Carnegie Mellon University
Software Engineering Institute

RESEARCH REVIEW 2021

the collaboration effect

Inspire. Integrate. Innovate.

The Collaboration Effect
Collaboration is an essential characteristic of the Carnegie Mellon University Software Engineering Institute (CMU SEI). It lies at the heart of our work to establish and advance software as a strategic advantage for national defense and security. As a federally funded research and development center (FFRDC) sponsored by the Under Secretary of Defense, Research and Engineering (USD(R&E)), collaboration ties together the research, development, piloting, transitioning, and policy input we conduct for the benefit of our sponsor and customers.

Collaborative interactions inspire new ideas that we integrate with our evolving research agenda to innovate new solutions. We call this process "the collaboration effect." This year's CMU SEI Research Review highlights the ways in which the collaboration effect advances the state of the art and successfully transitions these advances into practice.

The collaboration effect rests on three main touchpoints: the activities of study, make, and transition. These activities enable many connections that intertwine as we work with the Department of Defense (DoD), government agencies, academia, and industry to integrate our strategic areas of work in software engineering, cybersecurity, and artificial intelligence (AI). As we study and make, we iterate with collaborators on R&D and pilots. This work leads toward transition—often of robust, proven tools and techniques, sometimes in the form of expert advice that informs improved policy. Our work also produces artifacts to advance the state of the art and practice for software, such as technical articles, reports, and prototype software tools.

This book provides insights into research in our portfolio of public research projects for fiscal 2021 on behalf of our DoD sponsor and presented at the 2021 CMU SEI Research Review. In distribution parlance, these projects are labeled "Distribution A," which indicates that they may be shared openly to anyone without restriction. The projects highlighted include recently concluded work and work that continues in our research pipeline to study, make, and transition results to the benefit of DoD, the USG, academia, and the private sector.

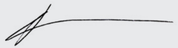
In the following pages, we take on the enduring challenges facing the DoD. Our decades-long engagement has informed our deep and nuanced understanding of the challenges faced across software, cyber, and artificial intelligence (AI). Our research springs from the DoD's need for software innovation and cybersecurity that continually evolves in support of its intensifying mission.

The DoD needs its software-enabled systems to

- bring capabilities that make new missions possible or improve the likelihood of success of existing ones
- be timely to enable the DoD to field new software-enabled systems and upgrades faster than our adversaries
- be trustworthy in construction and implementation and resilient in the face of operational uncertainties including known and yet-unseen adversary capabilities
- be affordable such that the cost of acquisition and operations, despite increased capability, is reduced and predictable and provides a cost advantage over our adversaries

Those requirements drive all CMU SEI work, whether for USD(R&E), DoD programs, federal civilian agencies, or industry.

I hope you enjoy reading about CMU SEI's fiscal 2021 research efforts, and that the following pages demonstrate the pride we take in this work. We stand by to work with you to help you make a difference, and we encourage you to contact us at info@sei.cmu.edu.



— **TOM LONGSTAFF**
Chief Technology Officer,
Carnegie Mellon University Software Engineering Institute

Contents

Automated Design Conformance During Continuous Integration

Principal Investigator
DR. ROBERT NORD
Principal Member of the Technical Staff

Software architecture enables our ability to innovate through extensible design, which provides for future growth in capability that is affordable and timely. To reduce the time needed to field capabilities and to lower lifecycle costs, the DoD has instructed program managers to consider a modular open systems approach (MOSA). MOSA promotes extensibility through technical standards such as the Future Airborne Capability Environment (FACE). However, a gap exists in verifying whether implemented capabilities satisfy the design constraints of a reference architecture such as FACE.

This project is creating an automated conformance checker that can be integrated into the continuous integration workflow to detect and report nonconformances in hours instead of the months or years it takes to discover these problems today. This technology will correctly identify design nonconformances with precision greater than 90%.

The central research of this project is recognizing abstractions commonly used in software architecture in C++ source code. Extracting design from code is hard because there are few indications of intent in the code and because implementations show significant variations. We see potential in applying code analysis, software architecture knowledge, and machine learning to extract design as implemented in the code and check conformance with the intended design. We are focusing on detecting nonconformance with architecture communication styles that are essential to achieving the goals of MOSA.

Developers can detect problems continuously and near the time they are introduced, allowing ***faster and more economical*** *realignment of implementation and design.*

The conformance checker will benefit developers and program managers. Developers can detect problems continuously and near the time they are introduced, allowing faster and more economical realignment of implementation and design. Program managers can hold developers (contractor or organic) accountable for delivering sustainable systems.

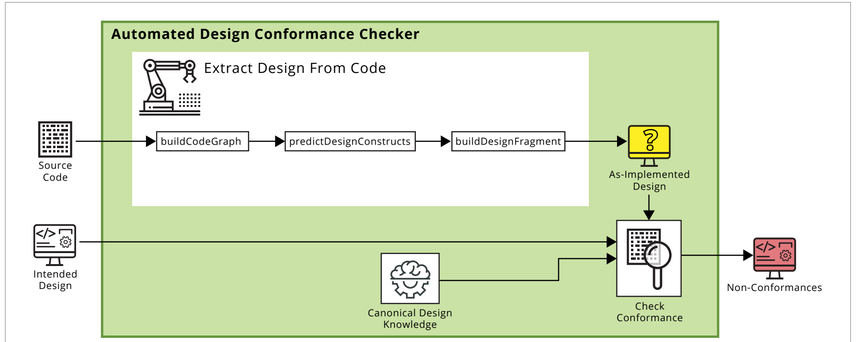


Fig 1. Prototype Design Conformance Checker: The approach builds on code analysis, software architecture, and machine learning.

IN CONTEXT

This FY2020–22 project

- advances the state of the art in applying machine learning (ML) to software engineering tasks
- aligns with SEI strategic focus areas of timely and trustworthy software by introducing automation into the development and acquisition lifecycle

SEI COLLABORATORS

JOHN KLEIN
Principal Member of the Technical Staff
Carnegie Mellon University Software Engineering Institute

LENA PONS
Software Architecture and AI Researcher
Carnegie Mellon University Software Engineering Institute

CHRIS SEIFRIED
Associate Engineer
Carnegie Mellon University Software Engineering Institute

JAMES IVERS
Principal Engineer
Carnegie Mellon University Software Engineering Institute

Combined Analysis for Source Code and Binary Code for Software Assurance

Principal Investigator
DR. WILL KLIEBER
Researcher

Many DoD entities need software assurance for both source code and binary code, as well as mixed systems (e.g., source code plus binary libraries). While there are many existing highly capable tools for static analysis of source code, tools for software assurance of binaries are fewer and much more limited. The objective of this line of work is to evaluate the feasibility of decompiling binaries for the purpose of (1) static analysis and (2) localized repairs to functions of the binary. More specifically, we aim to (1) develop a tool for determining whether individual functions have been correctly decompiled, (2) measure what percentage of functions are decompiled correctly on typical real-world binary code, and (3) measure how close static analysis on decompiled code approximates static analysis on the original source code.

We adapt an existing open-source decompiler (in particular, Ghidra) to produce decompiled code suitable for static analysis and/or repair, and we evaluate it with real-world (optimized) binary files. This project lays the groundwork for further work (including a follow-on FY22 project) to (1) enable the DoD to more accurately perform software assurance for projects that include binary components and (2) develop a framework for making localized repairs (either manual or automated) to functions of a binary library or executable.

*This line of work, if successful, will enable the DoD to **find and fix potential vulnerabilities** in binary code **that might otherwise be cost prohibitive to investigate or repair**, thereby increasing the trustworthiness of fielded software*

This line of work, if successful, will enable the DoD to find and fix potential vulnerabilities in binary code that might otherwise be cost prohibitive to investigate or repair, thereby increasing the trustworthiness of fielded software.

