2019

Integrated Safety and Security Engineering for Mission-Critical Systems

Dr. Sam Procter

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.

DM19-1076

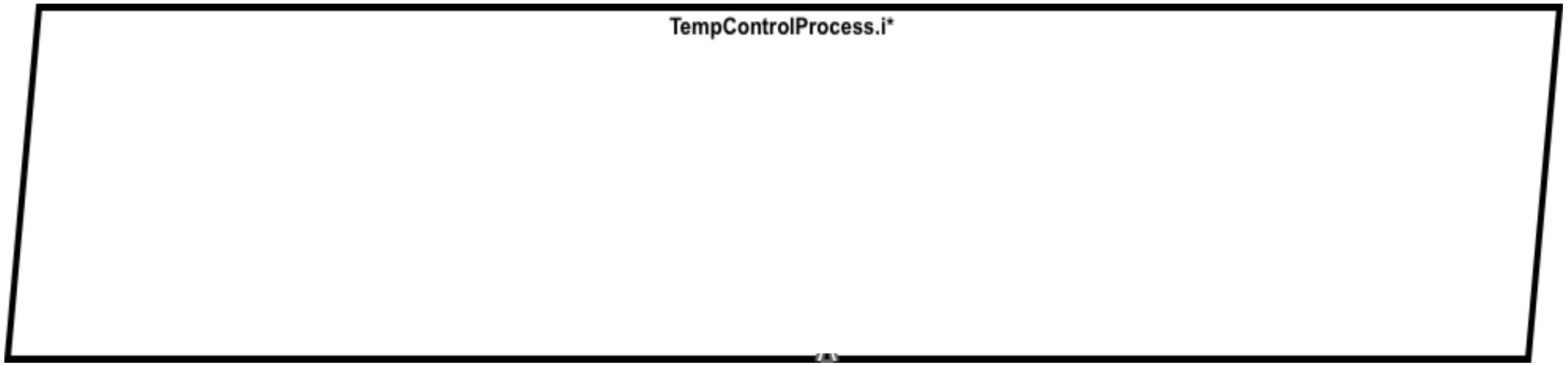
Making Critical Systems Safer and More Secure

- Modern embedded systems – such as those found in the CH47F Chinook, TARDEC Autonomous Truck, and Little Bird – need to be both safe and secure, but too often, a system's safety is designed and assessed separately from its security.
- The pace and scale of these systems' development are such that traditional analysis cannot keep up. We're developing software and processes that use a system's *architecture* to support developer intuition and improve safety and security.
- But AADL – the internationally standardized Architecture Analysis and Design Language – is for more than research: Alex Boydston will talk about how the U.S. Army is using prior research in model-based engineering to build systems that are safer and less expensive.

