Meriel von Stein

https://missmeriel.github.io/ | meriel@virginia.edu | linkedIn/meriel-von-stein-b7318060/ | github/MissMeriel

EDUCATION

PhD Candidate of Computer Science (Software Engineering & Robotics) Charlottesville, VA | Ongoing
University of Virginia

Master's of Computer Science (Software Engineering & Robotics) Charlottesville, VA | Aug. 2022
University of Virginia

BA Honours of Art History (Islamic Art & Architecture) Oberlin, OH | May 2014
Oberlin College

PUBLICATIONS

PHYSCOV: PHYSICAL TEST COVERAGE FOR AUTONOMOUS VEHICLES

Carl Hildebrandt, Meriel von Stein, Sebastian Elbaum
Under Submission.
Python, C++, ROS, Gazebo, automation, stochastic learning, test coverage
Adequately covering autonomous vehicles' (AV) behavior is fundamental to their validation. Quantifying coverage is challenging as AV behavior is influenced by large, complex physical environment(s) that produce unique spatial signatures when sensed. We introduce a new abstraction, *RSR*, and coverage metric, *PhysCov*. RSR integrates sensor readings and physical reachability analysis to estimate the input region affecting the AV, and characterizes that region through a parameterizable geometric approximation that can trade quality for cost. We demonstrate *PhysCov*'s ability to quantify AV test suite coverage, and highlights its value through high-positive correlation with vehicle crashes.

DEEPMANEUVER: ADVERSARIAL TEST GENERATION FOR TRAJECTORY MANIPULATION OF AUTONOMOUS VEHICLES

Meriel von Stein, David Shriver, Sebastian Elbaum
Under Submission.
Python, Lua, PyTorch, BeamNG, automation, adversarial testing
Adversarial test generation aims to produce input perturbations that cause a DNN to compute incorrect outputs. For DNN-reliant autonomous vehicles, the effects of such perturbations are attenuated by other parts of the system as the vehicle state changes. We argue that for adversarial perturbations to be effective on autonomous vehicles they must account for the subtle interplay between the DNN and the vehicle states. Building on that insight, we develop DeepManeuver, an automated framework that interleaves adversarial test generation with the vehicle trajectory simulation. DeepManeuver generates perturbations that force maneuvers more effectively than state-of-the-art techniques by 419% on average and achieves multi-objective maneuvers with a minimum 52% rate of success.

FINDING PROPERTY VIOLATIONS THROUGH NETWORK FALSIFICATION: CHALLENGES, ADAPTATIONS AND LESSONS LEARNED FROM OPENPILOT

Meriel von Stein, Sebastian Elbaum
ASE 2022. Ann Arbor, MI, U.S.A.
Python, C++, PyTorch, ONNX, automation, ML, property reduction, falsification
Like most semi-autonomous driving systems, Openpilot relies on a sophisticated DNN susceptible to safety property violations that can lead to crashes. To uncover such potential violations before deployment, we investigate the use of falsification, a form of directed testing that analyzes a DNN to generate an input that will cause such a violation. Our investigation reveals challenges in applying falsifiers to real-world DNNs, conveys engineering efforts to overcome such challenges, and showcases falsifiers' potential to detect property violations and provide meaningful counterexamples.

PREPARING SOFTWARE ENGINEERS TO DEVELOP ROBOT SYSTEMS

Carl Hildebrandt, Meriel von Stein, Trey Woodlief, Sebastian Elbaum
ICSE 2022. Pittsburgh, PA, U.S.A.
Python, C++, ROS, STEM education theory, virtualized environments, simulation
We have designed and delivered a course to better prepare students to develop software for robot systems and support ongoing rapid expansion of the field. Our course emphasizes the distinctive challenges of software development for robots paired with related software engineering techniques, it provides many opportunities for experiential learning across both disciplines, and it lowers the barriers for learning how to build robotic systems.

AUTOMATED ENVIRONMENT REDUCTION FOR DEBUGGING ROBOTIC SYSTEMS

Meriel von Stein, Sebastian Elbaum
ICRA 2021. Xi'an, China.
Python, C++, ROS, Gazebo, automation, stochastic learning
In this work we present the first automated approach for reducing the environment in which a robot failed. Similar to software debugging techniques, our approach systematically performs a partition of the environment space causing a failure, executes the robot in each partition containing a reduced environment, and further partitions reduced environments that still lead to a failure.

PROBABILISTIC CONDITIONAL SYSTEM INVARIANT GENERATION WITH BAYESIAN INFERENCE

Meriel von Stein, Sebastian Elbaum, Lu Feng, Shili Sheng
Available via Arxiv as of December 2020.
Python, Java, ROS, motion capture, automation, probabilistic analysis
Probabilistic invariants can encode a family of conditional patterns, are generated using Bayesian inference to leverage observed trace data against priors gleaned from previous experience and expert knowledge, and are ranked based on their surprise value and information content. Our studies on two semi-autonomous mobile robotic systems show how the proposed approach is able to generate valuable and previously hidden stateful invariants.