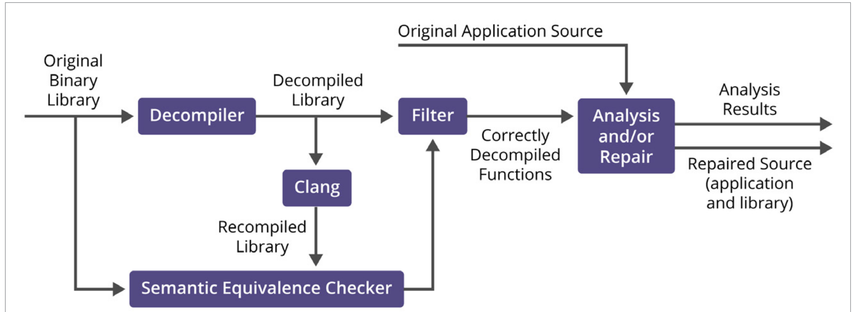


Fig 2. Envisioned Pipeline for Analysis and Repair of a Source-and-Binary Project

Our collaborators and interested transition partners at the DoD have binaries for which software assurance is desired; they will help us to evaluate and improve our tool, and they will be able to benefit from using the tool in practice when it is ready.

IN CONTEXT

This FY2021 project

- builds on DoD line-funded research on automated repair of code for integer overflow, inference of memory bounds, and automated code repair to ensure memory safety
- aligns with the CMU SEI technical objective to make software trustworthy in construction, correct in implementation, and resilient in the face of operational uncertainties, including known and yet-unseen adversary capabilities

SEI COLLABORATORS

DAVID SVOBODA
Software Security Engineer
Carnegie Mellon University Software Engineering Institute

EXTERNAL COLLABORATORS

DR. RUBEN MARTINS
Systems Scientist, Computer Science Department
Carnegie Mellon University

Knowing When You Don't Know: AI Engineering in an Uncertain World

Principal Investigator
DR. ERIC HEIM
Senior ML Researcher

The DoD is increasingly seeking to deploy AI systems for mission-critical tasks. Modern AI systems most commonly employ machine learning (ML) models to make important, domain-relevant inferences. However, due in part to uncertainty, state-of-the-art ML models can produce inaccurate inferences in scenarios where humans would reasonably expect high accuracy. Furthermore, many commonly used models do not provide accurate estimates about when they are uncertain about their predictions. Consequently, AI system components downstream from an ML model, or humans using the model's output to complete a task, must reason with incorrect inferences that they expect to be correct. Motivated by this gap, this project aims to accomplish the following objectives:

- Develop new techniques, and utilize existing ones, to give ML models the ability to express when they are likely to be wrong without drastically increasing the computational burden, requiring significantly more training data, or sacrificing accuracy.
- Develop techniques to detect the cause of uncertainty, learning algorithms that allow ML models to be improved after the cause of uncertainty is determined, and methods for reasoning in the presence of uncertainty without explicit retraining.
- Incorporate uncertainty modeling and methods to increase certainty into the ML models of government organizations.

Our work seeks to realize three overarching benefits. First, ML models in DoD AI systems will be made more transparent, resulting in safer, more reliable use of AI in mission-critical applications. Second, ML models will be more quickly and efficiently updated to adapt to dynamic changes in operational deployment environments. Third, we will make adoption of AI possible for missions where AI is currently deemed too unreliable or opaque to be used.

Our SEI team of Eric Heim, John Kirchenbauer, Jon Helland, and Jake Oaks brings expertise in the science and engineering of AI systems, human-computer interaction, enterprise-level infrastructures, and perspectives informed by a collective 50 years of experience leading

and conducting projects for both government and industry. Our CMU collaborators Dr. Zachary Lipton and Dr. Aarti Singh bring expertise in monitoring and improving ML models in the presence of uncertainty. They will provide important insight and graduate student support in producing high-quality research on topics related to detecting model uncertainty and mitigating its effects on the quality of model inference.

IN CONTEXT

This FY2021 project

- builds on DoD line-funded research, including graph algorithms and future architectures, big learning benchmarks; automated code generation for future-compatible high-performance graph libraries; data validation for large-scale analytics; and events, relationships, and script learning for situational awareness
- aligns with the CMU SEI technical objective to be timely so that the cadence of acquisition, delivery, and fielding is responsive to and anticipatory of the operational tempo of DoD warfighters and that the DoD is able to field these new software-enabled systems and their upgrades faster than our adversaries

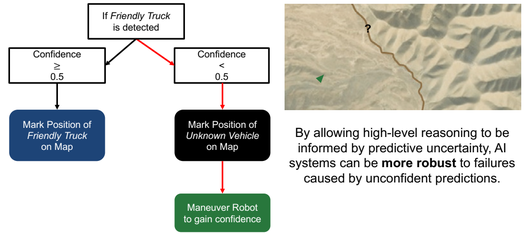


Fig 3. Quantifying Uncertainty: A Key Component for Informative and Robust AI Systems

SEI COLLABORATORS

JON HELLAND
Machine Learning Researcher
Carnegie Mellon University Software Engineering Institute

JOHN KIRCHENBAUER
Machine Learning Engineer
Carnegie Mellon University Software Engineering Institute

JACOB OAKS
Student Intern
Carnegie Mellon University Software Engineering Institute

EXTERNAL COLLABORATORS

AARTI SINGH
Associate Professor
Machine Learning Department
Carnegie Mellon University

ZACHARY LIPTON
Assistant Professor
Machine Learning Department
Carnegie Mellon University

Multicore Confidence

Principal Investigator
DR. BJÖRN ANDERSSON
Principal Researcher

Complex, cyber-physical DoD systems, such as aircraft, depend on correct timing to properly and reliably execute crucial sensing, computing, and actuation functions. Any timing failure can have disastrous consequences—an expected delay translating sensor data into actuation can cause system instability and loss of control. What's more, the complexity of today's DoD systems has increased the demand for use of multicore processors because unicore chips are either unavailable or not up to the task. However, concerns about timing have led to the practice of disabling all processor cores except one.

Any timing failure can have disastrous consequences—an expected delay translating sensor data into actuation can cause system instability and loss of control.

In this project, we aim to develop a solution to overcome this obstacle. This is a difficult challenge, because timing is determined by many shared resources in the memory system (including cache, memory banks, memory bus) with complex arbitration mechanisms, some of which are undocumented. The goal of our research is to demonstrate multicore timing confidence by achieving the following sub-objectives:

- **Verification.** Develop a method for timing verification that does not depend directly on undocumented design qualities and quantities.
- **Parameter extraction.** Develop a method for obtaining values for parameters in the model of a software system suited for the timing verification procedure mentioned above.
- **Configuration.** Develop a configuration procedure (such as assigning threads to processor cores or assigning priorities to threads) that takes a model as input and produces a configuration for which the verification will succeed (if such a configuration exists).

Photo U.S. Army

IN CONTEXT

This FY2019 project

- builds on prior DoD line-funded research and sponsored work on timing verification of undocumented multicore, verifying distributed adaptive real-time systems, high-confidence cyber-physical systems, and real-time scheduling for multicore architectures
- aligns with the CMU SEI technical objective to bring capabilities through software that make new missions possible or improve the likelihood of success of existing ones

SEI COLLABORATORS

BILL ANDERSON
Sr. Member of Technical Staff
Carnegie Mellon University Software Engineering Institute

DR. DIONISIO DE NIZ
Technical Director, Assuring Cyber-Physical Systems
Carnegie Mellon University Software Engineering Institute

ANTON HRISTOZOV
Software Engineer
Carnegie Mellon University Software Engineering Institute

MARK KLEIN
Principal Technical Advisor
Carnegie Mellon University Software Engineering Institute

EXTERNAL COLLABORATORS

HYOSEUNG KIM
Associate Professor, Department of Electrical and Computer Engineering
University of California, Riverside

JOHN LEHOCZKY
Thomas Lord University Professor of Statistics & Data Science, Department of Statistics & Data Science
Carnegie Mellon University

Predicting Inference Degradation in Production ML Systems

Principal Investigator
DR. GRACE LEWIS
Principal Researcher / TAS Initiative Lead

After machine learning (ML) systems are deployed, their models need to be retrained to account for differences between characteristics of training and production data. These differences over time lead to inference degradation—negative changes in the quality of ML inferences—which eventually reduces the trustworthiness of systems [DSB 2016; Gil 2019]. In DoD systems, failure to recognize inference degradation can lead to costly reengineering, system decommissioning, and misinformed decisions.

