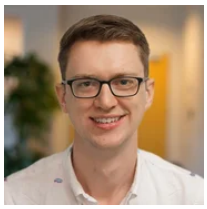
Search the blog

Software Engineering Institute

SEI Blog

Integrating Safety and Security Engineering for Mission-Critical Systems



SAM PROCTER AND **SHOLOM G. COHEN**

MAY 10, 2021

PUBLISHED IN

Software Architecture

CITE

Get Citation

TAGS

Safety-Related Requirements | Security-Related Requirements

Critical systems must be both safe from inadvertent harm and secure from malicious actors. However, safety and security practices have historically evolved in isolation. Safety-critical systems, such as aircraft and medical devices, have long been analyzed for problems that could arise accidentally or from component degradation. They have been considered standalone systems, however, that were impervious to security issues because they had no networking capabilities. Security research, on the other hand, focused to a large extent on low-level issues (e.g., buffer overflows) and often did not make explicit connections to safety goals. There is a growing understanding that the safety and security communities are not well coordinated, and a growing recognition that this disconnect is harmful. In this blog post, we describe how research on safety and security engineering at the SEI is being applied to improve this coordination.

Modern critical systems, such as the CH-47F Chinook, TARDEC Autonomous Truck, and Little Bird, must be shown to be both safe and secure, but this is proving challenging as they are also increasingly complex. Indeed, the pace and scale of development of these systems makes the traditional safety and security analyses cost prohibitive. At the SEI, we are developing software and processes that use a system's *architecture* as the starting point for assessing and improving safety and security.

Our work in this area is largely based on AADL, the internationally standardized Architecture Analysis and Design Language. Standardized under the auspices of SAE International, AADL is a modeling language that has been adopted throughout industry, particularly in the aeronautics sector for modeling and analyzing embedded computing-based cyber-physical systems. SEI has been the primary driver behind the language for its entire 15-year existence.

Systems can, of course, be represented in myriad formats in addition to AADL, such as box-and-line diagrams on a whiteboard, Unified Modeling Language (UML) or Systems Modeling Language (SysML) diagrams, or a large list of requirements. Each of these formats has different strengths and weaknesses, but AADL excels in the precision of its specification format targeting embedded computing systems. When developing modern cyber-physical systems that are both safety and security critical, it is imperative to be as precise as possible to minimize misunderstandings among various stakeholders, users, and system builders.

The SEI also maintains the reference implementation of AADL tooling, the Open Source AADL Tool Environment (OSATE). OSATE allows users to define

a model of their embedded computing systems in AADL and virtually integrate all the system elements. OSATE also comes with built-in analyses that can help identify timing issues or failures before deployment, thereby significantly reducing costs. One study from the System Architecture Virtual Integration (SAVI) initiative, led by the Aerospace Vehicle Systems Institute (AVSI), found that 80% of issues with embedded software systems are currently discovered after unit test, and their correction consumes 50% or more of the total system-development cost.

The SEI is developing an integrated approach to safety and security engineering based on MIT's and Kansas State University's work with systems theory and supported by an AADL-based workbench. This approach

- unifies safety and security analysis through a formalized taxonomy that is used to drive system verification via fault injection and simulation
- provides a design framework to combine safety and security mechanisms into a robust and resilient system architecture through continuous analytic verification
- ensures traceability by linking machine-readable requirements to the tests that verify them and the system elements that implement them

The U.S. Army has been a close collaborator in much of our architecture work, and AADL/OSATE are at the core of that collaboration, as discussed below.

SEI Collaboration with U.S. Army

We have applied AADL-based solutions in the Army Joint Multi-Role Rotorcraft (JMR) program, where contractor teams are piloting the Architecture-Centric Virtual Integration Process (ACVIP) as a key technology on a mission-critical system architecture. Our ongoing partnership with U.S. Army Aviation is a good example of how the SEI transitions its research results to influence key DoD programs, such as the program building the Army's next-generation rotorcraft fleet, Future Vertical Lift (FVL).

The U.S. Army Futures Command is developing future readiness at a time when technology is often key to preventing and mitigating threats. However, modernizing the Army and moving quickly can be daunting in a field where every piece of new technology must be exhaustively verified and validated to ensure the support of warfighters in mission-critical situations. Agile processes have been widely adopted within the DoD as a way of rapidly

fielding capabilities to soldiers. Agile adoption is challenging, however, in systems that must be certified and qualified to meet the level of safety and security needed to deploy new technologies in the field.