Integrated Safety and Security Engineering

AADL Overview

AADL Overview

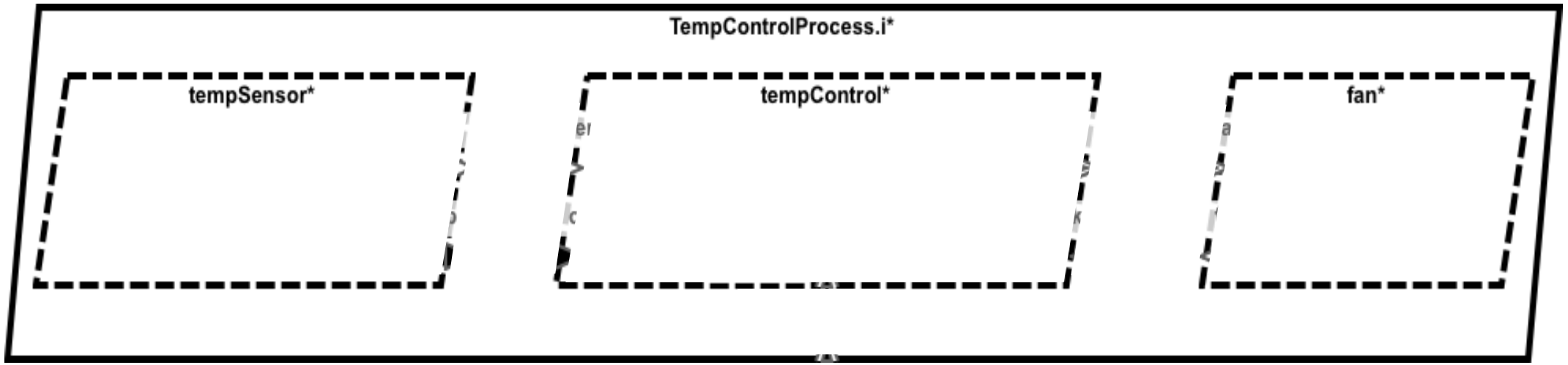


Like a lot of models that engineers draw every day on their whiteboards, AADL consists of boxes and lines

The difference between AADL and a whiteboard is that AADL has precise *semantics*

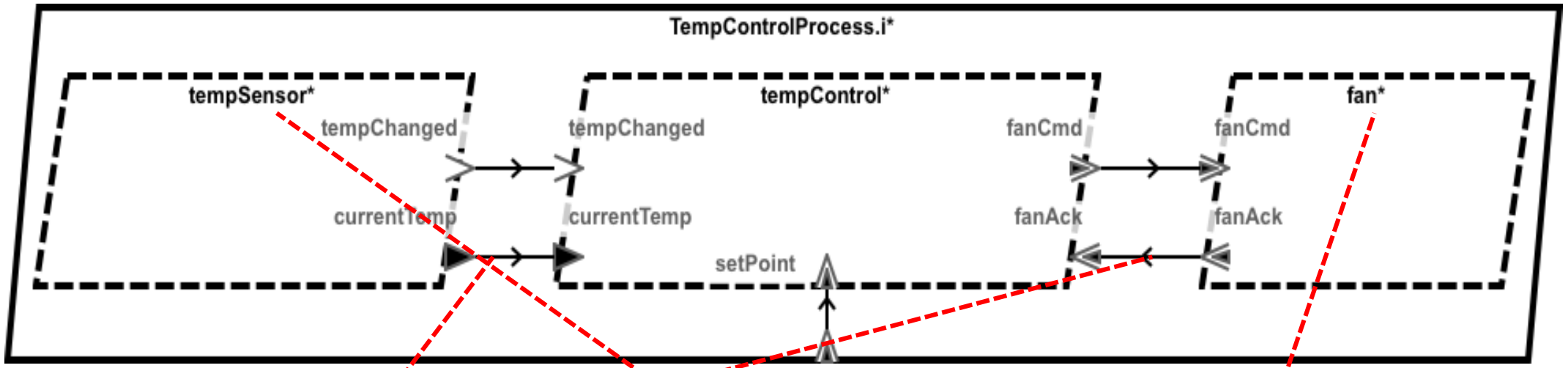
This box represents a computer process – a protected region of memory and a space where we can allocate individual threads

AADL Overview



Those threads are also boxes – but they have very precise meanings.

AADL Overview



We can connect the threads together using lines to represent different types of intra-process communication

This box shows a periodic thread – it is dispatched regularly according to some clock