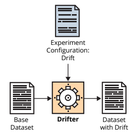
In DoD systems, failure to recognize inference degradation can lead to ***costly reengineering***, ***system decommissioning***, *and* ***misinformed decisions***.

Ideally, inference degradation would be quickly and reliably identified in production ML systems, allowing appropriate action to be taken (e.g., retraining, cautioning users, or taking a capability offline). The state of engineering practice in industry relies on periodic retraining and model redeployment strategies to evade data drift, without monitoring inference degradation. Without an analytic basis for the retraining interval, this frequent retraining strategy risks correcting for inference degradation too slowly (i.e., bad inference may be the basis for actions) or redeploying models too frequently (overconsuming potentially limited bandwidth if deployed in tactical scenarios and increasing the risk of taking a capability offline due to redeployment errors) [Diethe 2018; Manning 2018; and Tarraf 2019].

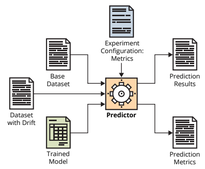
We propose to develop novel metrics that predict when a model's inference quality (e.g., positive predictive value [PPV], accuracy) will degrade below a threshold. The expected benefits of the metrics are that they will be able to determine (1) when a model really needs to be retrained so as to avoid spending resources on unnecessary retraining, and (2) when a model needs to be retrained before its scheduled retraining time so as to minimize the time that the model is producing sub-optimal results.

TEST HARNESS
Composed of Three Tools

Trainer trains a CNN-based model with a given set of parameters—based on TensorFlow.



Drifter generates a drifted data set from a base data set and a drift type (python).



Predictor runs a drift detection experiment and generates metrics for analysis (python).

We will focus on models based on convolutional neural networks (CNNs) for object detection and will use a publicly available satellite image data set as the source for test data. To further scope our study, we will focus on inference degradation stemming from the occurrence of data drift (frequency, recurrence, and abruptness drift).

Our vision for this work is that (1) our new metrics are incorporated into model development pipelines to provide better information on actions to take due to inference degradation, which includes starting the retraining process in a timely manner in order to provide continuous operations within accuracy thresholds, and (2) the community starts developing metrics and leveraging our test bed for models other than those based on CNNs and looks beyond drift metrics as the only predictor of inference degradation.

IN CONTEXT
This FY2021 project

- aligns with the CMU SEI technical objectives to bring capabilities that make new missions possible or improve the likelihood of success of existing ones and to be timely to enable the DoD to field new software-enabled systems and upgrades faster than our adversaries

The models we are developing to study inference quality are based on convolutional neural networks (CNNs) for object detection and will use a publicly available satellite image data set as the source for test data. To further scope our study, we will focus on inference degradation stemming from the occurrence of data drift (frequency, recurrence, and abruptness drift).

SEI COLLABORATORS

LENA PONS
Software Architecture and AI Researcher
Carnegie Mellon University Software Engineering Institute

SEBASTIÁN ECHEVERRÍA
Senior Engineer
Carnegie Mellon University Software Engineering Institute

JEFFREY CHRABASZCZ
Machine Learning Research Scientist
Carnegie Mellon University Software Engineering Institute

Projecting Quantum Computational Advantage versus Classical State of the Art

Principal Investigator
DR. JASON LARKIN
Senior Researcher

The potential of quantum computing, especially near term, is not going to be realized without close integration with state-of-the-art classical computing. Universal gate (UG) quantum computers share many foundational features with classical computers. Furthermore, UG quantum computers must show advantage against state-of-the-art classical software and/or hardware, and the two computing paradigms will be critically integrated as complementary technologies.

A major gap in achieving quantum advantage is the identification of applications in which quantum computing could provide computational advantage (in terms of time to solution, quality of solution, etc.). It is unclear which potential applications will realize quantum advantage among a variety of hardware, such as various UG technologies (e.g., superconducting qubit, trapped and neutral-atoms, photonics). Variation in hardware is typical in the near-term, noisy, intermediate-scale quantum (NISQ) computing era. This is a software–hardware co-synthesis challenge for quantum computing in the near-term.

UG quantum computing has emerged as the […] quantum computing technology that can demonstrate not just ***quantum supremacy*** *[…] but also* ***quantum advantage****.*

This project aims to produce a novel classical computing emulation and software–hardware co-synthesis framework for quantum computing technology aimed at applications driven by the portfolio of DoD research. UG quantum computing has emerged as the near-term (5- to 10-year) quantum computing technology that can demonstrate not just *quantum supremacy* (performing a computation not possible with a classical computer, regardless of usefulness), but also *quantum advantage* (performing a useful computation better and/or faster than a classical computer).

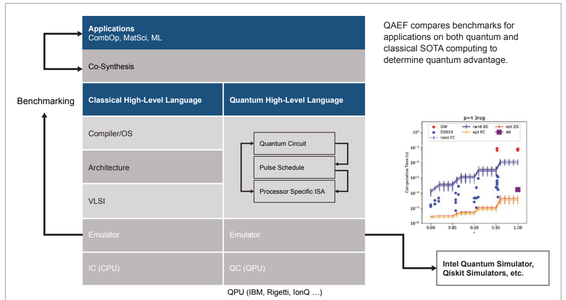


Fig 4. Quantum Advantage Evaluation Framework (QAEF)

IN CONTEXT

This FY2019–21 project

- relates to DoD interest in applying quantum computing to mission capability
- aligns with the CMU SEI technical objective to make software trustworthy in construction, correct in implementation, and resilient in the face of uncertainties, including known and yet-unseen adversary capabilities
- aligns with the CMU SEI technical objective to bring capabilities through software that make new missions possible or improve the likelihood of success for existing missions
- provides a gateway into futuristic computing architectures and increased computational power for artificial intelligence and machine learning

SEI COLLABORATORS

CATHERINE BERNACIAK
Machine Learning Research Scientist
Carnegie Mellon University Software Engineering Institute

BENJAMIN COMMEAU
Research Scientist, Quantum/ Advanced Computing
Carnegie Mellon University Software Engineering Institute

BRENT FRYE
Software Engineer
Carnegie Mellon University Software Engineering Institute

CHARLES HOLLAND
MTS, Principal Researcher
Carnegie Mellon University Software Engineering Institute

CHRIS INACIO
Chief Engineer
Carnegie Mellon University Software Engineering Institute

DANIEL JUSTICE
Software Developer
Carnegie Mellon University Software Engineering Institute

MARK SHERMAN
Technical Director, Cyber Security Foundations
Carnegie Mellon University Software Engineering Institute

EXTERNAL COLLABORATORS

FRANZ FRANCHETTI
Associate Dean for Research, College of Engineering
Carnegie Mellon University

MATÍAS JONSSON
CS/Physics Student
Carnegie Mellon University

SCOTT MIONIS
Electrical and Computer Engineering
Carnegie Mellon University

Rapid Adjudication of Static Analysis Alerts During Continuous Integration

Principal Investigator
DR. LORI FLYNN
Senior Software Security Engineer

The DoD has directed a shift toward continuous integration/continuous deployment (CI/CD) to maintain a competitive edge [McMurry 2018]. It is currently standard to run automated unit, integration, and stress tests during CI builds, but static analysis (SA) tools are not always part of builds because CI time frames are too short. However, SA tools could detect code flaws that are cheaper to fix earlier in the development process during CI builds.

