Integrating Safety and Security Engineering for Mission-Critical Systems

Overview

We're in the second year of a three-year project that aims to make systems safer and more secure. This project consists of four efforts, all of which utilize the Architecture Analysis and Design Language (AADL), an SEI-created, internationally standardized language for designing critical systems.

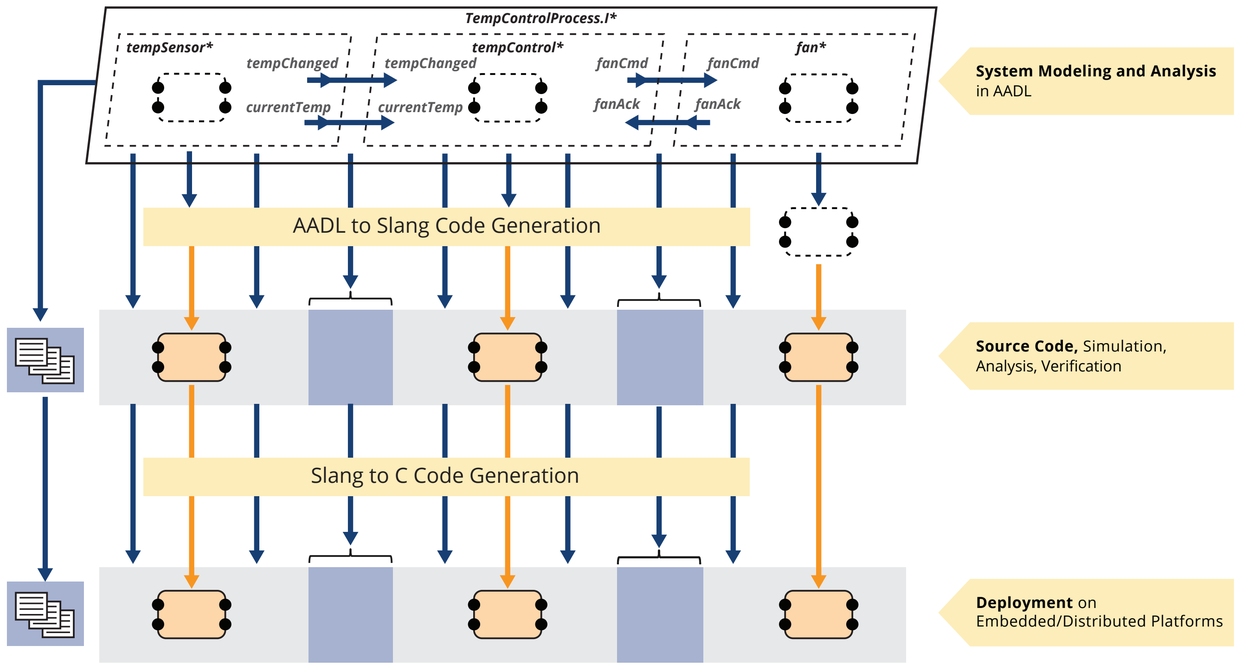
Security Annex & Patterns

There are many ways to design a system, and subtle changes can have large impacts on security and safety. One effort is looking at creating patterns—in AADL—that let our tooling automatically check for known security issues and offer suggestions for improvement.

ASAP

Performing a hazard analysis is a common way of examining a system for safety or security issues. This effort integrates a number of sources of system information—requirements, error behavior, Slang & HAMR, and more—into a set of dynamic reports. The Architecture-Supported Audit Processor (ASAP) will allow system analysts to query interesting portions of a system's architecture interactively, rather than read only what an analysis format specifies.

We're making it easier to **specify, design,** and **assure** critical systems that are safer and more secure.



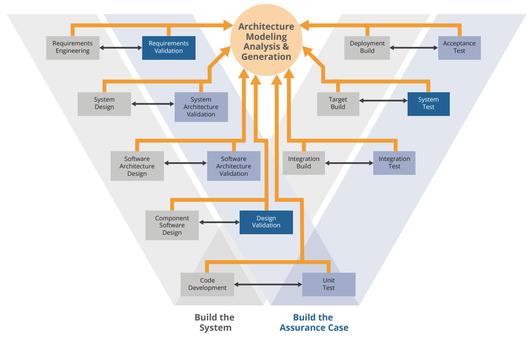
Slang and HAMR Integrate Verification with Code Generation. (Figure adapted from John Hatcliff).

Slang & HAMR

Slang, a safety-critical subset of Scala, and HAMR (High-Assurance Modeling and Rapid engineering for embedded systems) are in development by Kansas State University. These technologies support both system verification—of things like reachability properties and contract violations—and code generation to languages like C.

ALISA2

The Architecture-Led Incremental System Assurance (ALISA) project created a suite of languages and tools that let system designers specify requirements and verification activities in a machine-readable format that can be directly linked to AADL specifications. In this effort, we're updating ALISA to support the integration of other tools so system designers can use one unified interface.



Safety and Security Across the System Development Lifecycle

Carnegie Mellon University
Software Engineering Institute

Sam Procter and Peter Feiler | sprocter@sei.cmu.edu
Software Engineering Institute: Dave Gluch, Jerome Hugues, Lutz Wrage, Joe Seibel
Kansas State University: John Hatcliff, Eugene Vasserman, Robby, Hari Thiagarajan, Jason Belt

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.
The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

DM19-1035