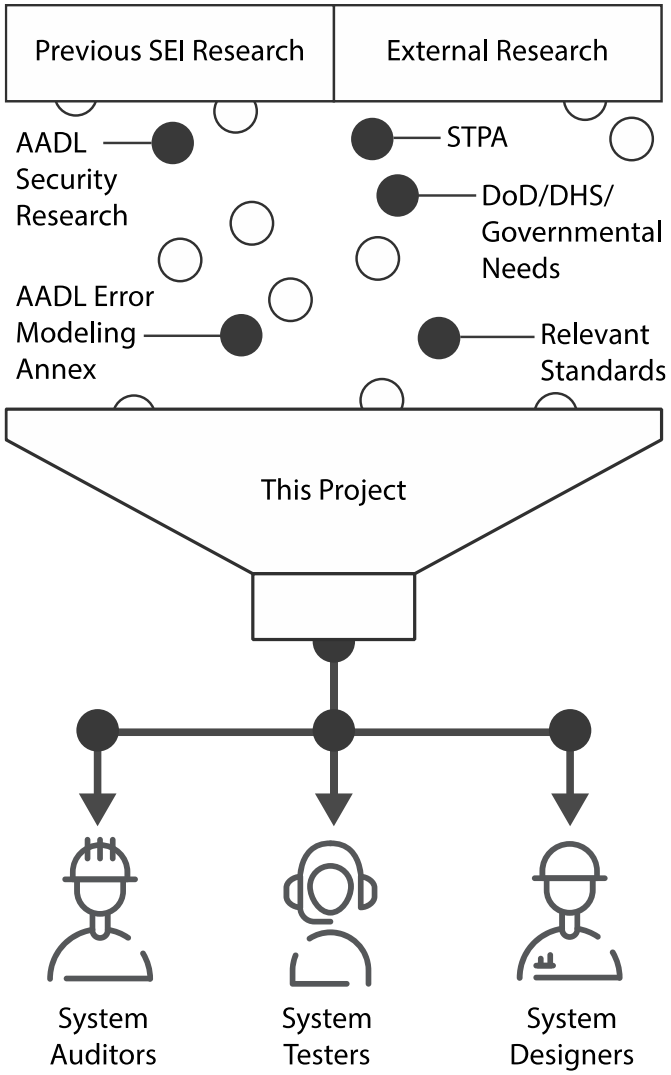
We add more semantics via *properties* – they are useful for both system analyses and to guide code generation

And this thread is sporadic – it is dispatched whenever a message arrives at a specified port

Integrated Safety and Security Engineering

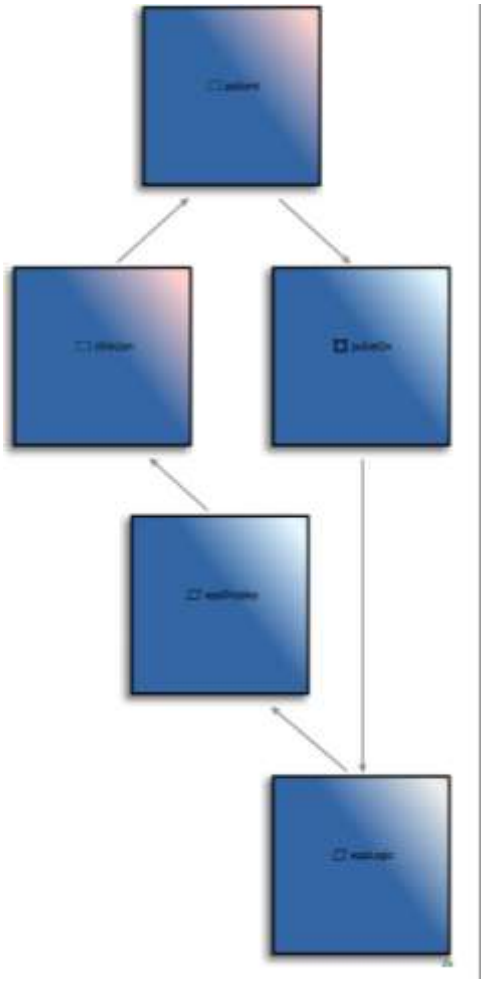
Transitioning Research to Practice

Research into Practice



- This project consists of a handful of tasks. Some are more theoretical and some more mature.
- All of the tasks, though, are implemented using AADL: a language already used by practitioners.
- This lets us rapidly move ideas from research – conducted here at the SEI, in academia, or in industry – to practice.

