL4® microkernel by lowering barriers to adoption across research, industry, and government.

We hope you find the 2025 Conference stimulating and informative. Thank you for your attendance and continued support of the HCSS community!

This year's HCSS talks focus on four key themes

AI and Models in the Software Development Lifecycle:

Exploring how AI and model-based approaches can enhance all phases of software development, from requirements and design to implementation and maintenance.

Formal Verification of AI/ML Systems:

Advancements in applying formal methods to verify the safety, robustness, and generalization properties of machine learning models.

Formalizing Pure Math:

Leveraging theorem provers and formal reasoning to advance mathematical research and enhance AI applications in mathematical domains.

Novel Automated Reasoning Applications:

Examining the use of automated reasoning tools to improve resilience, regulatory compliance, cybersecurity, and other critical domains.

Co-Located Meetings

May 15–16, 2025

Software Certification Consortium (SCC) Meeting

Location: Maryland Inn

The Software Certification Consortium (SCC) is a community of researchers and practitioners focused on the development and certification of high-integrity systems. The consortium typically meets twice a year, with each meeting structured around key themes that advance a long-term SCC work plan.

The upcoming SCC Meeting will focus on two critical themes:

Protecting safety-critical software-intensive systems from malicious action *(including integrated safety-security assurance)*

The potential of AI in safety assurance

Additional information about past meetings can be found at sos-vo.org/group/scc.

Trusted Computing Center of Excellence (TCCoE) Summit

Location: Governor Calvert Ballroom

The Trusted Computing Center of Excellence (TCCoE) Summit brings together leaders from government, industry, and academia to advance the principled development and deployment of trustworthy systems.

Originally focused on facilitating the adoption of the seL4 microkernel in critical defense applications, the TCCoE has since expanded its mission to support the adoption of formally verified software (including seL4) in systems requiring the highest levels of assurance.

This year's Summit will be held in conjunction with the High Confidence Software and Systems (HCSS) Conference, offering valuable synergies between communities with aligned missions and complementary technologies.

The theme of the 2025 TCCoE Summit is:

Building Trustworthy Systems with Proven Components

Additional information about TCCoE can be found at https://trustedcomputingcoe.org/.

Table of Contents

General Information

Registration

Conference registration will take place in the State Lobby of the Governor Calvert House and will be open Monday through Friday, from 8:00 AM to 4:30 PM.

For SCC Meeting participants, registration will also be available in the Maryland Inn Lobby on Thursday and Friday, from 8:00 AM to 4:30 PM.

Wi-Fi

Network Name: Historic_Conference
Password: Annapolis

Poster Session

A poster session will be held on Monday, May 12, from 3:00 to 3:30 PM in the Atrium.

Conference Presentations

Scan the QR code below to access conference presentations and posters online after the event.

Parking

Parking at the Historic Inns of Annapolis is available via valet service only. A special rate of $26 per day has been arranged for daily attendees. Please note: parking is extremely limited and spaces are available on a first-come, first-served basis. Nearby public parking options include the Noah Hillman Garage and Gott's Garage, both within walking distance.

Survey

We value your feedback! Please take a moment to complete our short post-event survey by visiting the URL below:
https://sos-vo.org/form/hcss-2025-survey

Conference Organization

Co-Chairs

Darren Cofer
Collins Aerospace

Darren Cofer is a Principal Fellow at Collins Aerospace. He earned his PhD in Electrical and Computer Engineering from The University of Texas at Austin. His area of expertise is developing and applying advanced analysis methods and tools for verification and certification of high-integrity 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.

Dr. Cofer has served as principal investigator on government-sponsored research programs with NASA, NSA, AFRL, and DARPA, developing and using formal methods for verification of safety and security properties.
He served on RTCA committee SC-205 developing new certification guidance for airborne software (DO-178C) and was one of the developers of the Formal Methods Supplement (DO-333). He is a member of SAE committee G-34 for Artificial Intelligence in Aviation, the Aerospace Control and Guidance Systems Committee (ACGSC), and a senior member of the IEEE.

Sandeep Neema
Vanderbilt University

Sandeep Neema is a Professor with the Department of Computer Science, and Director of the Institute for Software Integrated Systems, Vanderbilt University. He served a Program manager at DARPA's Information Innovation Office (I2O) from July 2016 till September 2022. In his tenure at DARPA he conceived, developed, and managed influential programs at the intersection of Artificial Intelligence and Cyber Physical Systems, that included programs such as Assured Autonomy, Symbiotic Design of Cyber Physical Systems, and Assured Neurosymbolic Learning and Reasoning. His research interests include Cyber Physical Systems, Model-based Design Methodologies, Artificial Intelligence and Machine Learning, and Distributed Real-time Systems. 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.

Conference Organization

Program Co-Chairs
Darren Cofer, Collins Aerospace
Sandeep Neema, Vanderbilt University

Steering Commitee
Perry Alexander, The University of Kansas
June Andronick, Proofcraft
Kathleen Fisher, DARPA
John Hatcliff, Kansas State University
John Launchbury, Galois, Inc.
Patrick Lincoln, SRI International
Stephen Magill, Sonatype
Brad Martin, NSA
Adam W, National Cyber Security Centre
Lee Pike, Amazon Web Services
Ray Richards, Leidos
Kristin Yvonne Rozier, Iowa State University
William Scherlis, Carnegie Mellon University
Eric W. Smith, Kestrel Institute
Sean Weaver, DARPA
Matthew Wilding, DARPA

Meeting Organization
Katie Dey, Vanderbilt University

Event Support
Anne Dyson, Independent
Regan Williams, Vanderbilt University

Graphic and Web Design
Tony Guzman, Vanderbilt University
Amy Karns, VU Retired

The HCSS Conference is made possible through the generous support of the following sponsors:

aws

galois

TRUSTED SCIENCE AND TECHNOLOGY

twosix TECHNOLOGIES

KU INSTITUTE FOR INFORMATION SCIENCES The University of Kansas

THE INSTITUTE

MONDAY, MAY 12

THEME: Formal Verification of AI/ML Systems

09:00 - 10:00 **KEYNOTE: From Neural Network Verification to Formally Verifying Neuro-Symbolic Artificial Intelligence (AI)**
Taylor Johnson, Vanderbilt University

10:00 - 10:30 **BREAK**

10:30 - 11:00 **AI for Formal Verification and Formal Verification for AI**
David Dalrymple, ARIA

11:00 - 11:30 **VeriX: Verified Explainability of Deep Neural Networks**
Min Wu, Stanford

11:30 - 12:00 **Semantic Verification of Foundation Models Using Mechanistic Verification and Concept Probes**
Susmit Jha, SRI

12:00 - 13:30 **LUNCH (on your own)**

THEME: Formalizing Pure Math

13:30 - 14:30 **KEYNOTE: Verified Collaboration: How Lean is Transforming Mathematics, Programming, and AI**
Leonardo De Moura, Lean FRO and AWS

14:30 - 15:00 **Reasoning-Enabling Representations of Math Challenges**
Marijn Heule, CMU

15:00 - 15:30 **Poster Session / Break**

15:30 - 16:00 **Future Program: Exponentiating Mathematics (expMath)**
Patrick Shafto, DARPA

16:00 - 16:30 **Neuro-Symbolic Techniques for LLM-based Code Generation and Auto-Formalization of Proofs**
Prithwish Jana and Vijay Ganesh, Georgia Institute of Technology

16:30 - 17:00 **From Modal Logic to Computing: The Case for Hybrid Logics in Formal Methods**
William Harrison, Galois

17:00 **Adjourn for the day**

TUESDAY, MAY 13

THEME: Novel Automated Reasoning Applications

09:00 - 10:00 **KEYNOTE: Formal Verification of Financial Infrastructure with Imandra**
Grant Passmore, Imandra

10:00 - 10:30 **BREAK**

10:30 - 11:00 **Automated Reasoning for UAV Safety & Security: The DATUM Protocol Stack**
Max Taylor, Idaho National Laboratory

11:00 - 11:30 **Modeling and Formal Analysis of High-Assurance Mixed-Reality Systems**
Junaid Babar and Isaac Amundson, Collins Aerospace

11:30 - 12:00 **Software Understanding for National Security (SUNS) Partnership Forum Report**
Douglas Ghormley, SNL

12:00 - 13:30 **LUNCH (on your own)**

13:30 - 14:30 **HCSS 25th Anniversary Panel**
Moderator: John Launchbury, Galois
Panelists: Patrick Lincoln, SRI
William Scherlis, CMU SEI
Matthew Wilding, DARPA

14:30 - 15:00 **An Experiment Using Layered Attestation**
Perry Alexander, The University of Kansas

15:00 - 15:30 **BREAK**

15:30 - 16:00 **Formalizing and Automating the Discovery of Weird States and Machine Primitives for High-Confidence Software**
Meera Sridhar, University of North Carolina Charlotte

16:00 - 16:30 **ModelForge: Using AI to Improve Security Protocols**
Martin Duclos, Mississippi State University

16:30 - 17:00 **Secure Protocols via LLMs and CPSA**
Lauren Brandt, MITRE

17:00 **Adjourn for the day**

18:30 **HCSS Conference Dinner**
Chart House Prime | 300 Second St, Annapolis, MD 21403

WEDNESDAY, MAY 14

THEME: AI and Models in the Software Development Lifecycle

09:00 - 10:00	**KEYNOTE: Neurosymbolic Programming and the Path to Safe AI** Armando Solar-Lezama, MIT CSAIL
10:00 - 10:30	**BREAK**
10:30 - 11:00	**Reconciling Distributed System Implementation and Design with Neuro-Symbolic Reasoning** Tristan Ravitch, AWS
11:00 - 11:30	**Translating C to Rust: Better, cheaper, faster** Per Larsen, Immunant, Inc.
11:30 - 12:00	**Functors as Bridges between AI-Generated Code and Formal Models for High-Confidence Software Systems** Sumit Jha, Florida International University, Miami
12:00 - 13:30	**LUNCH (on your own)**
13:30 - 14:00	**Secure AI Through Verification, Transparency, and Fairness** Jessica Inman, GTRI
14:00 - 14:30	**Automated SysML v2 System Model to Memory-Safe Language Code Generation with Integrated AI Assistance** David Hardin, Collins Aerospace
14:30 - 15:00	**LLM-enabled Software Testing** Fanxin Kong and Weizhe Xu, University of Notre Dame
15:00 - 15:30	**Break**
15:30 - 16:00	**Towards Trustworthy Integration of Generative AI in the MBSE Development Lifecycle** Amer Tahat, Collins Aerospace
16:00 - 16:30	**Combining AI and Models to Identify Faults in Business Logic** Daniel Balasubramanian, Vanderbilt University
16:30 - 17:00	**TCCoE & SCC Introductions** **HCSS Closing Remarks**

PROUD SPONSOR OF HCSS CONFERENCE 2025

INSTITUTE FOR INFORMATION SCIENCES

UNIVERSITY OF KANSAS

CREATING AND DISSEMINATING FUNDAMENTAL KNOWLEDGE AND NEW TECHNOLOGIES

SCAN THE QR CODE TO VISIT OUR WEBSITE:

KU INSTITUTE FOR INFORMATION SCIENCES
The University of Kansas

KEYNOTE

From Neural Network Verification to Formally Verifying Neuro-Symbolic Artificial Intelligence (AI)

Taylor Johnson
Vanderbilt University

Dr. Taylor T. Johnson, PE, is A. James and Alice B. Clark Foundation Chancellor Faculty Fellow, Associate Professor, and Associate Chair of Computer Science (CS) at Vanderbilt University, where he directs the Verification and Validation for Intelligent and Trustworthy Autonomy Laboratory (VeriVITAL) and is a Senior Research Scientist in the Institute for Software Integrated Systems (ISIS). Dr. Johnson's research has been published in venues such as AAAI, CAV, EMSOFT, FM, FORMATS, HSCC, ICSE, ICDM, ICCPS, IJCAI, MEMOCODE, NFM, RTSS, SEFM, STTT, TNNLS, UAI, among others, several of which have received best paper awards. Dr. Johnson earned a PhD in Electrical and Computer Engineering (ECE) from the University of Illinois at Urbana-Champaign in 2013, where he worked in the Coordinated Science Laboratory with Prof. Sayan Mitra, and a BSEE from Rice University in 2008. Dr. Johnson is a recipient of the Air Force Office of Scientific Research (AFOSR) Young Investigator Program (YIP) award and the National Science Foundation (NSF) Computer and Information Science and Engineering (CISE) Research Initiation Initiative (CRII) award, and his group's research is or has recently been supported by AFOSR, ARO, AFRL, DARPA, Mathworks, NSA, NSF, NVIDIA, ONR, Toyota, and USDOT.