It is increasingly common to use multiple SA tools and combine their alerts to maximize the identification of potential security flaws [Delaitre et al. 2018]. However, current SA tools produce some false positive (FP) alerts that require humans to inspect the code and manually adjudicate true vs. false alerts [Heckman 2011]. We use the term *alertCondition* to designate an alert from a tool mapped to a member of an external taxonomy of conditions (code flaws), for instance, CWE-190 from the CWE taxonomy. If SA is used within CI, alertConditions could stop a build and force human adjudication of true positive (TP) vs. FP, which slows development but might net an acceptable tradeoff if the slowdown is limited and/or occasional. Furthermore, many previously adjudicated FP alerts reappear each time an SA tool is run on a subsequent code version.

This research project will use machine learning and semantic analysis of data generated during CI/CD to ***reduce the number of alerts requiring human adjudication by 50%****.*

To maintain development velocity, DoD organizations with a continuous authority to operate (ATO) process have been forced to make tradeoffs in their security development testing and evaluation processes. For example, one organization removed SA tools from the CI/CD process, substituting a more expensive, less agile, and later manual review. Another kept SA tools, but reduced their sensitivity

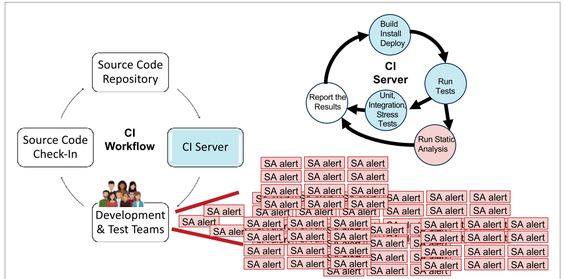


Fig 5. Rapid Handling of Static Analysis Meta-Alerts During CI

and analyzed only a small subset of the alerts, which introduced false negatives. We take the latter approach as a starting point, our goal being to increase efficiency by automating this process.

This research project will use machine learning and semantic analysis of data generated during CI/CD to reduce the number of alerts requiring human adjudication by 50% in multiple SA tool deployments without slowing the development process. More specifically, this project will

- improve the state of the art in reducing false positives and integrating SA tools into CI/CD processes
- improve the state of the practice by delivering and validating a prototype system that implements the new algorithms and measures the effectiveness of the techniques

IN CONTEXT

This FY2020–21 project

- builds on a number of previous projects, including "Rapid Construction of Accurate Automatic Alert Handling System: Model & Prototype" and "Running in the Cloud Without Breaking the Bank"
- aligns with the CMU SEI technical objective to make software trustworthy in construction, correct in implementation, and resilient in the face of operational uncertainties, including known and yet-unseen adversary capabilities

SEI COLLABORATORS

TYLER BROOKS
MTS, Engineer
Carnegie Mellon University Software Engineering Institute

RHONDA BROWN
Member of the Technical Staff
Carnegie Mellon University Software Engineering Institute

LYNDSI HUGHES
Systems Engineer
Carnegie Mellon University Software Engineering Institute

EBONIE MCNEIL
DevOps Engineer
Carnegie Mellon University Software Engineering Institute

JEFFREY MELLON
Machine Learning Research Scientist
Carnegie Mellon University Software Engineering Institute

WEI-REN MURRAY
Software Engineer
Carnegie Mellon University Software Engineering Institute

JOSEPH SIBLE
Associate Software Engineer
Carnegie Mellon University Software Engineering Institute

MATT SISK
Member of the Technical Staff
Carnegie Mellon University Software Engineering Institute

DAVID SVOBODA
Software Security Engineer
Carnegie Mellon University Software Engineering Institute

DUSTIN UPDYKE
MTS, Senior Engineer
Carnegie Mellon University Software Engineering Institute

JOSEPH YANKEL
MTS, Senior Engineer
Carnegie Mellon University Software Engineering Institute

Rapid Certifiable Trust

Principal Investigator
DR. DIONISIO DE NIZ
Technical Director, Assuring Cyber-Physical Systems

The DoD recognizes the need to field new cyber-physical systems (CPS) capabilities at an increasingly rapid pace, which is why it maintains a number of initiatives on rapid deployment. The demand for more rapid deployment, however, creates a need for verification techniques that can adapt to a faster deployment cadence, especially for CPS that are too big for traditional verification techniques and/or involve unpredictable aspects, such as machine learning.

The goal of Rapid Certifiable Trust is to reduce the deployment time of CPS by reducing the overall development and assurance times. We will do this by enabling the use of unverified commodity software components (e.g., open source drone piloting software) guarded by verified enforcers that guarantee the containment of unsafe component behavior. We are developing compositional verification techniques to allow us to use multiple enforced components minimizing and automatically removing conflicting enforcer assumptions (e.g., reducing a plane's airspeed to avoid a crash while increasing airspeed to prevent stalling). These techniques will allow us to assure full-scale systems, even if most of their functionality is implemented by unverified components.

The goal of Rapid Certifiable Trust is to reduce the deployment time of CPS by ***reducing the overall development and assurance times****.*

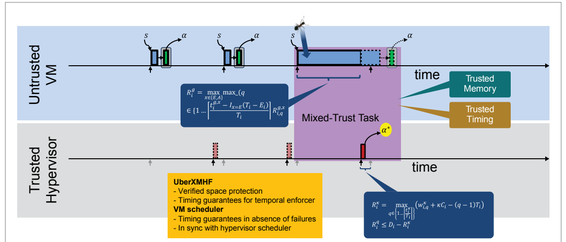


Fig 6. Real-Time Mixed-Trust Computation

IN CONTEXT

This FY2020–22 project

- seeks to verify software-reliant systems that interact with physical processes (e.g., aircrafts) to which existing verification technology does not scale
- will develop enforcing algorithms to identify unsafe control actions and replace them with safe actions
- uses drones to validate our approach in the SEI's drone lab
- aligns with the CMU SEI technical objective to make software trustworthy in construction, correct in implementation, and resilient in the face of operation uncertainties
- also aligns with the CMU SEI technical objective to make software delivery timely so that the cadence of acquisition delivery and fielding is responsive to and anticipatory of the operation tempo of DoD warfighters

SEI COLLABORATORS

DR. BJÖRN ANDERSSON
Principal Researcher
Carnegie Mellon University Software Engineering Institute

ANTON HRISTOZOV
Software Engineer
Carnegie Mellon University Software Engineering Institute

MARK KLEIN
Principal Technical Advisor
Carnegie Mellon University Software Engineering Institute

BRUCE KROGH
Faculty Emeritus
Carnegie Mellon University Software Engineering Institute

MICHAEL MCCALL
Associate Software Security Engineer
Carnegie Mellon University Software Engineering Institute

DR. GABRIEL MORENO
Senior Researcher
Carnegie Mellon University Software Engineering Institute

DR. AMIT VASUDEVAN
Senior Researcher
Carnegie Mellon University Software Engineering Institute

EXTERNAL COLLABORATORS

PAUL GRIFFIOEN
PhD Candidate, Department of Electrical and Computer Engineering
Carnegie Mellon University

HYOSEUNG KIM
Associate Professor, Department of Electrical and Computer Engineering
University of California, Riverside

JOHN LEHOCZKY
Professor of Statistics and Mathematics, Department of Statistics & Data Science
Carnegie Mellon University

DR. RUBEN MARTINS
Systems Scientist, Computer Science Department
Carnegie Mellon University

DR. RAFFAELE ROMAGNOLI
PostDoc Research Associate, Department of Electrical and Computer Engineering
Carnegie Mellon University

BRUNO SINOPOLI
Department Chair and Das Family Distinguished Professor
Washington University in St. Louis

README: A Learned Approach to Augmenting Software Documentation

Principal Investigator
DAN DECAPRIA
Senior Data Scientist

Modern software documentation processes in development, security, and operations (DevSecOps) software development lifecycles (SDLCs) are inadequate, time consuming, and difficult to quantify quantitatively. Anecdotally, any software documentation process can be painful [Shorter 2020]. For any given continuous integration/continuous deployment (CI/CD) SDLC methodology, crafting and maintaining high-quality software documentation content can be a subjective, tedious, meticulous process requiring significant understanding and domain knowledge. Additionally, in modern Agile CI/CD or DevSecOps sprinting paradigms, human-in-the-loop (HITL) software documentation blockers detract from development success-gauging metrics. This situation inspires negative perceptions of current documentation processes and efforts to mitigate the blocker through substandard (or even non-existent) iterative documentation efforts.