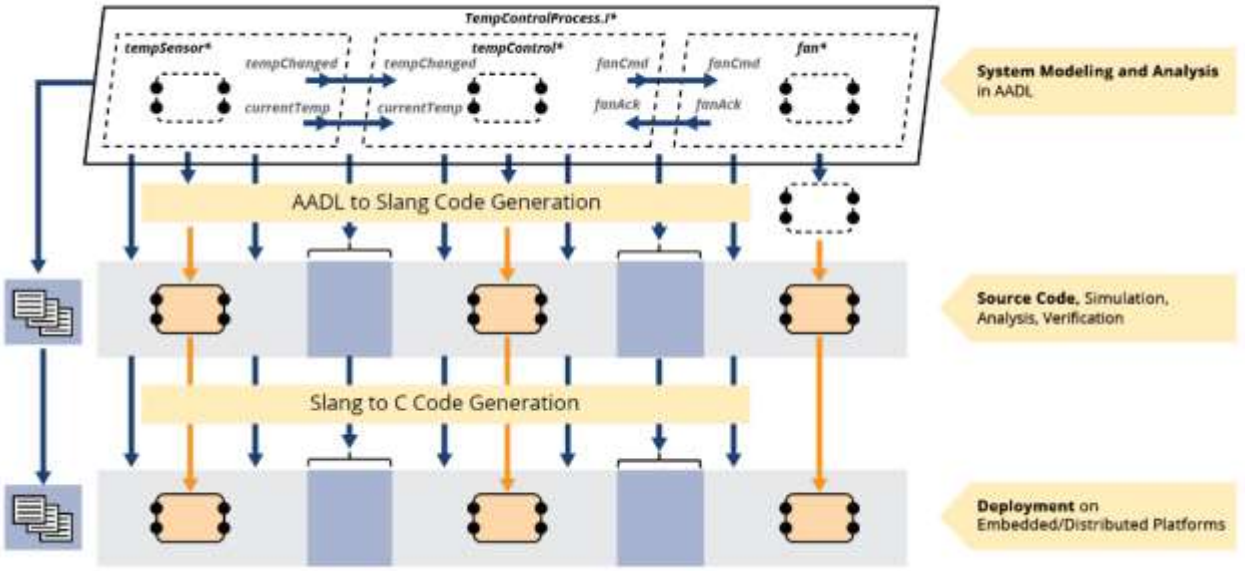
Hazard Analysis: Re-tooled for Modern System Development



- Hazard analysis – a way of assessing a system's safety – traditionally results in a large report.
- What if that report could be split into its constituent parts and
  - generated dynamically based on the system architecture?
  - queried interactively by an auditor?

We exploit state-of-the-art data-dependence analysis (developed by colleagues at Kansas State University) to power the report.

Slang and HAMR: Verification Integrated with Code Generation



We're working with Kansas State University on two related technologies that translate a system architecture (in AADL):