Abstract

The ongoing renaissance in artificial intelligence (AI) has led to the advent of data-driven machine learning (ML) methods deployed within components for sensing, perception, actuation, and control in safety-critical cyber-physical systems (CPS). While these learning-enabled components (LECs) are enabling autonomy in systems such as autonomous vehicles and robots, ensuring such components operate reliably in all scenarios is extraordinarily challenging, as demonstrated in part through recent accidents in semi-autonomous/autonomous CPS and by adversarial ML attacks. We will discuss formal methods for assuring specifications---mostly robustness and safety---in autonomous CPS and subcomponents thereof using our software tools NNV (https://github.com/verivital/nnv) and Veritex (https://github.com/verivital/veritex), developed partly in DARPA Assured Autonomy and Assured Neuro Symbolic Learning and Reasoning (ANSR) projects. These methods have been evaluated in CPS development with multiple industry partners in automotive, aerospace, and robotics domains, and allow for formally analyzing neural networks and their usage in closed-loop systems. We will then discuss how these methods are enabling verification for the third wave of AI systems, namely neuro-symbolic systems. We will describe a class of neuro-symbolic systems we have been developing called neuro-symbolic behavior trees (NSBTs), which are behavior trees (a form of hierarchical state machine becoming widely used in robotics) that execute neural networks, and for which we have been developing verification methods implemented in a tool called BehaVerify (https://github.com/verivital/behaverify). We will also discuss relevant ongoing community activities we help organize, such as the Verification of Neural Networks Competition (VNN-COMP) held with the International Conference on Computer-Aided Verification (CAV) and the Symposium on AI Verification (SAIV) the past few years, as well as the AI and Neural Network Control Systems (AINNCS) category of the hybrid systems verification competition (ARCH-COMP) also held the past few years. We will conclude with a discussion of future directions in the broader safe and trustworthy AI domain, such as in ongoing projects verifying neural networks used in medical imaging analysis and malware classifiers.

AI for Formal Verification and Formal Verification for AI

David Dalrymple
ARIA

David ‘davidad’ Dalrymple has spent much of the last five years formulating a vision for how mathematical approaches could guarantee reliable and trustworthy AI. He is also a founding ARIA Program Director, where he is building a program on how techniques from formal verification of cyber-physical systems could deliver safe and transformative AI.

Davidad was a Research Fellow in technical AI safety at the University of Oxford. He previously studied biophysics at Harvard, and AI at MIT, where he was the youngest ever graduate student, attaining a master’s at 16.

Outside of academia, Davidad has worked on advanced software engineering from multiple technological vantage points. He co-invented the top-40 cryptocurrency Filecoin, led an international neuroscience collaboration, and was a senior software engineer at Twitter and multiple start-ups.

Photo by Matilda Hill-Jenkins

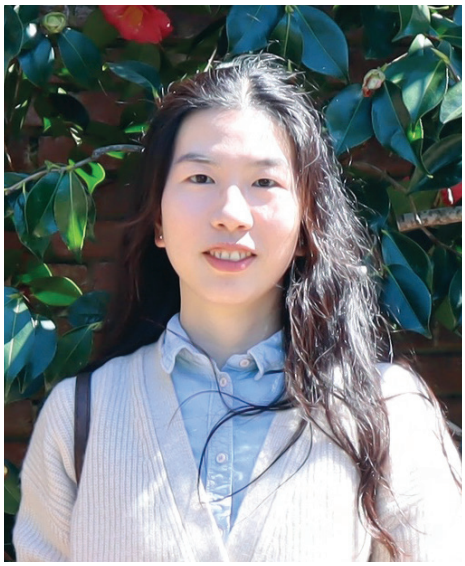
Abstract

This talk will cover three emerging trends at the intersection of AI and formal verification, and their interplay. (1) In formal verification of software, we are accustomed to scarcity of expert practitioners, from whom a lot of hours are required to create verified software, resulting in a relatively small amount of the world’s software actually being verified, even within critical infrastructure sectors. As LLMs advance in general coding abilities, we are also seeing increasing ability to construct formal proofs, including for software verification subgoals that could not previously be automated. Given the exponential trends at play, we should expect at least a 10-100x decrease in expert hours required to construct verified software over the next few years, resulting in greatly increased adoption and Jevons-paradoxical increase in demand for experts in the least mechanizable skills, especially specification review. (2) Increasingly powerful AI systems trained with reinforcement learning from feedback from so-called verifiers, which test Python code or the like, are increasingly cheating. It may soon become crucial to incorporate formal verifiers into the RL training process in order to continue pushing capabilities. The formal verification community should be eager to support this, because it will accelerate trend (1). (3) Beyond the next few years, even if frontier AI systems are RL-trained only to solve formally verifiable problems, they may begin to cheat and dissemble at deployment-time in very dangerous ways, unless they are also deployed only to solve formally verifiable problems. This implies that we ought to expand the scope of formal verification to cyber-physical systems and beyond. We may take as a premise that AI may be able to solve nearly any NP problems that arise in practice, as long as we can actually supply correct certificate checkers for every such problem, and the description length of problems, solutions, and certificates is manageable (within LLM context windows). Expanding formal verification to cyber-physical systems with small neural control components is the purpose of the Safeguarded AI programme that I direct at UK ARIA.

VeriX: Verified Explainability of Deep Neural Networks

Min Wu
Stanford

Min Wu is a Postdoctoral Scholar with Prof. Clark Barrett at the Department of Computer Science, Stanford University. She is also affiliated with Stanford Center for AI Safety and Center for Automated Reasoning. Previously, she completed her PhD in Computer Science under the supervision of Prof. Marta Kwiatkowska at the University of Oxford. Her research focuses on safe and trustworthy AI, positioned at the intersection of AI and formal methods. The grand vision of her work is to develop AI systems, particularly those deployed in high-stakes applications, that are verifiably reliable and transparent.



Abstract

In this paper, we present VeriX (Verified eXplainability), a tool for producing optimal verified explanations, in the sense that fixing a minimal subset of input features is sufficient to ensure the invariance of a model's prediction. We start by providing intuition for our VeriX approach by analyzing an example explanation in Figure 1. This explanation is generated for a fully-connected model trained on the MNIST dataset. Model-agnostic explainers such as Anchors rely on partitioning an image into a disjoint set of segments and then selecting the most prominent segment(s). Figure 1b shows "2" divided into 3 parts using k-means clustering. Based on this segmentation, the purple and yellow parts would be chosen for the explanation, suggesting that the model largely relies on these segments to make its decision. This also matches our intuition, as a human would immediately identify these pixels as containing information and disregard the background. However, does this mean it is enough to focus on the salient features when explaining a classifier's prediction? Not necessarily. VeriX's explanation is highlighted in green in Figure 1c. It demonstrates that whatever is prominent is important but what is absent in the background also matters. We observe that VeriX not only marks those white pixels forming the silhouette of "2" but also includes some background pixels that might affect the prediction if changed. For instance, neglecting the bottom white pixels may lead to a misclassification as a "7"; meanwhile, the classifier also needs to check if the pixels along the left and in the middle are not white to make sure it is not "0" or "3".

We present the overall workflow of our VeriX framework in Figure 2. Starting from the left, the inputs are a network f and an input x. The first step is to obtain a sensitivity map of all the input features and, by ranking their individual sensitivity, produce a traversal order. We use a bound propagation-based technique and a heuristic method to obtain such orders. The traversal order is then passed to the main traversal algorithm, which computes optimal verified explanations. We propose two optimizations, one based on binary search and one adapted from the well-known QuickXplain algorithm. These can significantly reduce computation time compared to the simple sequential method, which processes features one by one. The key difference between these two is that the former reduces the time but not the size, as it does not change the traversalorder, whereas the latter improves both size and time. Compared to the binary search-based technique, the QuickXplain technique computes comparatively smaller-sized explanations but takes more time, thus providing an alternative point in the size-time trade-off. The check procedure is used by the traversal methods to formally check the soundness of a candidate explanation. We incorporate a simple but efficient confidence ranking algorithm which further reduces generation time. The confidence ranking is orthogonal to the other optimizations and benefits all three traversal approaches.

We applied VeriX to the real-world safety-critical aircraft taxiing scenario shown in Figure 3. The vision-based autonomous taxiing system needs to make sure the aircraft stays on the taxiway utilizing only pictures taken from the camera on the right wing.

The task is to evaluate the cross-track position of the aircraft so that a controller can adjust its position accordingly. To achieve this, a regression model is used that takes a picture as input and produces an estimate of the current position. A preprocessing step crops out the sky and aircraft nose, keeping the crucial taxiway region (in the red box). This is then downsampled into a gray-scale image of size 27×54 pixels. We label each image with its corresponding lateral distance to the runway centerline together with the taxiway heading angle. We trained a fully-connected regression network on this dataset, referred to as the TaxiNet model to predict the aircraft's cross-track distance. Figure 4 exhibitsVeriX applied to the TaxiNet dataset, including a variety of taxiway images with different heading angles and number of lanes. For each taxiway, we show its VeriX explanation accompanied by the cross-track estimate. We observe that the model is capable of detecting the more remote line—its contour is clearly marked in green. Meanwhile, the model is mainly focused on the centerline (especially in Figures 4b, 4d, 4e, and 4f), which makes sense as it needs to measure how far the aircraft has deviated from the center. Interestingly, while we intuitively might assume that the model would focus on the white lanes and discard the rest, VeriX shows that the bottom middle region is also crucial to the explanation (e.g., as shown in Figures 4a and 4c). This is because the model must take into account the presence and absence of the centerline. This is in fact in consistent with our observations about the black background in MNIST images (Figure 1). We used $\epsilon = 5\%$ for these explanations, which suggests that for modest perturbations (e.g., brightness change due to different weather conditions) the predicted cross-track estimate will remain within an acceptable discrepancy, and taxiing will not be compromised.

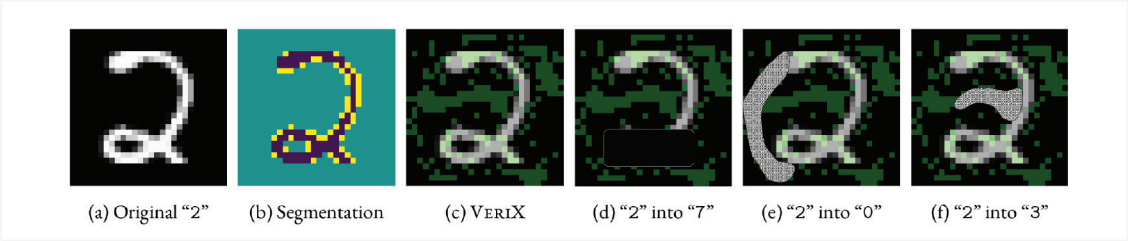


Figure 1: Intuition for our VeriX approach: (a) An MNIST handwritten "2"; (b) Segmentation of "2" into 3 partitions; (c) Our VeriX explanation (green pixels) of "2"; (d)(e)(f)Masking white pixels or whitening black pixels may turn "2" into possible counterfactuals.

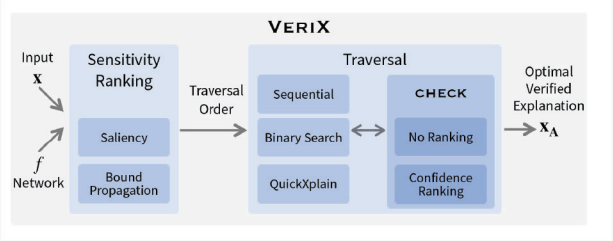


Figure 2: The VeriX framework.

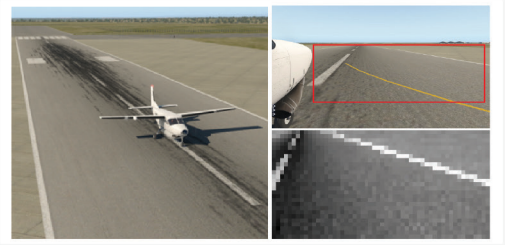


Figure 3: An autonomous aircraft taxiing scenario. Pictures taken from the camera fixed on the right wing are cropped (red box) and downsampled.

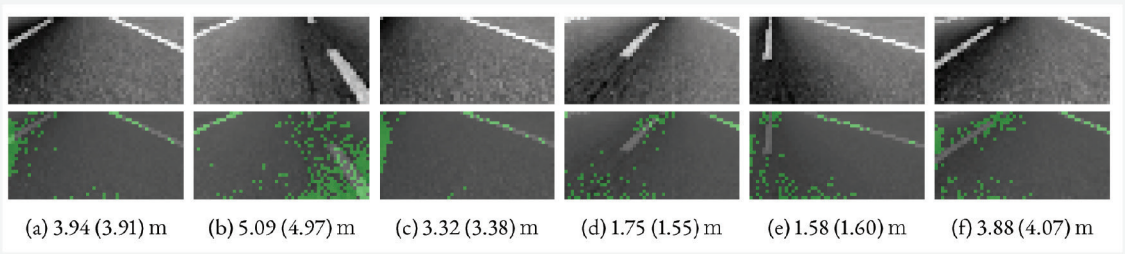


Figure 4: VeriX applied to theTaxiNet dataset – each column includes a sampled camera view (top), its VeriX explanation(bottom), and the cross-track estimate of the form "actual (estimate) meters".