The README research initiative is a strategic step forward towards a descriptive content generative process in modern DoD DevSecOps SDLCs. The README proof of concept (POC) is not a templating engine. Rather, the primary differentiator between the README POC approach and emerging approaches in Development, Documentation, and Operations (DevDocOps) is that software documentation content is directly inferred from the underlying source code itself, backed by the SDLC cadence via DevSecOps policy.

The README approach relies on leveraging a machine learning (ML) modular architecture for learning the nuanced associations between Python3.8 source code and corresponding software engineering (SWE) descriptive lexicon language, in an unsupervised manner, from thousands of open source publicly available repositories' commit transaction histories. The README POC release establishes a viable cross-domain forward inference POC model learned from software repositories, and a minimum-viable-product (MVP) DevSecOps service prototype of the model as an exemplar.

The README ML cross-domain translation architecture is defined as a latent translation bridging model nested between two pretrained models over orthogonal data

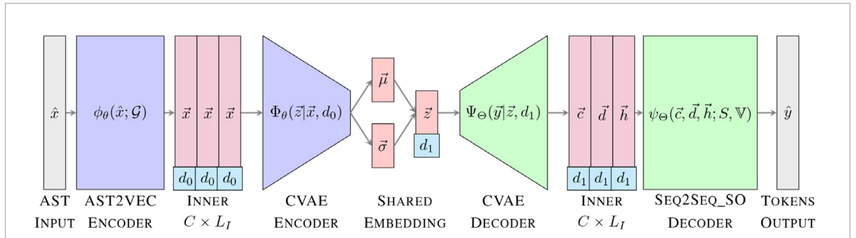


Fig 7. README POC Model

modalities [Tian 2019]. The README project refers to the nesting-based approach of pretrained models for cross-domain latent translation as the "Matryoshka Technique," facilitating domain modularity with a deeper network-forward through pretrained nested model reuse. The Matryoshka Technique provides a modular experimental harness for training and validation (T&V) of multiple pretrained models, under varying pretrained configuration hyper-parameterizations, for learning a nested, shared, latent space modeling structure between them.

For a software documentation content generative process, the cross-domain latent translation ML model, identified through this README research initiative, at reconstruction of each pretrained model's intermediate latent encodings, is a conditional variational auto-encoder (CVAE) nested between a pretrained encoder from AST2VEC and a decoder from Seq2Seq_SO with StackOverflow SWE vocabularies and word-similarity embeddings [Subramanian 2020; Paaß&n 2021; Cho 2014; Efstathiou 2018].

Results of the README research initiative are a successful answer to the research question; the Matryoshka Technique for nesting pretrained models for a learned cross-domain latent translation between source code snippets and SWE subjective language is viable. This approach establishes efficacy in a general approach facilitating domain modularity with a deeper network-forward through pretrained nested model reuse.

README will produce the following outcomes and deliverables:

- README DevSecOps SDLC MVP Prototype Service; Containerized Deployment Service Prototype
- README: A Learned Approach to Augmenting Software Documentation technical report

IN CONTEXT

This FY2020–21 project

- contributes to the SEI's strong portfolio of ongoing work in modernizing software development and acquisitions, AI, and autonomy
- ligns with the CMU SEI technical objective to bring capabilities that make new missions possible or improve the likelihood of success of existing ones
- aligns with the CMU SEI technical objective to be timely to enable the DoD to field new software-enabled systems and upgrades faster than our adversaries
- aligns with the CMU SEI technical objective to be affordable such that the cost of acquisition and operations, despite increased capability, is reduced and predictable and provides a cost advantage over our adversaries

SEI COLLABORATORS

VIOLET TURRI
Assistant Software Developer
Carnegie Mellon University
Software Engineering Institute

Safety Analysis and Fault Detection Isolation and Recovery Synthesis (SAFIR)

Principal Investigator
DR. JÉRÔME HUGUES
Senior Architecture Researcher

The operational complexity of cyber-physical systems (CPS) forces new autonomous features into day-to-day systems, such as vehicles and factories, a phenomenon termed Increasingly Autonomous CPS systems (IA-CPS) [Alves 2018]. IA-CPS have a complex architecture that weaves hardware, AI-enabled functions or decision-making processes, human operators, and software. They are time sensitive and substitute human actions with high-frequency real-time algorithms. In such systems, the conjunctions of faults and their timed propagation can cause fatal incidents, such as those involving autonomous cars. In these particular cases, the safety mechanisms were either too inefficient to prevent a fault or actually caused the incident.

This situation creates concerns for future DoD programs: These systems not only need to be able to detect failures and recover once, but they also need to be able to reconfigure multiple times—autonomously—as they adapt to different situations without human intervention.

The DoD's AI vision requires advances in safety analysis, and fault detection isolation and recovery synthesis (or SAFIR) to (1) model and analyze dynamic reconfiguration and fault propagation due to fault sequences, and (2) enforce safe reconfiguration. For these two concerns, SAFIR will improve architecture-led safety assessment processes by delivering new tool-supported analysis and code generation capabilities to designers.

These systems** not only **need to be able to detect failures and recover** once, but they also need to be able to **reconfigure multiple times—autonomously**—as they adapt to different situations **without human intervention.

More specifically, we will

- improve the state of practice in safety engineering in a model-based systems engineering (MBSE) context by considering timing propagation of failures in an Architecture Analysis & Design Language (AADL) based architectural description and improving AADL reconfiguration mechanisms to align with Dynamic Fault Tree (DFT) operators, and deliver an implementation of these operators
- apply DFT analysis to evaluate the effectiveness of existing Fault Detection Isolation and Recovery (FDIR) policies, synthesize FDIR policies by processing DFT simulation traces, and enrich architectural descriptions with specific fault detection and reconfiguration mechanisms

SAFIR addresses safety analysis of time-sensitive CPS in both its theoretical and practical dimensions, and contributes to the SEI's line of research on artificial intelligence and autonomy. At the end of the first year, SAFIR has established the theoretical foundation to perform safety evaluations in the context of time-dependent failure conditions.

IN CONTEXT

This FY2019–21 project

- builds on SEI expertise in Model-Based Systems Engineering, safety analysis and the AADL language, and extends past contributions from Integrated Safety and Security Engineering (ISSE) and TwinOps
- aligns with the CMU SEI technical objective to bring capabilities through software that make new missions possible or improve the likelihood of success of existing ones and to be trustworthy in construction and implementations
- also aligns with the CMU SEI technical objective to be resilient in the face of operational uncertainties, including known and yet-unseen adversary capabilities

Photo A drone produced by ANAFI USA, a manufacturer that has produced drones for the Blue Small Unmanned Aircraft Systems (sUAS) program sponsored by the U.S. Army and Defense Innovation Unit

SEI COLLABORATORS

DAVID GLUCH
Software Architecture Researcher
Carnegie Mellon University Software Engineering Institute

AARON GREENHOUSE
Senior Architecture Researcher
Carnegie Mellon University Software Engineering Institute

KEATON HANNA
Assistant Software Engineer
Carnegie Mellon University Software Engineering Institute

JOHN HUDAK
MTS, Principal Engineer
Carnegie Mellon University Software Engineering Institute

DR. SAM PROCTER
Senior Architecture Researcher
Carnegie Mellon University Software Engineering Institute

CHUCK WEINSTOCK
Principal Researcher
Carnegie Mellon University Software Engineering Institute

LUTZ WRAGE
Sr. Member of the Technical Staff
Carnegie Mellon University Software Engineering Institute

Safety Analysis and Towards Incremental and Compositionally Verifiable Security of CHIC-Centric Cyber Physical Systems

Principal Investigator
DR. AMIT VASUDEVAN
Senior Researcher

DoD cyber-physical systems (CPS) employ commodity heterogeneous interconnected computing (CHIC) platforms and associated software stacks (e.g., ARM/Linux) to deliver capabilities at the speed of relevance [Osborn 2020; Krazit 2019; Keller 2019; Villarreal 2019]. However, the DoD faces a challenge achieving assurance in CHIC-centric CPS implementation security, because such systems employ multiple hardware platforms and multiple, large, layered software. What's more, these systems are frequently produced by disparate developers. A recent U.S. Government Accountability Office (GAO) report highlights security issues in CHIC-centric CPS implementations [GAO 2018].