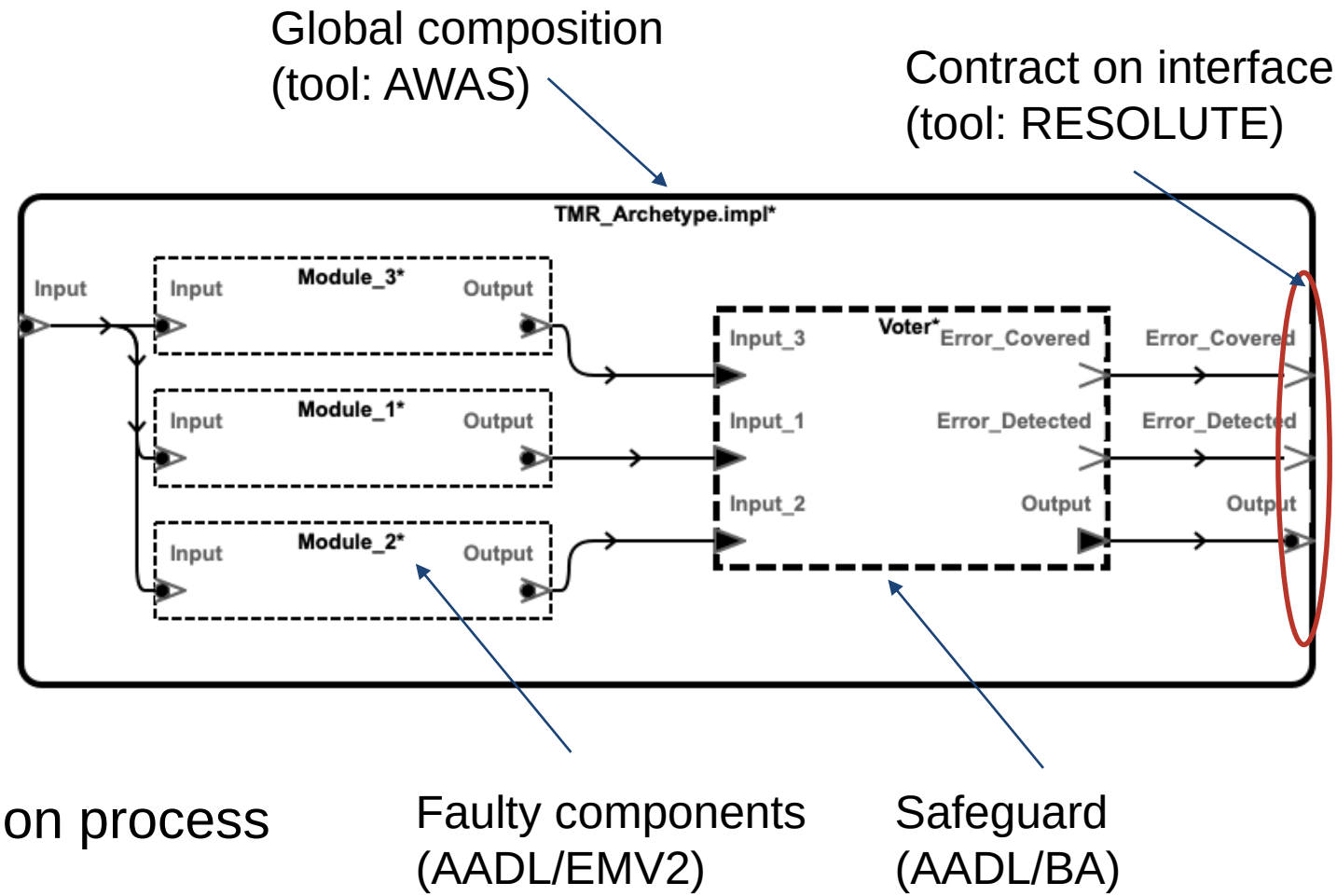
- *Slang* – an analyzable intermediate representation, and then
- *C / C++* – HAMR produces low-level source code targeted at a given platform

Derived from a model built by John Hatcliff, Kansas State University.

Safety and Security Design Patterns

Operationalize existing patterns:

- Stated in unambiguous AADL
- Machine checkable (via ALISA)



Outputs:

- A tool-supported library of patterns
- Moving through AADL standardization process

Integrated Safety and Security Engineering

DoD Impact

U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND – AVIATION & MISSILE CENTER

Architecture Centric Virtual Integration on Joint Multi-Role (JMR) Mission Systems Architecture Demonstration (MSAD)