Semantic Verification of Foundation Models Using Mechanistic Verification and Concept Probes

Susmit Jha
SRI

Dr. Susmit Jha is a Technical Director in the Computer Science Laboratory at SRI International. His research lies at the intersection of formal methods, machine learning, and control theory, with a focus on building trusted artificial intelligence and correct-by-construction autonomous systems.

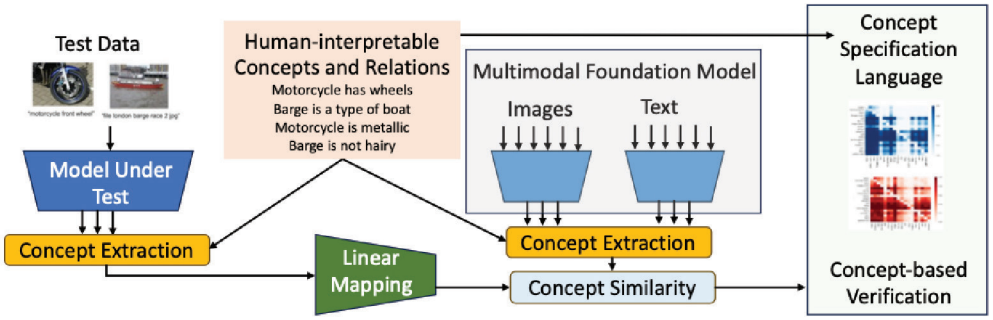
Dr. Jha earned his Ph.D. in Computer Science from UC Berkeley in 2011. His dissertation, "Automated Synthesis Using Structurally Constrained Induction and Deduction," was supported by the Berkeley Fellowship and received the Leon O. Chua Award. Prior to joining SRI, he worked at Intel Strategic CAD Labs and the Raytheon Technologies Research Center in Berkeley. He is the recipient of several prestigious awards, including the 10-year Most Influential Paper Award at IEEE/ACM 42nd International Conference on Software Engineering (ICSE) 2020, top 10% paper award at the 17th International Design Conference (DESIGN) 2022, and Best Paper Award Nominations at IEEE/ACM 14TH International Conference on Cyber-Physical Systems ICCPS 2023, 4th International Conference on AI Engineering – Software Engineering for AI, 2025, and IEEE 41st Military Communications Conference (MILCOM), 2023. He has published over 100 peer-reviewed publications with over 4500 citations in AI, ML, and automated reasoning venues such as NeurIPS, ICLR, ICML, CVPR, ICCV, AAAI, IJCAI, JAR, PLDI, and CAV.

Dr. Jha has been a Principal Investigator on DoD and other US Govt. programs on trustworthy, resilient, and neuro-symbolic AI, including DARPA TIAMAT, DARPA ANSR, DARPA SDCPS, DARPA Assured Autonomy, IARPA TrojAI, ARL IoBT CRA, NSA GenAI4Cyber, NSF Self-improving CPS, and NSF Duality-based Synthesis. The Defense Advanced Research Projects Agency (DARPA) has named Dr. Susmit Jha to the Information Science & Technology (ISAT) Study Group beginning in July 2023. He co-developed the NEDL technology in DARPA SoSITE, which led to the STITCHES Air Force Program of Record. Dr. Jha cofounded p-1.ai - a startup working to develop an artificial general engineering intelligence that can help the human species design physical systems more efficiently and at unprecedented levels of complexity and currently serves as a Senior Technical Advisor to p-1. He is also a Technical Advisor to Confidencial, focused on AI for Data Security.

Abstract

Neuro-symbolic Computing and Intelligence
SRI International
Menlo Park, CA, USA
susmit.jha@sri.com

The challenge of assurance of deep learning models, particularly large foundation models, can be addressed using neuro-symbolic methods. The concept probes developed to explain inference in foundation models can be used to align latent representations in neural networks with symbolic abstractions, helping to analyze and extract meaningful concepts. Further, mechanistic interpretation methods can enable the extraction of common circuits and algorithms from inference patterns of foundation models on similar queries. This enables the symbolic specification and verification of semantic properties that encode the logical composition of relationships over concepts represented in neural latent space. These properties go beyond the typical robustness properties used in traditional formal verification of deep neural networks. By grounding the assurance process in interpretable symbolic abstractions, neuro-symbolic methods enhance the trustworthiness and reliability of AI systems in safety-critical domains. In this talk, we will describe how this new paradigm has been used to scalably verify foundation models and check their alignment with semantic properties.



References

[1] Ravi Mangal, Nina Narodytska, Divya Gopinath, Boyue Caroline Hu, Anirban Roy, Susmit Jha, and Corina S P˘as˘areanu, "Concept-based analysis of neural networks via vision-language models," in International Symposium on AI Verification, 2024.
[2] Claudio Spiess, David Gros, Kunal Suresh Pai, Michael Pradel, Md Rafiqul Islam Rabin, Amin Alipour, Susmit Jha, Prem Devanbu, and Toufique Ahmed, "Calibration and correctness of language models for code," in IEEE/ACM International Conference on Software Engineering, 2024.
[3] Sumit Kumar Jha, Susmit Jha, Rickard Ewetz, and Alvaro Velasquez, "On the design of novel attention mechanism for enhanced efficiency of transformers," in Proceedings of the 61st ACM/IEEE Design Automation Conference, 2024, pp. 1–6.

KEYNOTE

Verified Collaboration: How Lean is Transforming Mathematics, Programming, and AI

Leonardo De Moura
Lean FRO and AWS

Leo is a Senior Principal Applied Scientist in the Automated Reasoning Group at AWS. In his spare time, he dedicates himself to serving as the Chief Architect of the Lean FRO, a non-profit organization that he proudly co-founded alongside Sebastian Ullrich. He is also honored to hold a position on the Board of Directors at the Lean FRO, where he actively contributes to its growth and development. Before joining AWS in 2023, he was a Senior Principal Researcher in the RiSE group at Microsoft Research, where he worked for 17 years starting in 2006. Prior to that, he worked as a Computer Scientist at SRI International. His research areas are automated reasoning, theorem proving, decision procedures, SAT and SMT. He is the main architect of several automated reasoning tools: Lean, Z3, Yices 1.0 and SAL. Leo's work in automated reasoning has been acknowledged with a series of prestigious awards, including the CAV, Haifa, and Herbrand awards, as well as the Programming Languages Software Award by the ACM. Leo's work has also been reported in the New York Times and many popular science magazines such as Wired, Quanta, and Nature News.

Abstract

Imagine a world where mathematicians, programmers, and AI systems can collaborate with complete trust in each other's work. This is the promise of Lean, an open-source project that's transforming how we approach mathematics, software development, and artificial intelligence. Lean provides machine-checkable proofs, eliminating the need for manual verification and allowing humans and AI to build upon each other's work with unprecedented confidence. By addressing the "Trust Bottleneck," Lean opens doors to cross-disciplinary collaboration. In this talk, we'll explore how Lean is impacting these fields. We'll see how it's providing mathematicians with a new way to construct and verify complex proofs, enabling software developers to rigorously verify critical systems, and creating a foundation for more reliable AI for science and mathematics. We'll also discuss the role of the Lean Focused Research Organization (FRO), a non-profit dedicated to advancing Lean and growing its community. The FRO is driving Lean's development as both a proof assistant and an extensible programming language, empowering users to customize its capabilities for diverse applications. Through real-world examples from academia and industry, we'll discover how Lean is paving the way for a more efficient, reliable, and collaborative future in mathematics, software development, and AI.

TRUSTED

SCIENCE AND TECHNOLOGY

Building Trustworthy Capabilities by Trusted People and Technology

Reasoning-Enabling Representations of Math Challenges

Marijn Heule
CMU

Marijn Heule is an Associate Professor of Computer Science at Carnegie Mellon University. His contributions to automated reasoning have enabled him and others to solve hard problems in formal verification and mathematics. He has developed award-winning satisfiability (SAT) solvers. His preprocessing and proof-producing techniques are used in many state-of-the-art automated reasoning tools. Marijn won multiple best paper awards at international conferences, including at SAT, CADE, IJCAR, TACAS, HVC, LPAR, and IJCAI-JAIR. He is one of the editors of the Handbook of Satisfiability. This 1500+ page handbook (second edition) has become the reference for SAT research.

Abstract

Automated reasoning has a rich history of success in solving mathematical challenges, including century-old open problems such as Keller's conjecture, the Empty Hexagon problem, and Schur Number Five. However, the effectiveness of these tools heavily depends on how problems are represented, making it difficult for non-experts to fully leverage their power. In this talk, we present reasoning-enhancing representations for several hard mathematical problems. We also discuss key principles for developing such representations and explore automation strategies to optimize them, ultimately improving accessibility for non-experts.

Exponentiating Mathematics (expMath)

Patrick Shafto
DARPA

Patrick Shafto is Program Manager in DARPA's Information Innovation Office (I2O) and Professor of Mathematics and Computer Science at Rutgers University Newark. Prior to joining DARPA, he spent 2 years in the School of Math at the Institute for Advanced Study in Princeton. His research and expertise are in modern AI and the intersection of mathematics, machine learning, and cognitive science.

Abstract

The goal of Exponentiating Mathematics (expMath) is to radically accelerate the rate of progress in pure mathematics through the development of an AI co-author capable proposing and proving useful abstractions. expMath will be comprised of teams focused on development of AI capable of autodecomposition and auto(in)formalization, and teams focused on evaluation with respect to professional-level mathematics. We will discuss the motivations of, logic behind, and ambitions for the program vis-à-vis state of the art in AI for mathematics.

Neuro-Symbolic Techniques for LLM-based Code Generation and Auto-Formalization of Proofs

Prithwish Jana
Georgia Institute of Technology

Prithwish Jana is a Ph.D. student in Computer Science at Georgia Tech, USA, advised by Prof. Vijay Ganesh. Before joining Georgia Tech in 2023, he spent a year (2022–2023) as a Ph.D. student at the University of Waterloo, Canada. Previously, he completed his M.Tech. (2022) and B.E. (2020), both in Computer Science and Engineering, from IIT Kharagpur and Jadavpur University, respectively. He was the department topper at both institutions, earning the Institute Silver Medal and University Gold Medal, respectively. Prithwish's research focuses on developing neuro-symbolic techniques to fine-tune large language models (LLMs), with an emphasis on bridging machine learning and automated reasoning. His work integrates symbolic reasoning tools and formal methods with LLMs, significantly enhancing their reasoning capabilities in fields such as software engineering (e.g., code translation and code generation) and mathematical reasoning (e.g., automated proof synthesis in LEAN and auto-formalization). More information about Prithwish can be found at https://sites.google.com/site/jprithwish/.

Vijay Ganesh
Georgia Institute of Technology

Dr. Vijay Ganesh is a Professor of Computer Science at the Georgia Institute of Technology. Prior to joining Georgia Tech in 2023, Vijay was a professor at the University of Waterloo in Canada from 2012 to 2023 and a research scientist at the Massachusetts Institute of Technology from 2007 to 2012. Vijay completed his PhD in Computer Science from Stanford University in 2007. Vijay's primary area of research is the theory and practice of SAT/SMT solvers, and their application in AI, software engineering, security, mathematics, and physics. In this context he has l ed the development of many SAT/SMT solvers, most notably, STP, Z3str4, AlphaZ3, MapleSAT, and MathCheck. He has also proved several decidability and complexity results in the context of first-order theories. More recently he has started working on topics at the intersection of learning and reasoning, especially the use of machine learning for efficient solvers, and the use of solvers aimed at making AI more trustworthy, secure, and robust. For his research, Vijay has won over 30 awards, honors, and medals to-date, including an ACM Impact Paper Award at ISSTA 2019, ACM Test of Time Award at CCS 2016, and a Ten-Year Most Influential Paper citation at DATE 2008. More information about Vijay can be found at https://vganesh1.github.io/.

Abstract

There is a growing trend of leveraging Large Language Models (LLMs) for formal language tasks such as code and proof generation, repair, and synthesis. As these tasks demand syntactic precision and logical consistency, there is increasing recognition that coupling LLMs with automated reasoning tools (such as SAT/SMT solvers, symbolic execution engines, and proof assistants) can significantly improve their effectiveness. Within this context, our talk explores neuro-symbolic approaches to enhance LLMs' complex reasoning capabilities by incorporating logical feedback from automated reasoning systems (e.g., SAT/SMT solvers, compilers, symbolic execution engines, and automated theorem provers) during fine-tuning. By establishing a corrective feedback loop between LLMs and these reasoning engines, we enable the models to generate initial abstractions that are verified by the engines, with the models subsequently refined through formal feedback. Further, this enables significantly smaller models to achieve far better reasoning capabilities. We have demonstrated this approach in practical software engineering tasks, such as code translation and code generation from natural language, as well as in mathematical reasoning tasks like auto-formalization and automated proof synthesis in LEAN.