*Our solution focuses on **development-compatible, implementation-level, protected, and verifiable execution** building blocks that retrofit with existing code, incrementally, at a fine granularity, with composability across multiple CHIC stack implementation layers.*

In this project, we draw from our published broad vision and strategy [Vasudevan 2020]. We explore the viability of provable, cost-effective, and innocuous (applicable on existing software and preserve existing functionality, such as NASA innocuity) CHIC-centric CPS implementation security [Halloway 2019]. Our solution focuses on development-compatible, implementation-level, protected, and verifiable execution building blocks that retrofit with existing code, incrementally, at a fine granularity, with composability across multiple CHIC stack implementation layers. Our scope in this project is the design, implementation, and verification of a critical execution path for CPS: secure on-platform sensor access that protects the integrity of the existing CPS application and sensor hardware/driver with trusted control and a data path between them. There are three high-level pieces to our approach (see Figure 8):

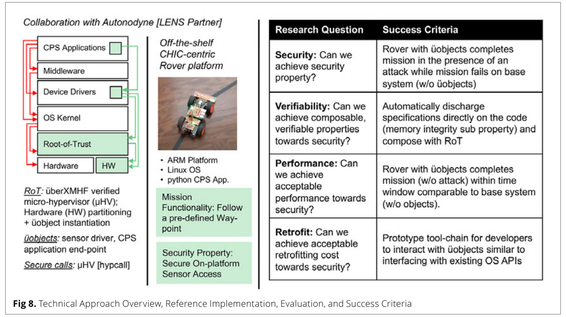
1. Interface confined implementation-level object abstractions (überobjects or üobjects): implementation-level building blocks that form fine-grained monitors around a system-level resource (e.g., data memory and I/O area) towards a security property
2. Runtime protected set of üobjects (üobject collections): a set of üobjects within a given address space at runtime, bootstrapped by a platform root-of-trust entity that endows memory protection and secure call routings
3. An implementation-level assume-guarantee reasoning framework that allow us to formally reason about interleaved executions of üobjects in the presence of unverified (and unavoidable) legacy components [Vasudevan 2016]

Among the planned outputs of this project is a demonstration of our approach on an off-the-shelf rover CPS platform with secure sensor access protection via üobjects that allows immunity against an entire class of memory integrity attacks. This will serve to showcase the viability of our approach to DoD and DoD industrial establishments. We will also make open source our associated prototype artifacts, code, and documentation (e.g., release via GitHub). This will enable DoD and DoD industrial establishments to start experimenting with üobjects within relevant application domains.

IN CONTEXT

This FY2021 project

- aligns with the CMU SEI technical objective to bring capabilities that make new missions possible or improve the likelihood of success of existing ones
- aligns with the CMU SEI technical objective to Be Trustworthy in construction and implementation, and resilient in the face of operational uncertainties including known and yet-unseen adversary capabilities.



Research Question	Success Criteria
Security: Can we achieve security property?	Rover with üobjects completes mission in the presence of an attack while mission fails on base system (w/o üobjects)
Verifiability: Can we achieve composable, verifiable properties towards security?	Automatically discharge specifications directly on the code (memory integrity sub property) and compose with RoT
Performance: Can we achieve acceptable performance towards security?	Rover with üobjects completes mission (w/o attack) within time window comparable to base system (w/o objects).
Retrofit: Can we achieve acceptable retrofitting cost towards security?	Prototype tool-chain for developers to interact with üobjects similar to interfacing with existing OS APIs

Fig 8. Technical Approach Overview, Reference Implementation, Evaluation, and Success Criteria

SEI COLLABORATORS

ANTON HRISTOZOV
Software Engineer
Carnegie Mellon University Software Engineering Institute

MICHAEL MCCALL
Associate Software Security Engineer
Carnegie Mellon University Software Engineering Institute

BRUCE KROGH
Faculty Emeritus
Carnegie Mellon University Software Engineering Institute

EXTERNAL COLLABORATORS

DR. RUBEN MARTINS
Systems Scientist, Computer Science Department
Carnegie Mellon University

DR. RAFFAELE ROMAGNOLI
PostDoc Research Associate, Department of Electrical and Computer Engineering
Carnegie Mellon University

Spiral/AIML: Co-optimization for High-Performance, Data-Intensive Computing in Resource-Constrained Environments

Principal Investigator
DR. SCOTT MCMILLAN
Principal Engineer

Commanders and warfighters in the field rely on data, and the Department of Defense and U.S. intelligence community have an overwhelming data collection capability. This capability far outpaces the ability of human teams to process, exploit, and disseminate information. Artificial intelligence (AI) and machine learning (ML) techniques show great promise for augmenting human intelligence analysis. However, most AI/ML algorithms are computationally expensive, data intensive, and difficult to implement efficiently in increasingly complex computer hardware and architectures. What's more, moving very large amounts of data through tactical and operational military networks requires forward deployment of advanced AI/ML techniques to support commanders and warfighters in theaters with equipment constrained by cost, size, weight, and power (CSWAP).

If successful, our solution will allow platform developers to realize high-performance AI/ML applications on leading-edge hardware architectures ***faster*** *and* ***cheaper****.*

As the military adopts AI/ML to augment human teams, the cost of implementing and re-implementing AI/ML software on new hardware platforms will be prohibitive. To address these challenges, we propose to build on CMU's Spiral technology, a hardware-software co-optimization system that will automatically

- search and select hardware configurations that meet CSWAP requirements
- generate optimized codes for the selected hardware configuration and the irregular, data-intensive computations required for AI/ML algorithms

If successful, our solution will allow platform developers to realize high-performance AI/ML applications on leading-edge hardware architectures faster and cheaper. These advances will allow for rapid development and deployment of capabilities across the spectrum of national and tactical needs.

Acknowledgments
Elliott Binder, Mark Blanco, Paul Brouwer, Mark Cheung, Anuva Kulkarni, Dr. Het Mankad, Peter Oostema, George Ralph, Courtney Rankin, Sanil Rao, Dr. Fazle Sadi, Sandra Sanjeev, John Shi, Dr. Daniele Spampinato, Upasana Sridhar, Arvind Srinivasan, Guanglin Xu, Vadim Zaliva, Jiyuan Zhang

IN CONTEXT

This FY2019–21 project

- builds on DoD line-funded research and sponsored work on automated code generation for future-compatible high-performance graph libraries, big learning benchmarks, GraphBLAS API specification, and graph algorithms on future architectures
- is related to a set of programs at the Defense Advanced Research Projects Administration (DARPA) under the Electronics Resurgence Initiative (ERI) umbrella (Hierarchical Identify Verify Exploit [HIVE], Software Defined Hardware [SDH], Domain Specific System on Chip [DSSoC], etc.) for which the CMU SEI has PWP work
- aligns with the CMU SEI technical objective to be affordable such that the cost of acquisition and operations, despite increased capability, is reduced and predictable and provides a cost advantage over our adversaries

Col. Drew Cukor, USMC, observed, "Rapidly delivering artificial intelligence to a combat zone won't be easy." To address this challenge, the SEI is developing Spiral/AIML: Co-optimization for High-Performance, Data-Intensive Computing in Resource-Constrained Environments.