Determined to create a better approach to building these critical embedded computing systems, the Army Futures Command sought help from the SEI; the solution turned out to be the application of AADL. Using AADL, with its strong semantics, introduces the capability to find and fix embedded computing-system integration problems earlier, significantly reducing the costs involved with fixing errors after software and hardware are developed. The viability of AADL has been demonstrated in a number of pilot projects on new and existing systems including the Defense Advanced Research Projects Agency (DARPA)-funded research programs, such as High-Assurance Cyber Military Systems (HACMS) and Cyber Assured Systems Engineering (CASE); the international commercial-aircraft industry-consortium SAVI initiative; and the European Space Agency (ESA) with participating contractors.

AADL has turned the risks associated with integration into an opportunity to improve. In the case of the CH47F Chinook, a heavy-lift cargo helicopter supporting Army combat, the SEI performed a virtual-integration analysis of the integrated vehicle health-management system (IVHMS) for the CH-47F several months before the contractor would have been able to integrate components in practice. By addressing identified integration problems before the critical design review (CDR), this virtual modeling and integration helped reduce the integration schedule by an estimated 12 months, which was half the projected timeframe.

Looking Ahead

DoD contractors are using the same tools that researchers use, including the AADL language and OSATE software. This convergence enables a more rapid transition from academic prototypes to mission-ready software. As we continue our research into the integration of safety and security engineering for mission-critical systems, we are investigating the following questions:

- **Near-term**—What assumptions underlying technologies that support increasing levels of autonomy (i.e., machine learning [ML], artificial intelligence [AI]) can we describe using AADL? How should AADL and OSATE be extended to support this technology?
- **Mid-term**—How can models be used at runtime? What are the connections between static, design-time models and dynamic models

used while a system is operating?
- **Long-term**—To what extent can we use ML/AI to help develop models rather than the other way around?

We are also looking for more sponsors to try out our tools, or just tell us their challenges with critical and embedded-system development. Please reach out!

Additional Resources

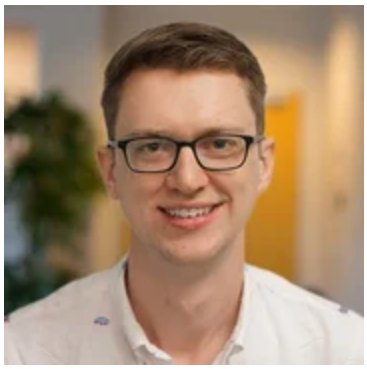
Read the SEI special report, Architecture-Led Safety Analysis of the Joint Multi-Role (JMR) Joint Common Architecture (JCA) Demonstration System.

Read the SEI technical report, ROI Analysis of the System Architecture Virtual Integration Initiative.

Read other blog posts about AADL and virtual integration.

Watch a webinar about Architecture Analysis with AADL.

WRITTEN BY



Sam Procter

DIGITAL LIBRARY PUBLICATIONS

SEND A MESSAGE

Sholom G. Cohen

DIGITAL LIBRARY PUBLICATIONS

SEND A MESSAGE

MORE BY THE AUTHORS

The OSATE Slicer: Fast Reachability Query Support for Architectural Models

NOVEMBER 13, 2023 • BY SAM PROCTER

Software Modeling: What to Model and Why

JANUARY 30, 2023 • BY JOHN MCGREGOR, SHOLOM G. COHEN

Modeling Languages for Model-Based Systems Engineering (MBSE)

NOVEMBER 21, 2022 • BY JOHN MCGREGOR, SHOLOM G. COHEN

A Model-Based Tool to Assist in the Design of Safety-Critical Systems

MARCH 7, 2022 • BY SAM PROCTER

The AADL Error Library: 4 Families of System Errors

MAY 20, 2019 • BY SAM PROCTER

MORE IN SOFTWARE ARCHITECTURE

Building Quality Software: 4 Engineering-Centric Techniques

AUGUST 19, 2024 • BY ALEJANDRO GOMEZ

The OSATE Slicer: Fast Reachability Query Support for Architectural Models

NOVEMBER 13, 2023 • BY SAM PROCTER

How to Use Docker and NS-3 to Create Realistic Network Simulations

MARCH 27, 2023 • BY ALEJANDRO GOMEZ

Software Isolation: Why It Matters to Software Evolution and Why Everybody Puts It Off

MARCH 20, 2023 • BY MARIO BENITEZ PRECIADO

Experiences Documenting and Remediating Enterprise Technical Debt

DECEMBER 19, 2022 • BY STEPHANY BELLOMO