In this talk, we present our work on AI for software engineering and mathematical reasoning. In the software engineering context, the talk will focus on LLM-based code translation (translating code between high-level programming languages). In our recent work (CoTran), we simultaneously fine-tune two LLMs, one for source-to-target and the other for target-to-source translation, with corrective feedback from a compiler and symbolic execution engine, assessing syntactic correctness and functional equivalence of the generated code. We will also discuss our LLM-based code generation approach, where a C++ code is generated from natural language pseudo-code using our Reinforcement Learning via Symbolic Feedback (RLSF) technique, which provides token-level feedback through poly-sized certificates identifying errors. In AI for mathematical reasoning, we will introduce an LLM+LEAN toolchain for auto-formalizing proofs in proof complexity theory. While many mathematical fields have been integrated into LEAN, there has been no dedicated effort to (auto-)formalize proof complexity within LEAN. This area is well-suited for auto-formalization due to its self-contained nature and the strong structural correspondence between informal and formal proofs (e.g., in bounded arithmetic). We will also explore fine-tuning LLMs for automated proof synthesis, aiming to generate formal proofs from conjectures. This neuro-symbolic theorem prover could potentially solve International Mathematics Olympiad (IMO)-level problems without human demonstrations, advancing LLMs in reasoning.

Overall, we demonstrate how neuro-symbolic methods enhance LLM reasoning and formalization in code and mathematics. By integrating feedback from automated reasoning systems during LLM fine-tuning for formal language tasks, our work advances both AI/ML verification and mathematical proof formalization, aligning with the 2025 HCSS conference themes of Formalizing Pure Math and Formal Verification of AI/ML Systems.

From Modal Logic to Computing: The Case for Hybrid Logics in FormMethods

William Harrison
Galois

Bill Harrison received his BA in Mathematics from Berkeley in 1986 and his doctorate from the University of Illinois at Urbana-Champaign in 2001 in Computer Science. From 2000-2003, he was a post-doctoral research associate at the Oregon Graduate Institute in Portland, Oregon. Until recently, Dr Harrison was on the faculty in the Electrical Engineering and Computer Science department at the University of Missouri. Currently, he is a Principal Scientist at Galois, Inc. He is the originator of the ReWire functional High-Level Synthesis language, a Haskell-embedded DSL for deisgning, implementing and verifying performant hardware. His research interests include all aspects of programming languages research (e.g., language-based computer security, semantics, design and implementation), reconfigurable computing, formal methods and malware analysis.

Abstract

1 Idaho National Laboratory, Idaho Falls, ID, USA
william.harrison@inl.gov
2 Cyber Manufacturing Innovation Institute (CyManII)
https://cymanii.org

There is a history of ideas and concepts migrating from Logic and Mathematics into Computing and, especially, Formal Methods has benefited from this intellectual technology transfer. Temporal logics, for example, originated in the 1950's in the work of the philosopher and logician, Arthur Prior [2], and, since Amir Pnueli's seminal work applying linear temporal logics to program verification [3] in 1977, temporal logics have become part of the standard toolbox for Formal Methods. In this talk, I will argue that hybrid logics [1] should take this trip, too. Although hybrid logic is perhaps not generally well-known now in Computing, I believe that its utility for Formal Methods will be clear to experienced researchers. Hybrid logic is a variety of modal logic with modalities that refer to individual states or worlds—these modalities are called indexicals. These special modalities give hybrid logic the ability to express fine-grained temporal properties beyond "vanilla" temporal logic [4, 6] for instance. In program analyses of all kinds, one frequently has intuitions of the form, "if the program execution reaches this program point, then P must have been true at this previous program point." Indexicals give you just the thing you need to formulate such intuitions. People familiar with incorrectness logic [7] may well have an A-ha! moment, if hybrid logic is new to them. I will illustrate ongoing research formally specifying of program weaknesses (in the sense of the Common Weakness Enumeration) using hybrid logic. I will present a formalization of hybrid logic in the Coq theorem prover in the style of Benzm¨uller and Paleo [5].

References

1. Patrick Blackburn, Maarten de Rijke, and Yde Venema. Modal Logic. Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.
2. Arthur N. Prior. Time and Modality. Oxford University Press, 1957.
3. Amir Pnueli. The temporal logic of programs. In Proceedings of the 18th Annual Symposium on Foundations of Computer Science, SFCS '77, pages 46–57, Washington, DC, USA, 1977. IEEE Computer Society.
4. P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. Logic Journal of the IGPL, 7(1):27–54, 1999.
5. Christoph Benzm¨uller and Bruno Woltzenlogel Paleo. Interacting with Modal Logics in the Coq Proof Assistant. In Computer Science – Theory and Applications, pages 398–411. Springer International Publishing, 2015.
6. F. Laroussinie, N. Markey, and P. Schnoebelen. Temporal logic with forgettable past. In Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS), pages 383–392, 2002.
7. Peter W. O'Hearn. Incorrectness Logic. Proc. ACM Program. Lang., 4(POPL), dec 2019.

KEYNOTE

Formal Verification of Financial Infrastructure with Imandra

Grant Passmore
Imandra

Grant Passmore is cofounder of Imandra Inc where he leads the design and development of the Imandra automated reasoning system. He is a widely published researcher in formal verification and symbolic AI whose contributions include advances in SMT solving (Z3), nonlinear proof procedures (MetiTarski, Imandra and Z3), and computational logic (Imandra). He earned his PhD on decision procedures for nonlinear arithmetic from the University of Edinburgh, is a graduate of UT Austin (BA in Mathematics) and the Mathematical Research Institute in the Netherlands (Master Class in Mathematical Logic), and is a Life Member of Clare Hall, University of Cambridge.

Abstract

Many deep issues plaguing today's financial markets are symptoms of a fundamental problem: The complexity of algorithms underlying modern finance has significantly outpaced the power of traditional tools used to design and regulate them. At Imandra, we have pioneered the application of formal verification to financial markets, and firms like Goldman Sachs, Itiviti and OneChronos rely upon Imandra's algorithm governance tools for the design, regulation and calibration of many of their most complex algorithms. With a focus on financial infrastructure (e.g., the matching logics of national exchanges and dark pools), we will describe the landscape and illustrate our Imandra system on a number of real-world examples. We'll sketch open problems and future directions along the way.

Bio: Grant Passmore is cofounder of Imandra Inc where he leads the design and development of the Imandra automated reasoning system. He is a widely published researcher in formal verification and symbolic AI whose contributions include advances in SMT solving (Z3), nonlinear proof procedures (MetiTarski, Imandra and Z3), and computational logic (Imandra). He earned his PhD on decision procedures for nonlinear arithmetic from the University of Edinburgh, is a graduate of UT Austin (BA in Mathematics) and the Mathematical Research Institute in the Netherlands (Master Class in Mathematical Logic), and is a Life Member of Clare Hall, University of Cambridge.

Welcome to the 25th Annual High Confidence Software and Systems Conference!

This year presents a milestone for the conference and the community of interest centered on assured engineering of software-intensive complex systems. The conference was initially born out of a desire to communicate the great work of our internal and external collaborating researchers to our stakeholders and transition partners. Those first conference years set the stage for technologies that would be core interests in continuing years, as well as for visionary ideas where substantial progress has since been made, and breakthroughs seem inevitable. For instance, within the first year of the conference, we saw an initial presentation by John Launchbury on Cryptol, a domain-specific language for cryptography, as well as a presentation by John Rushby relating the potential impact of automation towards "Disappearing Formal Methods." Regarding Cryptol, a handful of years later, we would see a presentation on "Proving Amazon's s2n Correct", and similarly towards the usability of formal methods, we have seen presentations over the years highlighting the strong push in that direction and leveraging into evolving ecosystems, experience, and increasing opportunities.

These early years also saw the herculean efforts behind the scenes by Tim Thimmesch, a trusted colleague and long-time friend, in getting us to a place to execute the conference.

After passing the 10-year mark, the HCSS Conference moved to a more traditional planning committee with co-chairs and a steering group, incorporating people, ideas, and insights that have moved the conference ahead wonderfully.

As the years have progressed, we've witnessed leading conversations at the HCSS Conference around a range of capabilities, from the lightest of weight to heaviest of methods. We've also seen the application of such capabilities to various domains and explored broad key topics for this community, such as constructive and analytic approaches to achieving high assurance, certification, proof engineering, and assuring AI.

Over the years, the conference has included notable invited speakers, too numerous to mention, spanning academia, industry, and government. We have also witnessed the loss of community members who were there with us at the start, Mike Gordon and Cordell Green, both true gentlemen and trusted colleagues who were terrifically generous to me personally.

Over the past 2 ½ decades, we have also seen an extension of the conference to an HCSS Conference Week to include workshops and more substantially co-located meetings, namely, the Software Certification Consortium and, more recently, the Trusted Computing Center of Excellence. Meetings focused respectively on certification and lowering the barrier to the adoption of the principled development and deployment of trustworthy systems.

Wonderfully community-building elements of the conference have included over the years the mainstay Chart House dinner, an evening pick-up card game, conference premiums, to include my favorites, yo-yos, gliders, and juggling balls, and terrifically engaging and substantive sidebar meetings with trusted colleagues.

Although very tactical at the onset, the conference has evolved into an extremely valuable and strategic vehicle for conversations of national interest and impact. One exciting and sustained attribute of the High Confidence Software and Systems Conference for me, at its inception 25 years ago and continuing today, is that of the meeting being a welcoming venue. It is a meeting where participants look forward to joining year after year. At the heart of this welcoming sustainability, not to mention the smoothness in coordination and execution of the meeting, is the Vanderbilt University team and its incomparable lead, Katie Dey.

Thanks for joining us for the 25th Annual High Confidence Software and Systems Conference—as always, I'm looking forward to the technical content, seeing old friends, and making new ones.

HCSS Conference Founder/Planning Committee Member
William B. Martin

Automated Reasoning for UAV Safety & Security: The DATUM Protocol Stack

Max Taylor
Idaho National Laboratory

Dr. Max Taylor is a formal methods scientist at Idaho National Laboratory (INL), where he is part of the Provably Secure and Resilient Systems group in collaboration with researchers from Carnegie Mellon University, the University of Pittsburgh, and the University of Central Florida. His research focuses on integrating formal methods with software engineering methodologies to improve the security of critical systems. Before joining INL, Dr. Taylor earned his PhD in Computer Science and Engineering from The Ohio State University. During his time at Ohio State, his research was funded by AFRL, his work won departmental research awards, and he received the departmental teaching prize.

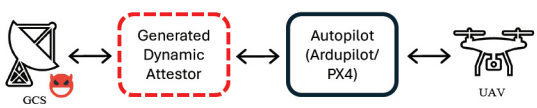
Abstract

Arthur Amorim1, Max Taylor2, Gary T. Leavens1
William L. Harrison2, Lance Joneckis2, and Trevor Kann3
1 University of Central Florida, Orlando, FL, USA
2 Idaho National Laboratory, Idaho Falls, ID, USA
3 Carnegie Mellon University, Pittsburgh, PA, USA

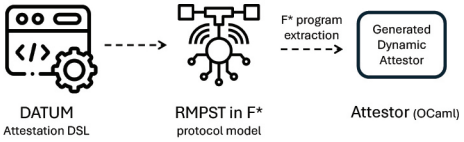
Stealthy attacks manipulate a system into an unsafe state by using operations that conform to the system's own protocols and, hence for obvious reasons, they are difficult to characterize and mitigate. In the context of UAVs, for example, a compromised ground control system (Fig 1a) may send perfectly legal commands that lead it to unsafe or destructive situations. Such attacks are challenging to characterize, because message interfaces can only define the individual messages but cannot express system designers' intentions for how they should be used. All manner of cyberphysicalsystems (e.g., industrial control systems) are vulnerable to such attacks.

The High Assurance Industrial Systems group at Idaho National Laboratories and our academic partners have developed a language-based approach [1] organized and driven by the F* (https://fstar-lang.org) theorem proving system. Our system—dynamically assured typed universal messaging (DATUM)—allows designers to define legal, safe message sequences as DATUM protocols and to then generate a dynamic attestor that enforces that protocol. DATUM (Fig. 1b) has a formalized semantics in F* in terms of refined multiparty session types (RMPST) [2,3,4]. Type-checking in the DATUM RMPST semantics (Fig. 1b) is fully automated with F*'s SMTsolver integration which is key to making the approach practical. The safety of DATUM protocols themselves can be validated in F* (i.e., that a protocol can only lead to safe states) and performant attestors are generated automatically via F* program extraction.

In this talk, we illustrate DATUM mitigation of safety and security issues in the MAVLink protocol commonly used to control UAVs [5], formalizing a substantial portion of the MAVLink protocol [6]. To our best knowledge, DATUM is the largest-scale application of RMPSTs to date. We also outline ongoing our ongoing efforts to integrate DATUM with HACMS-style [7] seL4-based systems. We illustrate the effectiveness countering stealth attacks with a software-in-the-loop Autopilot simulation.



(a) Interposed Attestor generated by DATUM notifies UAV in case of stealthy attack from compromised Ground Control System (GCS).



(b) DATUM protocols have RMPST semantics formalized in F* and is implemented in OCaml via F* program extraction.

Fig. 1: DATUM: DSL for high assurance attestation protocols to counter stealth attacks.