Photo U.S. Army

EXTERNAL COLLABORATORS

FRANZ FRANCHETTI
Associate Dean for Research, College of Engineering
Carnegie Mellon University

JAMES C. HOE
Professor, Electrical and Computer Engineering
Carnegie Mellon University

TZE MENG LOW
Assistant Research Professor, Electrical and Computer Engineering
Carnegie Mellon University

JOSÉ MOURA
Professor, Electrical and Computer Engineering
Carnegie Mellon University

SEI COLLABORATORS

JOHN WOHLBIER
Senior Research Scientist
Carnegie Mellon University Software Engineering Institute

OREN WRIGHT
Senior Researcher
Carnegie Mellon University Software Engineering Institute

DR. JASON LARKIN
Senior Researcher
Carnegie Mellon University Software Engineering Institute

Train, but Verify: Towards Practical AI Robustness

Principal Investigator
DR. NATHAN VANHOUDNOS
Senior Machine Learning Research Scientist

The current challenges to the training and verification of secure machine learning (ML) stem from

1. the difficulty of enforcing quality attributes in a system that is trained on data instead of directly constructed from requirements
2. the fundamental advantage that an attacker has, namely that the attacker needs to only violate a single security policy, while the defender needs to enforce all of the security policies

The DoD has not been exempt from these challenges. The current state of the art in secure ML is to train systems to either enforce a single security policy or train auxiliary systems to detect violations of a single security policy. Very little extant work focuses on multiple security policies. For example, there exist systems in the DoD that make high-stakes decisions and yet were also trained on sensitive data. This implies that the system should enforce at least two security policies simultaneously (i.e., the ML system should neither do the wrong thing when presented with adversarial input nor reveal sensitive information about the training data during its operation).

*…the ML system should **neither** do the wrong thing when presented with adversarial input **nor reveal sensitive information** about the training data during its operation.*

In this "Train, but Verify" project, we will attempt to address the gap in the state of the art on secure training of ML systems with two objectives:

1. Train secure AI systems by training ML models to enforce at least two security policies.
2. Verify the security of AI systems by testing against declarative, realistic threat models.

We consider security policies from the Beieler taxonomy: Ensure that an ML system does not learn the wrong thing during training (e.g., data poisoning), do the wrong thing during operation (e.g., adversarial examples), or reveal the wrong thing during operation (e.g., model inversion or membership inference).

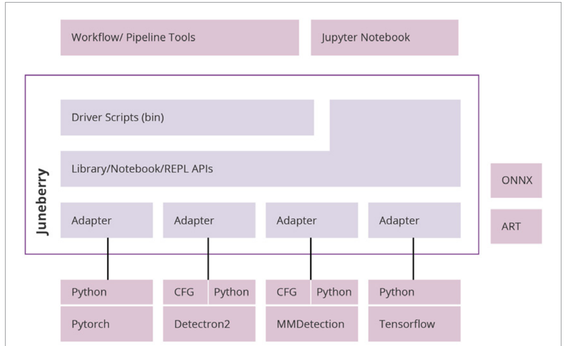


Fig 9. Juneberry, an architecture we created to support our research on Train, but Verify, improves the experience of machine learning experimentation by providing a framework for automating the training, evaluation, and comparison of multiple models against multiple datasets, reducing errors and improving reproducibility. We've made Juneberry available to other researchers on GitHub: https://github.com/cmu-sei/Juneberry

IN CONTEXT

This FY2020-22 project

- aligns with the CMU SEI technical objective to be trustworthy in construction and implementation and resilient in the face of operational uncertainties, including known and yet-unseen adversary capabilities

SEI COLLABORATORS

ANDREW MELLINGER
Senior Software Developer
Carnegie Mellon University
Software Engineering Institute

SEI Contributors

- Rob Beveridge
- Matthew Churilla
- Jon Helland
- Annika Horgan
- Bill Shaw
- Violet Turri
- Tina Sciullo-Schade
- Nick Winski

EXTERNAL COLLABORATORS

LUJO BAUER
Professor
Carnegie Mellon University

MATT FREDRIKSON
Associate Professor
Carnegie Mellon University

BRYAN PARNO
Associate Professor
Carnegie Mellon University

Students

- Aymeric Fromherz (Carnegie Mellon University)
- Clement Fung (Carnegie Mellon University)
- Klas Leino (Carnegie Mellon University)
- Kevin Li (Carnegie Mellon University)
- Weiran Lin (Carnegie Mellon University)
- Zifan Wang (Carnegie Mellon University)

Untangling the Knot: Automating Software Isolation

Principal Investigator
JAMES IVERS
Principal Engineer
Lead, Architecture Design, Analysis & Automation

Software-reliant systems need to evolve over time to meet new requirements and take advantage of new technology. However, all too often the structure of software becomes too complicated to allow rapid and cost-effective improvements. This challenge is common in long-lived DoD systems and not uncommon even in newer systems, and it makes isolating a collection of functionality for use in a new context, or clean replacement by an improved version, difficult. Software refactoring can facilitate such changes, but can require tens of thousands of staff hours.

This project has created a refactoring assistant that generates recommended refactorings that isolate functionality from its tangle of system dependencies. Our goal is to reduce the time required for this kind of software refactoring by two-thirds. In one DoD example, a contractor estimated 14 thousand hours of software development work alone (excluding integration and testing) to isolate a mission capability from the underlying hardware platform. If successful, our work would reduce the development time required to less than 5 thousand hours.

*Our goal is to **reduce the time required** for this kind of software refactoring **by two-thirds**.*

Our prototype combines advances in search-based software engineering with static code analysis and refactoring knowledge. It is unique in its focus on mission-relevant goals as opposed to improving general software metrics. This goal is incorporated in genetic algorithms through fitness functions that guide the search to solutions for the project-specific goal. In practice, our prototype recommends solutions that solve more than 85% of the problem on typical projects, suggesting that our effort reduction goal is obtainable. The search algorithm relies on a representation derived from static code analysis and uses formalizations of refactorings as operations to apply during search.

This work has broad implications for moving existing software to modern architectures and infrastructures

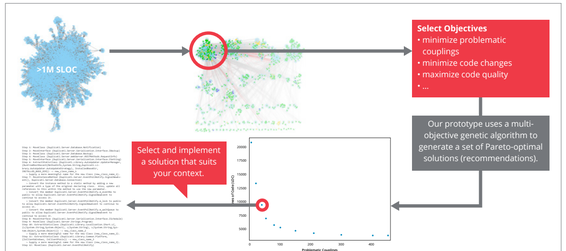


Fig 10. Generating Refactoring Recommendations

such as service-based, microservice, cloud environments, and containers. It also addresses a pervasive research challenge in improving automated support for software refactoring tasks.

IN CONTEXT

This FY2019–21 project

- builds on prior DoD line-funded research in software architecture analysis, static code analysis, and identifying technical debt
- aligns with the CMU SEI technical objective to make software delivery timely so that the cadence of acquisition, delivery, and fielding is responsive to and anticipatory of the operational tempo of DoD warfighters
- addresses a widespread, recurring need in software organizations (as requirements and technology are never frozen in time, the need to adapt working software to new contexts is likely to remain a common need across many software systems)

SEI COLLABORATORS

DR. ROBERT NORD
Principal Member of the Technical Staff
Carnegie Mellon University Software Engineering Institute

DR. IPEK OZKAYA
Tech. Director, Engineering Intelligent Software Systems
Carnegie Mellon University Software Engineering Institute

CHRIS SEIFRIED
Associate Engineer
Carnegie Mellon University Software Engineering Institute

SEI Contributors

- Mario Benítez
- Vaughn Coates
- Andrew Kotov
- Reed Little
- Craig Mazzotta
- Scott Sinclair
- Jake Tanenbaum

EXTERNAL COLLABORATORS

THIAGO FERREIRA
Assistant Professor
University of Michigan, Dearborn

CLEM IZURIETA
Associate Professor
Montana State University

MAROUANE KESSENTINI
Associate Professor
University of Michigan, Dearborn

SCOTT PAVETTI
Assistant Teaching Professor
Carnegie Mellon University

CHRIS TIMPERLEY
Systems Scientist
Carnegie Mellon University

Students

- Chaima Abid (University of Michigan)
- Gavin Austin (Montana State University)
- Jared Frank (University of Pittsburgh)
- Carly Jones (Carnegie Mellon University)
- Katie Li (Carnegie Mellon University)
- Red Rajput (Carnegie Mellon University)
- Amy Tang (Carnegie Mellon University)
- Jeff Yackley (University of Michigan)

References

[Alves 2018]
Alves, E. E.; Devesh, B.; Hall, B.; Driscoll, K.; Murugesan,A.; & Rushby, J. *Considerations in Assuring Safety of Increasingly Autonomous Systems*. Technical Report NASA/CR-2018-220080, NF1676L-30426, NASA Air TransportationAnd Safety. 2018.