WORK EXPERIENCE

UNIVERSITY OF VIRGINIA | PhD Candidate — Charlottesville, VA | ongoing

- Research assistant
  - Joined program August 2018, qualified in March 2020.
  - **Project leader; State-aware adversarial testing of system-embedded DNNs.** Adversarial DNN testing does not take into account the spatiotemporal relationships between the system containing the DNN and the environment containing the input space. State can be incorporated into the perturbation process and constrain the direction and magnitude of perturbation according to system state.
  - **Project leader; Extending falsification for real-world end-to-end neural nets.** Falsification tool development usually focuses on falsifying properties for small, "toy" neural network benchmarks. This leaves a dearth of functionality when applying to real-world models, which involve higher complexity. Extending these tools allows for falsification of networks in the critical path of real systems using deployment-relevant properties.
  - **Project leader; spatial partitioning and prioritization for debugging robotic systems.** Environment configurations can expose faults in robotic systems, but can also complicate the output complexity. Test environments can be programmatically reduced to produce the same failure and cut down on efforts to debug.
- Teaching assistant:
  - Introduced project concepts for robotics-oriented software analysis.
  - Developed course materials: designed, coded, and taught 2 labs and final project, and supported co-TAs' work in **Robotics for Software Engineers**; education paper accepted to ICSE 2022.
  - Led office hours, graded, discussed syllabus adjustments, and came up with project suggestions for **Software Analysis** and **Introduction to Embedded Computer Systems**.

NASA GODDARD SPACE FLIGHT CENTER | Pathways Program — Greenbelt, MD | Aug 2017 – Aug 2018

- Develop, update & maintain GMSEC satellite ground system API and component code.
- Reconcile federal infosec requirements with implementation from a top-down/bottom-up approach.
- Support code reviews & evaluate software systems from a security assurance perspective.
- Interview stakeholders on current & projected implementation of NIST and internal security standards.

NASA KENNEDY SPACE CENTER | Software Engineer Intern — Cape Canaveral, FL | Dec 2016 – Aug 2017

- Build & test proof-of-concept Beowulf cluster for granular mechanics and robotics simulations.
- Provide in-house software support for SwampWorks robotics & UAV projects.
- Design & implement in-house lab network and secure teleconferencing platform.
- Develop and debug testing suites for various prototypes.
- Design, develop and debug automated testing software for a future launch control system.
- Write, fix and execute unit tests; participating in and facilitating code inspections.
- Maintain technical points of contact and report project progress from a matrix management environment.

UNIV. OF NEBRASKA - LINCOLN | Cybersecurity Research Fellow Lincoln, NE | Jun 2016 – Aug 2016

- Ran static analysis on malicious Android apps using bash scripting, isolating data flows.
- Wrote C++ module for static analysis tool to handle Java 8 Reflection calls and represent them in Graphviz.
- Wrote colluding Android apps in Eclipse and Android Studio for sample runs of static analysis tool.
- Wrote scripts to analyze tool output via MySQL database & text parsing of application dexcode.
- Presented progress reports to supervising faculty & suggested follow-up courses of action.

PROJECTS

DEEPMANEUVER Python, Lua
Reproduce the technique outlined in **DeepBillboard** and improve upon it to leverage the kinematics of the vehicle and state of the test environment.

DDENV Python, ROS, Gazebo Simulation
End-to-end tool for delta-debugging robotic environments with a semi-known failure distribution.

ROBOTICS FOR SOFTWARE ENGINEERS Jekyll, Python, C++, Object-Oriented Design, ROS, Robotics, Simulation
End-to-end tool for delta-debugging robotic environments with a semi-known failure distribution.

BAYESIAN INFERENCE ENGINE FOR ROBOTICS HTML, Python, Bash, Java, CSS, Lex
Probabilistic inference engine for time series data traces.

GITHUB REPO SCRAPER FOR ROBOTIC SWARM PROJECTS Python
Scrape github for swarm projects that fit criteria for project maturity and centering around swarm control.

ROSBAG DATA CLEANING AND SPLINE INTERPOLATION Python, C++, ROS
Find largest common subinterval among publish rates and interpolate values to populate subintervals by data type.

ACTIVITIES AND HONORS

CSGSG 2022 Mentoring Chair Help new students transition to graduate life through enriching mentoring program, organizing events, and assisting with orientation and prospective visits.
ICSE 2021 Organizing Volunteer Main conference organizing volunteer supporting paper presentation sessions.
FIRST Robotics Software Engineering Mentor Software mentor and software-hardware working group liason.
Million Woman Mentor Project, Software and Electrical Engineering mentor, grades 1 through 4.
Association for Computing Machinery (ACM) member Rowan chapter, App Development working group.
Contributor to An Efficient, Robust, and Scalable Approach for Analyzing Interacting Android Apps at University of Nebraska-Lincoln Paper accepted by 2017 ICSE conference
Panelist, thesis presenter at James A. Rawley Graduate Conference in the Humanities, University of Nebraska-Lincoln placed second overall for best undergraduate paper.
Recipient of Oberlin College Grant, John F. Oberlin Scholarship (2010-2014).

SKILLS

Languages: Java, Python, Bash, C/C++, SQL
Robotics/ML Frameworks: ROS, Keras, TensorFlow, PyTorch, ONNX
Simulation Development: Gazebo, Unity, Blender, FlightGoggles, BeamNG, particle simulation
Spoken/Written Languages: English, Spanish, Turkish, French.
Other: Git/GitHub, MatLab, R, AutoCAD, VirtualBox, Docker, LaTeX, Gimp, JavaScript, TypeScript, HTML/CSS