References

1. Prashant Anantharaman, Michael Locasto, Gabriela F. Ciocarlie, and Ulf Lindqvist. Building hardened internetof-things clients with language-theoretic security. In 2017 IEEE Security and Privacy Workshops (SPW), pages 120–126, 2017.
2. Fangyi Zhou, Francisco Ferreira, Raymond Hu, Rumyana Neykova, and Nobuko Yoshida. Statically verified refinements for multiparty protocols. Proceedings of the ACM on Programming Languages, 4(OOPSLA):1–30, 2020.
3. Fangyi Zhou, Nobuko Yoshida, and Iain Phillips. Refinement session types. PhD thesis, Master's thesis. Imperial College London, 2019.
4. Martin Vassor and Nobuko Yoshida. Refinements for multiparty message-passing protocols: specification-agnostic theory and implementation. 2024.
5. M. A. Hamza, M. Mohsin, M. Khalil, and S. Kazmi. MAVLink protocol: A survey of security threats and countermeasures. In 2024 4th International Conference on Digital Futures and Transformative Technologies (ICoDT2), pages 1–8. IEEE, 2024.
6. MAVLink Development Team. Mavlink: Micro air vehicle communication protocol, 2024. Accessed: 2024-11-19.
7. Kathleen Fisher, John Launchbury, and Raymond Richards. The hacms program: using formal methods to eliminate exploitable bugs. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 375(2104):20150401, 2017.

Modeling and Formal Analysis of High-Assurance Mixed-Reality Systems

Junaid Babar
Collins Aerospace

Dr. Junaid Babar is an expert in symbolic model checking and temporal logics and has extensive experience in their application to high-assurance systems. His journey into formal methods began at Intel when attempting to prove the correctness of distributed fault-tolerant applications deployed on highly parallel telecommunication servers. His PhD thesis at Iowa State University focused on model checking for Petri Nets and data-structures to the improve the scalability of CTL model checking, resulting in the introduction of ESRBDDs – canonical decision diagrams that combine the benefits of BDDs and ZDDs without any performance tradeoffs.

He is a Principal Systems Engineer at Collins Aerospace, where he has focused on transitioning formal methods into internal programs, and lent his expertise to DARPA programs such as CASE and PROVERS – programs aimed at making formal methods robust and accessible to developers of high-assurance systems such as those deployed by DoD and Collins. His recent efforts on ICS and ANSR aim to reason about systems combining symbolic reasoning with probabilistic and human behavioral models.

Abstract

Junaid Babar
Principal Engineer, Collins Aerospace
junaid.babar@collins.com

Isaac Amundson
Research Fellow, Collins Aerospace
isaac.amundson@collins.com

The past decade has witnessed rapid advancement of immersive system technologies, including mixed-reality safety-critical systems, in mission-critical aerospace and defense applications. However, design-time analyses of such high-assurance systems typically leave out one key ingredient: the user. Even when aspects of user behavior are included in the analyses, it is often assumed that the user will respond rationally to stimuli and always correctly perform the procedures necessary to achieve mission objectives. In high-assurance mixed-reality systems, the interactions between the user, system, and environment are entwined to such an extent that they must be modeled and analyzed together to ensure the entire human-machine system is protected against a variety of failure modes and vulnerabilities. This requires accurate models of human cognitive behavior as well as new formal analysis methods and tools that can provide the rigorous assurance needed for the safe and secure deployment to these systems.

The need for advanced analysis techniques for mixed-reality system designs has been recognized by the US Department of Defense, as evidenced by the DARPA Intrinsic Cognitive Security (ICS) program, which kicked off in July 2024. ICS is studying the feasibility of applying formal methods to the cognitive modeling domain to verify that users of tactical mixed-reality systems will be protected from adversarial cognitive attacks. This new class of attacks exploit the intimate connection between users and mixed-reality devices. On ICS, our team is developing the Modeling and Analysis Toolkit for Realizable Intrinsic Cognitive Security (MATRICS) – a framework for facilitating the development of provably secure mixed-reality systems.

Future immersive systems will likely find applications in everything from consumer electronics to high-assurance national security systems, and we must ensure rigorous analysis methods and evaluation criteria such as those being developed on ICS are established and matured in advance. In this presentation, we provide an overview of MATRICS as well as initial results demonstrating the feasibility of our approach.

Software Understanding for National Security (SUNS) Partnership Forum Report

Douglas Ghormley
SNL

Douglas Ghormley is a Senior Scientist at Sandia National Labs where he's worked for 27 years. Prior to Sandia, he received a BS in Computer Engineering from CMU in 1991 and a PhD in Computer Science from UC Berkeley in 1998. At Sandia, Doug has spent a career in cybersecurity research and development working a variety of challenges for different national security missions. His efforts have included researching novel tools for network protocol analysis, system forensics, agile embedded software architectures, malware analysis, malware signature discovery, source code analysis, vulnerability detection, and more.

Currently, his focus is on the cyber challenges posed by rapid adoption of third-party software into nearly every aspect of critical infrastructure and national security.Christopher Harrison is a Distinguished Member of Technical Staff at Sandia National Laboratories with a PhD in Computer Science from Auburn University. Christopher has spent his career primarily engaged in the research and development of software understanding tools for a variety of national security missions. Christopher has led R&D projects in the areas of virtual machine introspection, symbolic execution, abstract interpretation, model checking, transformers, ensembles of decision trees, counter-adversarial machine learning, fuzzing, and others. Christopher recently finished an assignment with CISA as a senior technical advisor to the Technical Director of the CISA/Cybersecurity Division.

Abstract

The widespread use of software that cannot be adequately characterized places society and government at unmeasurable risk. Software is embedded in countless systems responsible for providing U.S. critical infrastructure services that Americans rely on as well as systems providing national security capabilities. Every deterrent capability the U.S. possesses—ranging from conventional and nuclear forces to economic tariffs and sanctions—vitally depends on software. Earlier this year, four U.S. Government agencies collaborated on a report titled "Closing the Software Understanding Gap". Subsequently, the 2025 Software Understanding for National Security (SUNS) Partnership Forum (SPF-25) convened representatives from the government, academia, and industry to discuss the national need for improved software understanding, the technical and non-technical issues, and possible next steps. This talk will outline the software understanding challenges facing the nation and summarize key discussions from the SPF-25 event.

An Experiment Using Layered Attestation

Perry Alexander
The University of Kansas

Dr. Perry Alexander is the AT&T Foundation Distinguished Professor of Electrical Engineering and Computer Science and Director of the Institute for Information Sciences at The University of Kansas. His research interests include system-level modeling, formal verification, 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. Prior to joining KU, he was a faculty member in the Electrical and Computer Engineering and Computer Science department at The University of Cincinnati. Dr. Alexander has been Principal or Co-Principal investigator on over $38 million in research projects funded by various agencies including DARPA, NSA, NSF, AFRL, and NASA. Dr. Alexander has published over 130 refereed research papers and is author of System-Level Design using Rosetta published by Morgan Kaufman. He has won 23 teaching awards and was named the ASEE's Midwest Region Teacher of the Year in 2003.

Abstract

William Thomas[1], Logan Schmalz[1], Adam Petz[1], Joshua Guttman[2], and PerryAlexander[1]
[1]Institute for Information Sciences
The University of Kansas
[2]The MITRE Corporation

Remote attestation [1, 2] is the process of gathering evidence from a remote actor with the intent of establishing its trustworthiness. A relying party requests evidence from a target. The target responds by gathering allowable evidence and meta-evidence. Target evidence and meta-evidence is appraised to establish that the target is in a good operational state.

Growth in system complexity makes an attestation architecture with a single target impractical.
Any modern measurement target comprises many subsystems and depends on many others. Gathering evidence from a single component provides a limited picture of a measurement target. Instead we should treat attestation targets as collections of interacting, distributed components, gathering and composing evidence for entire systems. Layered attestation is an enhanced attestation process where attestation managers execute protocols that perform multiple component measurements and bundle resulting evidence for appraisal.

The MAESTRO tool suite provides a mechanism for building layered attestation systems around the execution of Copland protocols [3, 4]. Users specify a protocol to be executed and MAESTRO configures a common attestation manager core and attestation service providers to execute that protocol on a target system. With few exceptions, MAESTRO components are either formally verified or synthesized from formal specifications providing assurance that protocol execution faithfully implements the Copland semantics.

Our presentation covers the first significant empirical study of MAESTRO layered attestation. MAESTRO systems have until now been evaluated analytically focusing on their formal semantics limited to small systems. In this demonstration we use MAESTRO tools to generate an attestation system and run experimental attacks to determine its effectiveness.

The target is a classical cross-domain system where messages move from high side to low side. The application is a four-stage pipeline that consumes high-side messages, rewrites message content, filters message recipients, and delivers messages to a low-side recipient. The layered attestation system measures configuration files and software comprising the cross domain system in conjunction with measuring the operational state of its Linux environment and ensuring that correct components boot at the correct time during system startup. The entire attestation system is rooted on a TPM that ensures strong binding of the attestation signing key to the attestation manager.

The presentation will focus on design decisions and trade-offs, simulated attack results, and the system-level trust argument. It will compliment earlier presentations on MAESTRO's design and formal semantics. 2 F. Author et al.

Bibliography

[1] Coker, G., Guttman, J., Loscocco, P., Herzog, A., Millen, J., O'Hanlon, B., Ramsdell, J., Segall, A., Sheehy, J., Sniffen, B.: Principles of remote attestation. International Journal of Information Security 10(2), 63–81 (June 2011)
[2] Haldar, V., Chandra, D., Franz, M.: Semantic remote attestation – a virtual machine directed approach to trusted computing. In: Proceedings of the Third Virtual Machine Research and Technology Symposium. San Jose, CA (May 2004)
[3] Petz, A., Thomas, W., Fritz, A., Barclay, T.J., Schmalz, L., Alexander, P.: Verified configuration and deployment of layered attestation managers. In: Software Engineering and Formal Methods: 22nd International Conference, SEFM 2024, Aveiro, Portugal, November 6-8, 2024, Proceedings. pp. 290–308. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-77382-2 17
[4] Ramsdell, J., Rowe, P.D., Alexander, P., Helble, S., Loscocco, P., Pendergrass, J.A., Petz, A.: Orchestrating layered attestations. In: Principles of Security and Trust (POST'19). Prague, Czech Republic (April 8-11 2019)

Formalizing and Automating the Discovery of Weird States and Machine Primitives for High-Confidence Software

Meera Sridhar
University of North Carolina Charlotte

Dr. Meera Sridhar is an Associate Professor in the Department of Software and Information Systems at UNC Charlotte. Dr. Sridhar received her Ph.D. in computer science from the University of Texas at Dallas and her B.S. and M.S. degrees in computer science from Carnegie Mellon University. Dr. Sridhar has more than 20 years of experience in software and systems security, language-basedsecurity, formal methods, cyber-physical systems security, and cybersecurity education. Her research is funded by the National Science Foundation, the NC General Assembly, and NSA; her work has been published in top security and formal methods venues.

Dr. Meera Sridhar serves as the Director of the Center for Energy Security and Reliability (CESAR), a state-funded, collaborative initiative between UNC Charlotte, North Carolina State University, and North Carolina A&T University. In this role, she oversees research and education efforts focused on protecting critical energy infrastructure and developing resilient, sustainable power grids. Her leadership includes fostering partnerships with industry and government, securing research funding, and guiding workforce development to prepare future professionals in energy security.

As Director of the external-facing Smart Home IoT Lab, Dr. Sridhar not only advances research on IoT security for home and urban environments but also supports faculty and students in research, teaching, and outreach in IoT. The lab actively engages in outreach programs for K-12 schools, community colleges, and underrepresented groups, helping to broaden participation and awareness in IoT security and technology fields.

Abstract

Modern software systems are increasingly complex, leading to the emergence of unexpected behaviors and vulnerabilities. This talk delves into a novel approach to formally analyze and identify these vulnerabilities, focusing on "weird states" and "weird machine primitives." Weird states are unusual program states that deviate from expected behavior, while weird machine primitives are instructions or operations that can create or exploit these states.

We present a formal framework based on program semantics to model and analyze program behavior, enabling the precise identification of discrepancies and potential vulnerabilities. This framework leverages both axiomatic and operational semantics to capture the intricate interactions between instructions and data within a program.

Through case studies, including the NSO FORCEDENTRY exploit, we demonstrate the effectiveness of our approach in uncovering hidden vulnerabilities. By understanding the underlying principles of weird states and machine primitives, we can develop more robust security measures and mitigate the risks associated with complex software systems.

This research aligns with the HCSS conference's focus on high-confidence software and systems by providing a rigorous and formal approach to ensure the reliability and security of complex software systems. A particularly relevant application of our approach lies in the domainof AI-generated code. As AI models become more sophisticated, they can generate code that is both innovative and error-prone. Our formal methods can be used to analyze AI-generated code for potential vulnerabilities, such as buffer overflows, memory leaks, and security breaches. By identifying and addressing these vulnerabilities early in the development process, we can significantly improve the quality and security of AI-generated software.