[Cho 2014]
Cho, K.; Merrienboer, B.V.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; & Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Presented at the *Conference on Empirical Methods in Natural Language Processing*. October 2014.

[Delaitre et al. 2018]
Delaitre, Aurelien M.; Bertrand C. Stivalet; Paul E. Black; Vadim Okun; Terry S. Cohen; and Athos Ribeiro. *SATE V Report: Ten Years of Static Analysis Tool Expositions*. Special Publication (NIST SP)-500-326. National Institute of Standards and Technology. 2018.

[Diethe 2019]
Diethe, T.; Borchert, T.; Thereska, E.; Balle, B.; & Lawrence, N. Continual Learning in Practice. Presented at the *NeurIPS 2018 Workshop on Continual Learning*. December 2018. https://arxiv.org/pdf/1903.05202.pdf

[DSB 2016]
Defense Science Board. *Summer Study on Autonomy*. Office of the Under Secretary of Defense for Acquisition, Technology and Logistics. June 2016. https://www.hsdl.org/?view&did=794641

[Efstathiou 2018]
Efstathiou, Vasiliki; Chatzilenas, Christos; & Spinellis, Diomidis. Pages 38–41. Word embeddings for the software engineering domain. In Proceedings of the 15th International Conference on Mining Software Repositories (MSR '18). Association for Computing Machinery, New York, New York. 2018. DOI: https://doi.org/10.1145/3196398.3196448

[GAO 2018]
United States Government Accountability Office. *Weapon Systems Cybersecurity: DoD Just Beginning to Grapple with Scale of Vulnerabilities*. October 2018. https://www.gao.gov/assets/700/694913.pdf

[Gil 2019]
Gil, Y. & Selman, B. *A 20-Year Community Roadmap for Artificial Intelligence Research in the US. Computing Community Consortium (CCC) and the Association for the Advancement of Artificial Intelligence (AAAI)*. Computing Community Consortium. August 2019. https://arxiv.org/abs/1908.02624

[Halloway 2019]
Halloway, Michael C. *Understanding the Overarching properties*. NASA/TM–2019–220292. National Aeronautics and Space Administration. July 1, 2019. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20190029284.pdf

[Heckman 2011]
Heckman, Sarah & Laurie Williams. A systematic literature review of actionable alert identification techniques for automated static code analysis. *Information and Software Technology*. Number 53. Volume 4. April 2011. Pages 363-387.

[Keller 2019]
Keller, John. Navy chooses open-architecture water-cooled shipboard computers from GTS for SEWIP and self defense systems. *Military Aerospace*. January 16, 2019. https://www.militaryaerospace.com/computers/article/16722033/navy-chooses-openarchitecture-watercooled-shipboardcomputers-from-gts-for-sewip-and-self-defense-systems

[Krazit 2019]
Krazit, Tom. How the U.S. Air Force Deployed Kubernetes and Istio on an F-16 in 45 days. *The New Stack*. December 24, 2019. https://thenewstack.io/how-the-u-s-air-forcedeployed-kubernetes-and-istio-on-an-f-16-in-45-days/

[Manning 2018]
Manning, J.; Langerman, D.; Ramesh, B.; Gretok, E.; Wilson, C.; George, A.; & Crum, G. Machine-Learning Space Applications on SmallSat Platforms with TensorFlow. Presented at the *32nd AIAA/USU Conference on Small Satellites. 2018*. https://digitalcommons.usu.edu/smallsat/2018/all2018/458/

[McMurray 2018]
McMurry, Robert D. & Roper, William B. *Establishment of Air Force Program Executive Officer (PEO) Digital*. [Memorandum for all AFPEOs.] Department of the Air Force. August 29, 2018. https://www.hanscomreps.org/wp-content/uploads/2018/09/20180829-PEO-Digital-Establishment-Memo-Signed.pdf

[Osborn 2020]
Osborn, Kris. New Air Force B-21 stealth bomber takes key technology step toward war readiness. *Fox News*. June 2, 2020. https://www.foxnews.com/tech/new-air-force-b-21-stealthbomber-takes-key-technology-step-toward-war-readiness

[Paaßen 2021]
Paaßen, B.; McBroom, J.; Jeffries, B.; Koprinska, I.; & Yacef, K. Mapping Python Programs to Vectors using Recursive Neural Encodings. *Journal of Educational Datamining*. 2021. [In press.]

[Shorter 2020]
Shorter, Cameron. What is good documentation for software projects? *Opensource.com*. April 6, 2020. https://opensource.com/article/20/4/documentation

[Subramanian 2020]
Subramanian, A. *PyTorch-VAE. 2020*. https://github.com/AntixK/PyTorch-VAE

[Tarraf 2019]
Tarraf, Danielle C. et al. The Department of Defense Posture for Artificial Intelligence: Assessment and Recommendations. *RAND Corporation*. 2019. https://www.rand.org/pubs/research_reports/RR4229.html

[Tian 2019]
Tian, Y. & Engel, J. Latent Translation: Crossing Modalities by Bridging Generative Models. arXiv:1902.08261. *arXiv*. February 21, 2019. https://arxiv.org/abs/1902.08261

[Vasudevan 2016]
Vasudevan, Amit; Chaki, Sagar; Maniatis, Petros; Jia, Limin; & Datta, Anupam. überSpark: Enforcing Verifiable Object Abstractions for Automated Compositional Security Analysis of a Hypervisor. Pages 87-104. In *Proceedings of USENIX Security Symposium*. Austin, Texas. August 2016. https://www.usenix.org/sites/default/files/sec16_full_proceedings.pdf

[Vasudevan 2020]
Vasudevan, Amit; Maniatis, Petros; & Martins, Ruben. überSpark: Practical, Provable, End-to-End Guarantees on Commodity Heterogenous Interconnected Computing Platforms. *ACM SIGOPS Operating Systems Review Journal – Special Issue on Formal Methods & Verification*. Volume 54. Number 1. July 2020. Pages 8-22. https://doi.org/10.1145/3421473.3421476

[Villarreal 2019]
Villarreal, Jennifer. GE Aviation and Auterion provide All-in-one hardware and software platform for commercial drones. *GE Aviation*. November 5, 2021 [accessed]. https://www.geaviation.com/press-release/systems/ge-aviation-and-auterion-team-provide-all-one-hardware-and-software-platform

Copyright

About Us

The Software Engineering Institute (SEI) at Carnegie Mellon University is a Federally Funded Research and Development Center (FFRDC)—a nonprofit, public–private partnership that conducts research for the United States government. One of only 10 FFRDCs sponsored by the U.S. Department of Defense (DoD), the SEI conducts R&D in software engineering, systems engineering, cybersecurity, and many other
areas of computing, working to introduce private-sector innovations into government.

As the only FFRDC sponsored by the DoD that is also authorized to work with organizations outside of the DoD, the SEI is unique. We work with partners throughout the U.S. government, the private sector, and academia. These partnerships enable us to take innovations from concept to practice, closing the gap between research and use.

Contact Us

Carnegie Mellon University
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

412.268.5800 | 888.201.4479
sei.cmu.edu | info@sei.cmu.edu