Alex Boydston, MSEE

JMR MSAD / FARA Project Engineer

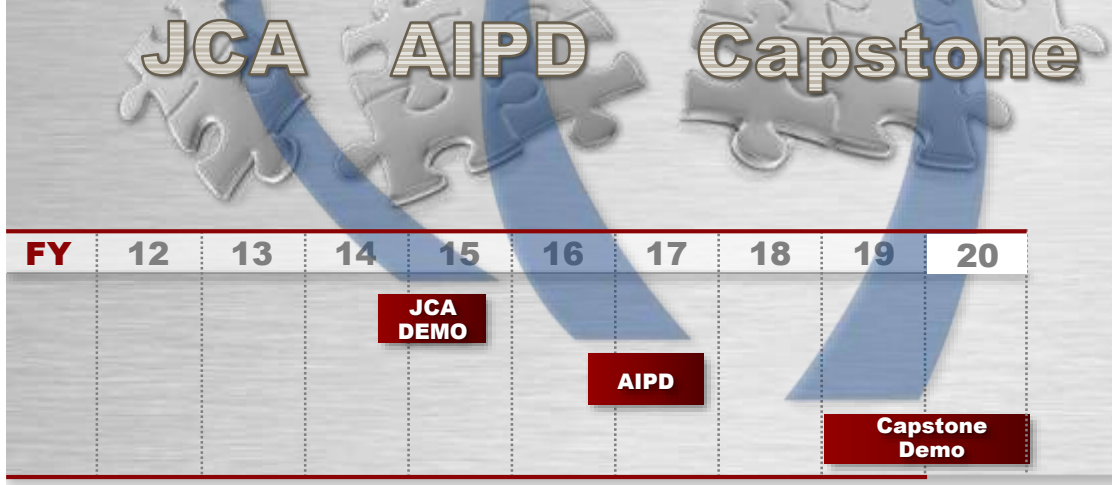
CCDC AvMC

1 Oct 2019

JMR TD MISSION SYSTEMS ARCHITECTURE DEMO (MSAD)

We need tools that help do the job, not become the job!

- **Effective Acquisition**
  - Competitive Opportunities
  - Reduced Vendor Lock
  - Increased Affordability
- **Efficient Integration**
  - Reduced Time to Field
- **Improved Capabilities**
  - Portable / Reusable
  - Interoperable
  - Upgradeable / Resilient
  - Planned Variability
  - Virtual Integration/Analysis
- **Efficient Qualification**
  - Safe/Secure



Purpose:

Investigate/Mature processes, tools and standards necessary to specify, analyze, design, implement and qualify a Mission Systems Architecture in support of emerging FVL PoR that meets Army business goals

Approach:

- Leverage or develop the standards and tools necessary to successfully implement a mission systems architecture
- Execute a series of increasingly complex demos - Learn by doing

Focus Areas:

- Implementation of Open Systems Architectures (OSA)
  - Joint Common Architecture (JCA)
  - FACE™ Technical Standard
  - Hardware Open Systems Technologies (HOST)
- Application of Model Based Engineering (MBE)
  - Model-based specification/acquisition
- Execution of an Architecture Centric Virtual Integration Process (ACVIP)
  - Predictive performance assessment

RAH-66 COMANCHE SOFTWARE REWORK & INTEGRATION COSTS

Photo Credit: Boeing-Sikorsky

Two major software (SW) rebuilds occurred during development indicating significant integration issues

- **1st increment: 75% of SW replaced**
- **2nd increment: 50% of SW replaced**

- ***In 1983, the Army planned to buy 5,023 vehicles at $12.1 million/copy.***
- ***Test schedule delays and increasing development costs scaled down the planned buy to 650 aircraft at $58.9 million/copy.***
- ***Most testing involved integration of the complete Mission Equipment Package, which incorporated a radar, infrared, and image-intensified television sensors for night flying and target acquisition.***
- ***Technical challenges remained in software development, integration of mission equipment, radar and infrared signatures, and radar perf.***
- ***The first flight had been originally planned to take place during August 1995, but was delayed by a number of structural and software problems that had been encountered.***
- ***Key program elements, including development and integration of certain software capabilities, failed to foster confidence with Army overseers; several capabilities were viewed as having been unproven and risky.***
- ***The anticipated consumption of up to 40% of the aviation budget by the Comanche alone for a number of years was considered to be extreme.***

References:

- *http://www.defense-aerospace.com/articles-view/release/3/32273/pentagon-hit-over-comanche-failings-(jan.-23).html*
- *https://en.wikipedia.org/wiki/Boeing%E2%80%93Sikorsky_RAH-66_Comanche#cite_note-26*
- *https://en.wikipedia.org/wiki/Boeing%E2%80%93Sikorsky_RAH-66_Comanche#cite_note-Eden_p139-9)*

Comanche costs were expected to consume up to 40% of US Army Aviation budget resulting in cancellation. Integration and software rework were significant cost contributors.