This talk will provide a clear and accessible overview of our approach, emphasizing its practical implications for enhancing software security. We will discuss how to apply these techniques in real-world scenarios, and how to integrate them into existing security development lifecycles using automated model checking and reasoning processes to systematically identify weird states and machine primitives in large-scale software systems.
By automating the process of vulnerability discovery, we can further enhance the reliability and security of software systems. We are particularly interested in leveraging advances in automated reasoning and formal verification to develop tools that can automatically analyze AI-generated code for potential vulnerabilities, ensuring the safety and security of these increasingly complex systems.

ModelForge: Using AI to Improve Security Protocols

Martin Duclos
Mississippi State University

Martin Duclos is a Ph.D. candidate in Computer Science at Mississippi State University, where his research focuses on applying artificial intelligence and machine learning to cybersecurity challenges, including cryptographic protocol analysis and malware detection. He currently serves as a Director of Digital Transformation within the Defense Industrial Base, participating in strategic technology initiatives that support national security and defense applications. With a career in information technology that began in 2000, Martin brings deep experience spanning the private sector, academia, and, most recently, the Defense Industrial Base. His background includes work in cloud computing, network security, and software engineering. He holds both undergraduate and graduate degrees from Mississippi State University.

Abstract

Martin Duclos
Mississippi State University
md128@msstate.edu

Sudip Mittal
Mississippi State University
mittal@cse.msstate.edu

Edward Zieglar
National Security Agency
evziegl@uwe.nsa.gov

ModelForge automates portions of the translation process of protocol specifications into CPSA models, simplifying formal analysis in the development of security protocols. Using a fine-tuned large language model and customized prompts, it addresses the IETF's call for accessible verification tools and encourages broader adoption by developers.

1 Introduction & Background

As the complexity and usage of internet protocols continue to grow, ensuring their security and accuracy has become a daunting challenge. To address this, the Internet Engineering Task Force (IETF) has encouraged protocol developers to integrate formal analysis and validation into their development lifecycle [1]. However, formal analysis is complex and requires specialized expertise, making it inaccessible to many developers [2]. This hinders the ability to verify the stated security properties of a protocol, creating a barrier to the broader adoption of formal methods.

One approach to formal analysis, as described by Meadows [3] and initially proposed by Kemmerer [4], is to model a protocol in a formal language. However, translating protocol specifications into models compatible with formal methods tools, such as the Cryptographic Protocol Shapes Analyzer (CPSA), requires domain expertise and manual effort. This reliance on expertise and manual effort often leaves protocols without formal proofs, complicating security verification in the IETF standards process.

To address these challenges, we support the IETF's efforts by simplifying key tasks in formal protocol analysis and validation. Our contributions include: 1) A prototype to automate a portion of the formal analysis process, 2) Enhancing the accessibility of CPSA to a broader audience of users, and 3) Improving the security of future protocols by reducing the effort and expertise required for formal analysis.

This abstract and proposed presentation introduces ModelForge, a generative AI tool that automates portions of the process for converting protocol specifications into CPSA models. By reducing manual effort and reliance on domain expertise, ModelForge streamlines formal protocol analysis and improves accessibility.

2 ModelForge

ModelForge automates portions of the process for translating protocol specifications into CPSA models, reducing the need for manual effort and domain expertise in formal analysis. As shown in Figure 1, its architecture includes pre-processing and interactive components.

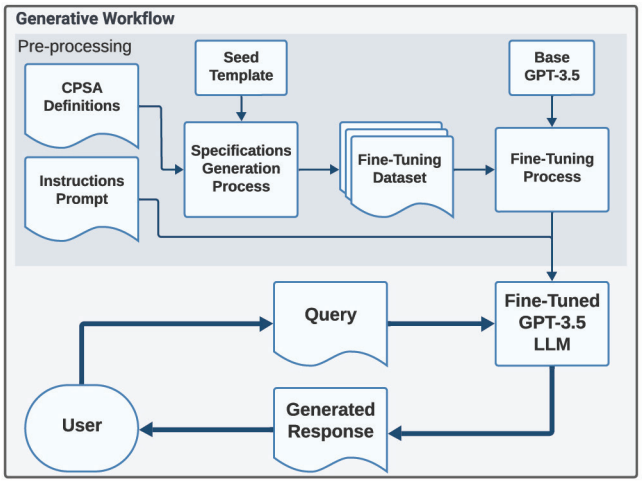


Figure 1. ModelForge Architecture

The shaded areas highlight preprocessing steps, such as fine-tuning the LLM with real and synthetic CPSA models and custom prompts. The lower half of the diagram illustrates the data flow, where user-provided protocol specifications are processed by the fine-tuned LLM to generate CPSA models.

Using generative AI, ModelForge reduces the need for specialized expertise and manual effort, improving the efficiency and accessibility of formal analysis for protocol developers.

3 Conclusion

In this abstract we introduced ModelForge, an improved domain-specific LLM translator designed to translate protocol specifications into CPSA models. Building on prior work [5], ModelForge was evaluated against other LLMs in generating CPSA protocol definitions from natural language. Expert assessments of correctness, clarity, and completeness showed consistently good performance, making it a promising tool for automating CPSA model generation.

References

[1] IETF. Usable Formal Methods Proposed Research Group (UFMRG). https://datatracker.ietf.org/doc/charter-irtf-ufmrg/01/, January 2023.
[2] Stefanos Gritzalis, Diomidis Spinellis, and Panagiotis Georgiadis. Security protocols over open networks and distributed systems: Formal methods for their analysis, design, and verification. Computer Communications, 22(8): 697–709, 1999.
[3] Catherine Meadows. Applying formal methods to the analysis of a key management protocol. Journal of Computer security, 1(1):5–35, 1992.
[4] RA Kemmerer. Using formal methods to analyze encryption protocols. IEEE J. Select. Areas Commun., 7(4):448–457, 1989.
[5] Martin Duclos, Ivan A. Fernandez, Kaneesha Moore, Sudip Mittal, and Edward Zieglar. Utilizing large language models to translate rfc protocolspecifications to cpsa definitions, 2024.

Secure Protocols via LLMs and CPSA

Lauren Brandt
MITRE

Lauren Brandt holds an M.A. in Mathematics from Arizona State University and is currently a cybersecurity engineer at MITRE. Her research focuses on cryptography, formal methods, and AI for cybersecurity. She's especially passionate about the moments where these fields intersect and enjoys tackling the complex challenges that arise there. Currently, Lauren leads a research effort exploring how generative AI can improve the usability of formal methods tools for cryptographic protocols. She's excitedto share insights into how this integration can help overcome barriers in protocol design ytand analysis.

Abstract

Lauren Brandt, Dr. Andres Molina-Markham, and Dr. Joshua Guttman
MITRE, Bedford, MA

Cryptographic protocols define the rules for securely transmitting messages, specifying what information to include and what to sign or encrypt to ensure authenticity and confidentiality. Developing these protocols is a complex and error-prone process, typically requiring advanced tools and significant expertise. This work integrates Large Language Models (LLMs) with MITRE's Cryptographic Protocol Shapes Analyzer (CPSA), a formal methods tool for analyzing cryptographic protocols, to demonstrate how AI can increase the usability of such tools and enhance the protocol design process. The integration combines the efficiency of LLMs with the reliability of CPSA: LLMs simplify complex CPSA tasks, while CPSA validates the LLM-generated results. This system leverages the strengths of LLMs to improve workflows with minimal risk. Even when LLM outputs are imperfect, CPSA serves as a reliable ground truth, enabling practical and trustworthy applications in the cybersecurity domain.

CPSA takes a protocol written in a Lisp-like format as input. It outputs a list of all the minimal, essentially different executions that a strong malicious adversary could cause. We call these "shapes." Converting protocols to CPSA's input format and interpreting its shape-based output can be challenging. This work aims to integrate LLMs alongside CPSA to automate these conversions. LLMs will translate protocols from natural language (NL) into CPSA's input format and interpret CPSA's shape-based analysis back into NL, thereby simplifying the user experience and reducing manual effort. Even if the LLMs generate imperfect translations, CPSA's analysis will highlight any flaws, allowing users to easily verify and adjust the protocol based on the shape analysis.

To achieve this, we determined the optimal NL format, created a dataset, defined a scoring method, and conducted experiments to both maximize and measure the effectiveness of each LLM in performing these translations. Using a LLM as a translator from NL to CPSA has proven highly effective, with GPT-4o achieving an impressive 97.7% average accuracy across 89 protocols using only two example translations. Many of these protocols were translated perfectly. Research on translating CPSA outputs back into NL is ongoing, with the goal of having the LLM provide simple targeted suggestions based on CPSA analysis.

By significantly enhancing the usability of complex tools like CPSA, LLMs have the potential to transform the formal methods community, reducing user burden and enabling a more interactive and iterative protocol design process. Developers can now create, analyze, and refine protocols more efficiently, identifying flaws and areas for improvement with greater ease. Beyond formal methods, this approach has broader implications for the cybersecurity and AI fields. It offers a practical way to integrate AI into cybersecurity tools, enhancing their accessibility and usability while maintaining the essential high level of trust. Even when LLM outputs are not fully accurate, CPSA acts as a reliable ground truth, allowing for easy validation and confirmation of the LLM-generated results.

KEYNOTE

Neurosymbolic Programming and the Path to Safe AI

Armando Solar-Lezama
MIT CSAIL

Professor Armando Solar-Lezama is Associate Director and COO of MIT CSAIL, where he leads the Computer-Aided Programming Group. His research focuses on reducing the skill and effort required to develop secure, reliable, and efficient software. One of his research group's central contributions to this goal is the development of new approaches to software synthesis that can combine information from different sources to produce the code that the programmer wants. The group's research ranges from the design of new analysis techniques and automated reasoning mechanisms to the development of new programming models that automate challenging aspects of programming.

Prof. Solar-Lezama is most interested in software synthesis and its applications to particular program domains such as high-performance computing. He first found this niche area of program synthesis as a graduate student at Berkeley, for which his thesis project, a language called Sketch, treats program synthesis as a search problem in which the algorithms pare down the search space to make the search faster and more efficient. Since then, program synthesis research has greatly expanded into the active field it is today.

Abstract

Machine learning and AI techniques have made enormous strides in the past decade, allowing us to build systems that were considered science fiction only a few years ago. However, safety and reliability remain serious obstacles in many application areas. In this talk, I will describe how the combination of deep learning with techniques developed for program analysis and synthesis, neurosymbolic programming, can help address some of the gaps of more traditional learning-based approaches.

Bio: Armando Solar-Lezama is a Distinguished Professor of Computing at the MIT Schwarzman College of Computing and Associate Director and COO of the Computer Science and Artificial Intelligence Lab at MIT (CSAIL). Prof. Solar-Lezama is best known for his pioneering work on program synthesis. More recently, his work focuses on the intersection of machine learning and programming technology, with a special focus on leveraging ideas from programming systems to improve learning. He currently leads the NSF Expeditions project "Understanding the World Through Code" which aims to develop neurosymbolic learning techniques to support scientific discovery.
Prof. Solar-Lezama is the recipient of the 2024 Robin Milner Young Researcher Award from ACM SIGPLAN.

Reconciling Distributed System Implementation and Design with Neuro-Symbolic Reasoning

Tristan Ravitch
Amazon Web Services

Tristan Ravitch is a Principal Applied Scientist in AWS. He has been building program analysis and verification tools for 15 years with a focus on improving confidence in legacy systems. He has worked on binary analysis, static dataflow analysis, bounded model checking, and architectural recovery applied to systems ranging from microcontrollers to distributed cloud systems. While his passion is proving systems correct, Tristan has recently been working on automatically generating high-fidelity models to improve the connection between code and system models.

Abstract

In this talk, we will report on our experience at AWS combining Automated Reasoning and AI techniques (neuro-symbolic reasoning) to accelerate security and compliance assessments for distributed systems. These assessment activities require understanding the intended behavior of a system, its actual behaviors, and how those relate to a changing environment (e.g., regulatory requirements). While a future where all major systems are built and maintained using Model-Based Systems Engineering (MBSE), that is not our reality today. The intended behaviors of a system are captured in design artifacts. In most cases, those design artifacts diverge from the reality of the system as it evolves.

To understand the current behavior and implementation of our systems, we use scalable Automated Reasoning techniques to recover the architecture of systems as-deployed and to map the flow of data throughout the system. While this analysis effectively captures the current behavior of the system, it is difficult to automatically relate the observed system behaviors back to the intent of the system as represented by the original design artifacts because 1) the state of the system has drifted from the original design, and 2) the design artifacts were never intended to be machine readable.

We bridge the gap by using Generative AI to process design artifacts and automatically relate them to our view of our systems' behaviors as inferred by Automated Reasoning. We will report on the types of prompting we used for this process and relate our success using man small and targeted prompts. The talk will explore an example of automating the collection of evidence for compliance with data sovereignty policies, which introduces challenges like categorizing data schemas based on criteria that did not exist when the original design artifacts were constructed (e.g., due to new categories of data introduced by regulation). We will describe early evaluation of the results with service owners. We also explore how this process will work for periodic re-assessments and how this neuro-symbolic approach can bootstrap a more systematic and sustainable automated assessment regime.

Translating C to Rust: Better, cheaper, faster

Per Larsen
Immunant, Inc.

Per Larsen is the CEO of Immunant, Inc., a firm he co-founded to make legacy code safer through migration or mitigation. The team at Immunant has made practical contributions to exploit mitigation and memory safety, including the development of the C2Rust migration framework. Immunant's work is used by multiple branches of the US military and several large enterprises.

Per got his Ph.D. in Computer Science from the Technical University of Denmark in 2011. Following a decade-long tenure as a Postdoctoral Scholar at UC Irvine, Dr. Larsen continues to bridge the gap between academic research and practical security solutions through his leadership at Immunant.

Abstract

Many of the most important systems in the world are written in unsafe languages such as C. It would be beneficial to rewrite them in safe-by-design languages, such as Rust, but migration of a real system by hand is enormously expensive. This is why, for almost a decade, Galois and Immunant have been developing `c2rust`, an automatic migration tool for C systems code. `c2rust` has been widely used — for example, the popular serde_yaml Rust crate wraps c2rust-transpiled code.

Our original `c2rust` transformed C code into unsafe, C-like Rust code. This is the first step of migration, but ultimately, we want Rust that is safe, performant, and idiomatic. The current state of the art is to run c2rust, and then for an expert team to migrate the rest of the way. Immunant have done exactly this on the migration of the dav1d codec library to Rust.

In this talk, I will discuss our work extending `c2rust` to support more of the migration pipeline. I will discuss the properties of C and Rust that make migration between these languages difficult. I will describe our new migration tool, which uses static and dynamic analysis to migrate some, but not all, C code patterns. Finally, I will explain what difficulties remain and show examples of how modern AI can further drive down costs of code migration.

Functors as Bridges Between AI-Generated Code and Formal Models for High-Confidence Software Systems

Sumit Jha,
Florida International University, Miami

Dr. Sumit Kumar Jha is Eminent Scholar Chair Professor of Computer Science at Florida International University. He earned his Ph.D. in Computer Science from Carnegie Mellon University and has held multiple summer faculty appointments with the Air Force Research Laboratory Information Directorate.

Dr. Jha has led interdisciplinary, multi-institutional teams on projects funded by the National Science Foundation (NSF), Defense Advanced Research Projects Agency (DARPA), Department of Energy (DOE), and other federal agencies. His work has been published at premier venues, such as AAAI, DAC, DATE, ICCAD, ICLR, IJCAI, and NeurIPS.

His research focuses on building high-assurance and efficient AI at both algorithmic and hardware levels. He has developed methods to make AI systems more transparent and resistant to threats by combining symbolic decision procedures, human expertise, foundation models, and deductive reasoning. Dr. Jha has also pioneered flow-based in-memory computing techniques for data-intensive applications, advancing a new paradigm for efficient and sustainable AI in his research funded by the NSF over the past 11 years.

Dr. Jha has been a PI on projects from NSF Software and Hardware Foundations (SHF), NSF Exploiting Parallelism and Scalability (XPS), NSF Scalable Parallelism in the Extreme (SPX), and NSF Formal Methods in the Field (FMitF), NGA Boosting Innovative GEOINT, NNSA/ORNL, ONR Science of AI, DARPA GARD, DARPA ANSR, DARPA TIAMAT, Department of Energy, AFRL, and the Royal Bank of Canada Innovation Lab. His work has earned multiple best paper awards and nominations at various forums (IEEE DATE, ACM/IEEE ICCAD, IEEE MILCOM, IEEE ICCABS), as well as the prestigious Air Force Office of Scientific Research Young Investigator Program (AFOSR YIP) Award.

Abstract

Computer Science Department, Florida International University, Miami, FL, USA

Introduction: Large Language Models (LLMs) have demonstrated remarkable capabilities in generating code from natural language prompts. However, their integration into safety- and mission-critical software remains challenging due to their lack of formal guarantees.

Technical Approach: To address this gap, we propose a co-synthesis framework that enables LLMs to generate (i) code, (ii) its formal verification model, and (iii) functor mapping the code to the formal model, in parallel. The core idea is to employ functors to establish a structured mapping between code constructs and their corresponding formal representations. The process begins with the LLM generating code based on a formal specification. Simultaneously, the LLM produces a formal model in the Symbolic Model Verifier (SMV) language, which captures the key state transitions and temporal logic properties. To ensure consistency between the generated code and model, a functor is also constructed by the LLM to provide a systematic mapping between these two representations, allowing for human-auditable verification. Model checking is then employed to verify the formal model against temporal logic specifications, such as safety and liveness properties. If any violations are detected, counterexamples are analyzed, and feedback is fed back into the LLM to refine both the code and the model iteratively. This feedback loop continues until formal properties are satisfied, minimizing the risk of errors.

Experimental Validation: We evaluate our approach on the classic Dining Philosophers problem, a well-established benchmark in concurrent computing. The results demonstrate that GPT-4o required 2 iterations for code synthesis and 4 for model correction, Claude-3.5 Sonnet succeeded in a single iteration, and Llama-3.1 405B struggled, requiring 15+ iterations with significant human intervention. We verified key correctness properties using the NuSMV model checker, confirming that no two adjacent philosophers enter the eating state simultaneously, every philosopher eventually progresses to eating without starvation, and there are no deadlocks.

Conclusion and Future Work: This work advances AI-assisted software engineering by introducing co-synthesis of functors using LLMs. This integration of category theory ensures human-auditable alignment between generated code and its formal representation. Future efforts will focus on (i) automating functor validation to reduce human oversight, (ii) scaling the approach to larger and more complex distributed systems, and (iii) enhancing AI model robustness to improve consistency in formal model generation. By combining AI-driven synthesis with functors and rigorous verification, we aim to bridge the gap between software implementation and formal assurance, paving the way for safer, more reliable software systems.

References

[1] S.K. Jha, S. Jha, R. Ewetz, and A. Velasquez, "Co-Synthesis of Code and Formal Models Using Large Language Models and Functors," IEEE Military Communications Conference (MILCOM), Oct 2024.
[2] A. Nunez, N.T. Islam, S.K. Jha, and P. Najafirad, "A Multi-Agent Framework for Securing LLM Code Generation through Static Analysis and Fuzz Testing," Workshop on Safe & Trustworthy Agents (SATA) at NeurIPS, 2024.
[3] S.K. Jha, S. Jha, R. Ewetz, and A. Velasquez, "Solving Mystery Planning Problems Using Category Theory, Functors, and Large Language Models," 3rd International Conference on Assured Autonomy (ICAA), Oct 2024.

Secure AI Through Verification, Transparency, and Fairness

Jessica Inman
GTRI

Dr. Jessica Inman is Division Chief of the Assured Software and Information Division (ASID) at the Georgia Tech Research Institute (GTRI) and a Senior Research Scientist in the Cybersecurity, Information Protection, and Hardware Evaluation Research (CIPHER) laboratory. She leads a research portfolio focused on low-TRL research to assure DoD algorithms, software, information, and communications, with a personal research focus on artificial intelligence and machine learning (AI/ML) security and applied AI/ML for cybersecurity.

Abstract

Artificial Intelligence (AI) and Machine Learning (ML) have shown incredible promise in a wide variety of domains. However, to deploy AI-enabled systems in high-risk domains such as cybersecurity, autonomous control, or medical workflows, we must consider the security of AI/ML algorithms as a component of an AI-enabled system. To address this, we will explore metrics, tools, and techniques for quantifying and enhancing AI/ML model robustness, trust, verification, transparency, and fairness. By applying software development paradigms, we will discuss how to integrate these approaches into AI/ML development and deployment pipelines for Secure AI. Furthermore, we will examine the extension of these paradigms to large language models (LLMs) for AI model assessment and LLM workflow evaluation.

Automated SysML v2 System Model to Memory-Safe Language Code Generation with Integrated AI Assistance

David Hardin
Collins Aerospace

David S. Hardin has made contributions in the areas of formal methods, computer architecture for High Assurance systems, as well as memory-safe programming languages. He is currently Chief Technologist for Trusted Methods in the Applied Research and Technology organization at Collins Aerospace, where he has served as Principal Investigator or Researcher for several U.S. DoD Research and Development programs. Most recently he has contributed to the DARPA PROVERS and CASE programs. 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. He is author or co-author of more than 80 peer-reviewed publications, and is an inventor or co-inventor on 15 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. In 2023, he was inducted into the Woodford County, Kentucky Public Schools Hall of Fame.

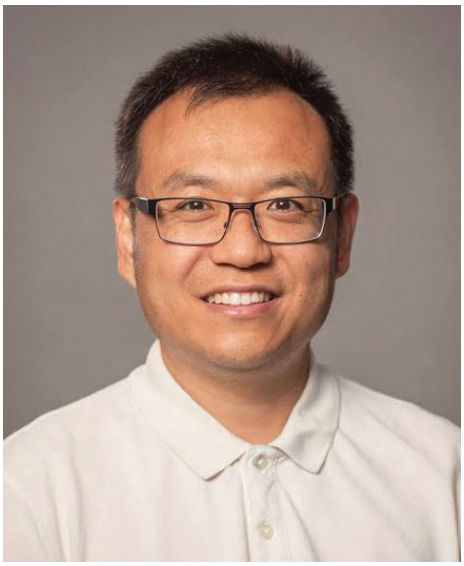
Abstract

Authors: Isaac Amundson, Junaid Babar, Jason Belt, Darren Cofer, David Hardin, Saqib Hasan, John Hatcliff, Robby, Amer Tahat

One of the biggest challenges of Model-Based Systems Engineering (MBSE) is providing allocation to/traceability from the corresponding software and hardware design, particularly ensuring that the system model and design/implementation remain "in-sync" during product development, as well as ensuring that assurance properties established at the system model level hold at the design/implementation level. As part of the DARPA PROVERS program, we are developing a revolutionary MBSE environment that allows non-specialist developers to specify models in the SysML v2 systems modeling language, automatically generate skeletal implementations of those models in a memory-safe language (Rust), as well as state and prove properties about the system model that can be refined and reproved against the generated design. These systems can be selected to be hosted on either Linux or the verified seL4 microkernel. The latter provides mathematically proven separation between threads, thus providing a very high assurance foundation upon which to produce application-level property proofs. We support memory safe code generation in Rust, including the Verus Rust annotation and verification system, as memory-safe languages avoid many common vulnerabilities that are the bane of C/C++ development. Memory-safe languages are increasingly required for high-assurance development, and our ability to automatically generate code in a memory-safe language from a system model helps to produce complete systems, even when memory-safe language developers are scarce. Formal property specification and proof can be difficult for non-specialists to "get right", so we provide aids to the developer at the model and design levels in the form of neuro-symbolic property specification assistants. These assistants utilize generative AI techniques to aid the developer in the composition of properties from natural language specifications, coupled with vetting of those property specifications via mathematically-precise formal analysis tools. This vetting eliminates any "hallucinatory" generative AI output. Additionally, the AI assistants can serve to interpret the sometimes obscure counterexamples that model checkers generate in natural language, as well as to suggest means to eliminate those counterexamples. In this talk, we will present our SysML v2-based toolchain, demonstrate its code generation capability on a simple system example, and demonstrate its property specification and AI-assisted proof capability, all in the context of a Continuous Integration/Continuous Development (CI/CD) framework familiar to developers.

LLM-enabled Software Testing

Fanxin Kong
University of Notre Dame



Dr. Fanxin Kong is an assistant professor in the Department of Computer Science and Engineering at University of Notre Dame. Before that, he worked as an assistant professor at Syracuse University and as a postdoctoral researcher at University of Pennsylvania. He obtained his Ph.D. in Computer Science at McGill University. Dr. Kong's research includes security/safety/assurance, and machine learning/large-language models/foundation models for Cyber-Physical Systems. He has published over 80 research papers at highly reputable venues. His research is supported by NSF, AFRL, AFOSR and DARPA, and receives multiple awards. He received NSF CAREER Award in 2025.

Weizhe Xu
University of Notre Dame

Weizhe Xu is a third-year Ph.D. student in Computer Science and Engineering at the University of Notre Dame, advised by Dr. Fanxin Kong. He received his Master's degree from the University of Manchester, UK, and his Bachelor's degree from Tongji University, China.
His research focuses on the intersection of safety and security in cyber-physical systems (CPS), with particular interest in formal methods and runtime assurance.
He is also exploring the integration of large language models (LLMs) into safety-critical systems.

Abstract

Traditional unit test generation techniques leverage search-based, constraint-based, or random-based strategies to generate a suite of unit tests with the main goal of maximizing the coverage in the software under test. However, the coverage and meaningfulness of the generated tests are still far from satisfactory. Large Language Models (LLMs), deep learning architectures typically comprising over 10 billion parameters, have recently been utilized for test generation. Contemporary LLM-enabled test generators demonstrate remarkable potential. However, while existing LLM-based test generation solutions exhibit impressive results on small code snippets, they face significant challenges in handling large-scale, real-world software projects.

A key difficulty arises from token limitations. When facing large-scale code, users cannot include all the necessary code in a single prompt due to the inherent token limitation of the LLMs, and the amount of code that LLMs can output at one time is also restricted. Meanwhile, the efficiency of the prompt also needs to be considered. Some approaches attempt to include as much information as possible in the prompt. However, in practice, irrelevant information in the prompt can lead to a decline in LLMs' performance. Therefore, it is necessary to enhance the efficiency of prompts to avoid misleading the LLMs and producing incorrect results. Moreover, context retention becomes paramount in multi-turn Q&A scenarios. As the number of interactions grows, summarizing previous outputs and referencing them effectively allows LLMs to maintain comprehensive knowledge of the overall codebase without reintroducing extensive, repeated content.

By targeting these challenges, i.e., token limitation, prompt engineering, and context retention, our research seeks to provide a scalable and accurate LLM-enabled test generation framework for real-world, large-scale software. Specifically, our solution consists of three primary steps. First, we employ program analysis methods to segment the code into smaller, logically coherent snippets, ensuring that each query to the LLM remains within token limits while retaining relevant sections for test generation. Second, we develop a reasonable summarization technique that compresses less critical information from prior interactions, ensuring that the LLM retains a global understanding of the entire project without being overwhelmed by excessive details. Finally, we design benchmarks to evaluate the performance of LLM-based approaches across multiple dimensions, such as object-oriented and procedural programming styles, reflecting the diverse requirements of large-scale software development.

Towards Trustworthy Integration of Generative AI in the MBSE Development Lifecycle

Amer Tahat
Collins Aerospace

Dr. Amer Tahat is a Senior Engineer at Collins Aerospace/ RTX. Dr. Tahat specializes in formal methods, Orgnizational GenAI, multimodal and LLMs'safety, security, and reliability. His current research aims to build reliable LLM-based solutions, such as high-assurance coding copilots for aerospace critical systems. His previous roles include Assistant Research Professor at Pennsylvania State University (PSU) in the College of Information Sciences and Technology and Assistant Research Professor in Electrical and Computer Engineering at Virginia Tech University. Dr. Tahat earned his Ph.D. in Computational Science and Engineering from the Computer Science Department at Michigan Technological University, with his doctoral research co-supervised by NASA Langley Research Center, focusing on the safety of aerospace critical system developments.

In addition, he has completed postgraduate programs in Advanced Cybersecurity from Stanford University, Cloud Computing from the University of Texas at Austin, and a Data Science and AI PGE from MIT. Dr. Tahat has worked on projects supported by several agencies, including DARPA (Defense Advanced Research Projects Agency), the Office of Naval Research (ONR) served as a Co-PI, and the National Science Foundation (NSF).

Abstract

Authors: Amer Tahat, Isaac Amundson, David Hardin, and Darren Cofer.

Copilot technology is reshaping systems development, yet incorporating generative AI into a formally verifiable Model-Based Systems Engineering (MBSE) development lifecycle raises crucial questions about trust, abstraction fidelity, and usability. Although formal verification is indispensable for ensuring mission-critical properties—particularly in aerospace and defense—difficulties in interpreting complex formal verification artifacts such as counterexamples and keeping models synchronized with code remain persistent barriers. AI-driven copilot technology offers a path to alleviating these obstacles, but it must be carefully designed to uphold logical soundness, compliance, and user confidence across all stages of system development.

In this talk, we present AGREE-Dog, a cross-platform, generative AI copilot for the Assume-Guarantee Reasoning Environment (AGREE), representing a significant advance in explainable AI-augmented MBSE. By leveraging a neurosymbolic cycle under human oversight, AGREE-Dog facilitates requirements and architecture reviews, automates model generation and repair, and guides formal verification and validation tasks. Its integration with the Open Source AADL Tool Environment (OSATE) enables interactive exploration of complex formal verification artifacts such as counterexamples, accelerates feedback loops, and fosters collaboration among diverse teams of traditional system engineers and formal verification experts.

We also highlight existing limitations and open research problems regarding abstraction fidelity, risk management, and long-term maintainability. We underline the need for further research to refine AI-human collaboration, develop robust evaluation metrics, and assess para-functional properties in AI-augmented processes supported by automated reasoning tools. Our primary goal is to share our view of both the progress made and the challenges that remain, illuminating a path toward reliable, accessible, and effective integration of trustworthy AI copilot technology in mission-critical MBSE and ultimately bridging the gap between formal verification and traditional system engineers at the industrial scale.

Engineering the Future of Autonomy

THE INSTITUTE

Pursuing fundamental as well as applied research in systems, information science, and engineering.

isis.vanderbilt.edu

Combining AI and Models to Identify Faults in Business Logic

Daniel Balasubramanian
Vanderbilt University

Dr. Daniel Balasubramanian is a Senior Research Scientist at the Institute for Software Integrated Systems and an Adjunct Associate Professor in the School of Engineering at Vanderbilt University. He is currently a PI on the DARPA Business Process Logic (BPL) program, the DARPA Cyber Agents for Security Testing and Learning Environments (CASTLE) program, and the DARPA Verified Security and Performance Enhancement of Large Legacy Software (V-SPELLS) program. He has served as a co-PI on the DARPA Space-Time Analysis for Cybersecurity (STAC) program, was the PI on an NSF Smart and Connected Communities project, and has research experience on a multitude of projects including the Model-Based Integration of Embedded Systems (MoBIES) project, the DARPA Producible and Adaptable Model-based Software (PAMS) project, the NASA Model-Transformation and Verification project, the DARPA Instant Foundry Adaptive through Bits project, the AFRL Resilient Software Systems (ReSoS) project, the DARPA META project, and model-based development tools from Microsoft Research.

Abstract

Business logic systems underpin practically every large-scale company and control defense-critical workflows in many sectors, including manufacturing, infrastructure, and logistics. The business processes implemented in these systems rely on a mixed collection of software systems and natural language artifacts for planning, modeling, execution, and auditing. Vulnerabilities and faults in these processes and the systems that implement them have wide-ranging consequences, from minor annoyances to significant loss of income and product delays. Thus, identifying vulnerabilities and faults in these systems is essential to the resiliency of defense-critical workflows. However, human factors, unformalized domain knowledge, and implicit requirements complicate the typical challenges involved in analyzing and securing these heterogenous systems. For example, the processes describing a company's quality management (QM) system might be spread across multiple documents describing how, and how often, QM activities are conducted. These processes might be modeled in a manufacturing execution system (MES), such as Solumina; material usage and availability might be logged in enterprise resource planning (ERP) software, such as Costpoint; QM activities might be logged on paper, or digitally; and the QM system as a whole might be subject to additional internal and external standards, such as ISO 9000. Actively identifying relevant artifacts and reasoning over them involves more than connecting and exploring well-defined semantics: tools must be able to ingest and formalize natural language, navigate complex rule hierarchies, and incorporate aspects of domain knowledge and human intuition. Further complicating the problem is the fact that the underlying processes are also subject to human failure.

To address the challenges above, we have developed an approach for analyzing business processes that combines recent advances in artificial intelligence and natural language processing with established model-based engineering and verification technologies. Our approach is underpinned by a formal modeling and integration language, defined in our web-based modeling tool, WebGME, into which we lift standard languages for describing business processes, such as Business Process Model and Notation (BPMN) and Solumina. These models serve as a semantic bridge between industry-standard description languages and a collection of formal methods tools, including the P model checker and the FORMULA symbolic analysis framework. We use these tools to validate processes against a set of universal faults, such as infinite loops and unreachable subprocesses, and we also use them to evaluate processes against rules and records extracted from artifacts written in natural language. Our pipeline for analyzing natural language artifacts uses large language models (LLMs) with enhancements such as self-reflection, process-aware chunking, and an annotation language that exposes essential aspects of business processes.

In the proposed presentation, we discuss the unique challenges that emerge in the analysis of business processes and present our progress towards creating an integrated solution. We frame these insights in the context of our team's work on Model-Enhanced Target-Adaptive Business Process Logic (META-BPL) as part of DARPA's Business Process Logic (BPL) program and the program's ontology of business process faults.

POSTER

Building an Integrated Assurance Ecosystem for DAHCS

Kesha Hietala
Sandia National Laboratories

Dr. Kesha Hietala is a Senior Member of Technical Staff at Sandia National Laboratories. Prior to joining Sandia, she spent two years in the automated reasoning group at Amazon Web Services, where she contributed to the development of the Cedar authorization language. She presented on the high-assurance development process for Cedar at HCSS 2023. Kesha earned her PhD in 2022 from the University of Maryland where she focused on formally verifying the quantum software stack.

Abstract

Digital technologies offer significant benefits in development speed, cost, and flexibility; however, their integration into historically analog systems introduces potential new failure modes. Currently, we lack efficient methods to rigorously evaluate these digital technologies with the confidence required for high-stakes applications.

To address this challenge, Sandia National Laboratories launched the Digital Assurance for High Consequence Systems (DAHCS, pronounced “Dax”) Mission Campaign (MC) in 2024. This ambitious 7-year, $45 million research initiative, part of Sandia’s Laboratory Directed Research and Development (LDRD) program, aims to develop innovative tools and methodologies that provide empirical, measurable digital assurance of high consequence systems across critical domains, including nuclear weapons, hypersonics, space, and critical infrastructure. The ultimate goal is to empower decision-makers to make confident, evidence-based trade-offs that account for mission risks posed by current and future digital threats.

Over the next seven years, the DAHCS MC will support a variety of projects across the digital assurance landscape. While project proposals must be submitted by Sandia staff, we actively encourage external collaborations. To measure progress, the Test & Evaluation (T&E) team has selected an open-source satellite project, Satellite Identification and Location (SIDLOC), as a realistic test case for the digital assurance ecosystem. Project participants will apply the tools, techniques, and methods developed under the DAHCS MC to SIDLOC, while the T&E team focuses on developing a comprehensive end-to-end assurance case for the application. This approach will help guide the DAHCS MC and identify gaps within the digital assurance ecosystem.

POSTER

Modeling Control Loops for Principled Microservices Software Development

Sandip Roy
Texas A&M University

Sandip Roy has served as Professor of Electrical & Computer Engineering at Texas A&M University, and Director of the Global Cyber Research Institute, since August 2023. Before that, he was on the faculty of Washington State University from 2003-2023. He has also held a joint appointment as Chief Scientist with the Pacific Northwest National Laboratory since 2020, and served as a Program Director (rotator) at the U.S. National Science from 2019-2022. His current research interests are focused on the assessing and managing the resilience of cyber- and cyber- physical systems, using systems&control theory methods. Throughout his career, he has also extensively worked on the estimation and control of dynamical network processes, for a range of applications.

Abstract

Testbench and Models for Principled Microservices Software Development in High-Consequence Applications

Thomas Gumienny, Albert Ma, Patrick Churchill, Grace Harris, and Sandip Roy, Texas A&M University
Benjamin Drozdenko, Naval Undersea Warfare Center

Microservices architectures for software can provide tremendous benefits in terms of scalability, portability, and efficiency. Therefore, it is not surprising that microservices solutions are being envisioned for not only enterprise applications but high-consequence defense and infrastructure systems. However, development and instantiation of microservices-based software is challenging, because of: 1) the intrinsic scale and complexity of the application software given its decomposition into microservices, 2) the heterogeneity of the platforms and operating systems on which the applications are run, and 3) the reliance on multiple sophisticated support tools and constructs (e.g., container orchestrators like Kubernetes) during software development and operation, among other factors. These challenges are often accentuated for high-consequence defense and infrastructure applications, where software development involves multiple public and private stakeholders, and software failures can have profound impacts on engineered system performance.

Given these challenges, microservices software development for high-performance applications would benefit from both: 1) test benches for pre-assessing microservices software architectures and constructs prior to full-scale software deployment; and 2) mathematical models for prediction of software performance. To our knowledge, few experimental or formal modeling capabilities of these types have been developed, that support microservices software development for high-consequence applications.

The purpose of this poster is to report on initial work at the Global Cyber Research Institute at Texas A&M University, to develop experimental and mathematical models for microservices software in high-consequence applications. The experimental model or test bench is a small-scale Kubernetes cluster that replicates a Command-and-Control application, wherein data from high-bandwidth sensors are ingested and communicated, for the purpose of surveillance and decision-making. The test bench is primarily envisaged as being used to assess cybersecurity and resilience for such Kubernetes clusters, through attack/failure emulation, telemetry to assess threat impacts, and modular testing of cyber defenses. Meanwhile, the mathematical modeling effort is focused on capturing adaptive or responsive functions that are intrinsic to large-scale microservices deployment, which can naturally be modeled as feedback control loops. As one example, the horizontal pod autoscaler in container orchestrators like Kubernetes undertake spin up or down of pods in response to workload measurements, so as to meet target per-pod workloads. We will discuss the development of feedback-control models for such responsive microservices orchestration functions, and describe how control-theory concepts can be brought to bear for principled analysis and design of the functions. Finally, we will discuss how both the experimental and mathematical models can be used in the software design pipeline for high-consequence systems.

Welcome to

2025