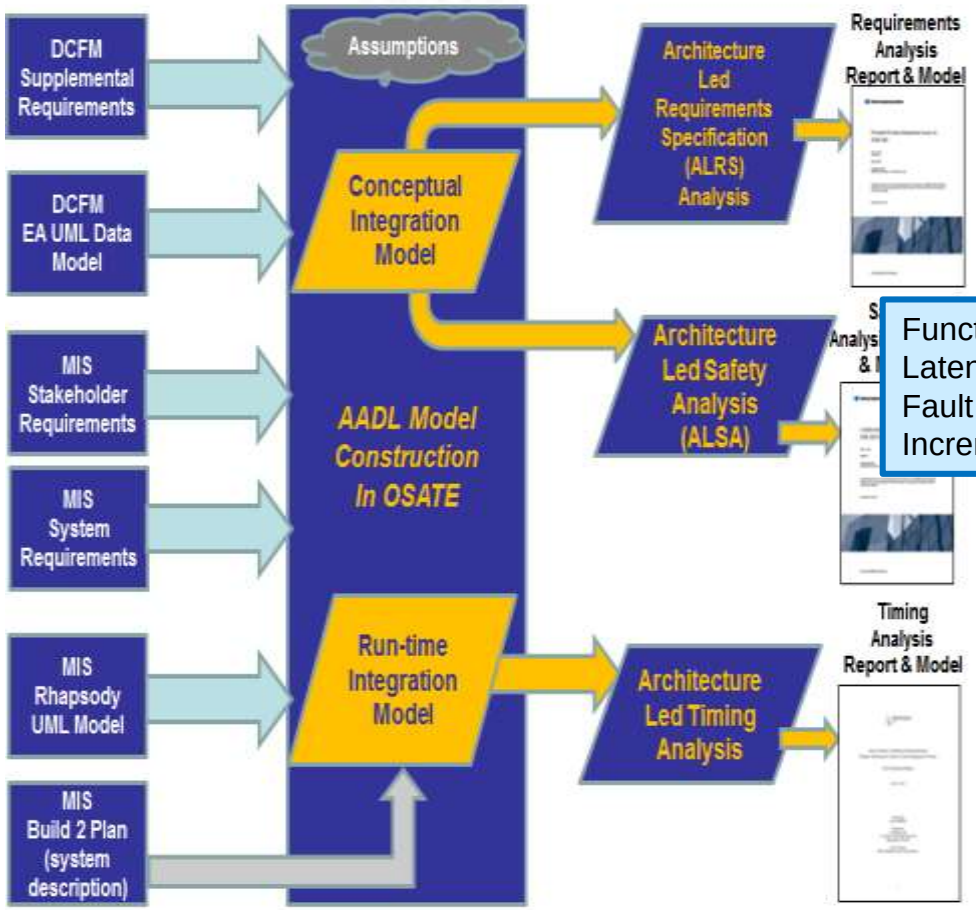
ARCHITECTURE CENTRIC VIRTUAL INTEGRATION PROCESS (ACVIP)

- Origin (2009): Aerospace Vehicle Systems Institute's System Architecture Virtual Integration (SAVI) concept for incremental virtual integration using AADL.
- First step, embedded systems architecture modeling in AADL, a language for precisely specifying key components and properties of embedded systems.
- Virtual integration process uses AADL-enabled analyses of real-time safety- and security-critical computing systems to identify issues early before integration.
- Automated continuous virtual integration enables architecture-based incremental and compositional modeling & analysis as system evolves.
- Provides increasing assurance confidence; complements testing.
- Provides a "Single Authoritative Source of Truth.
- Enabler of MOSA to provide a standard analyzable and processable architecture description for embedded systems.

"Model, Integrate, Analyze, then Build"

Virtual Integration of Software, Hardware, and System supporting verification, airworthiness, safety and cybersecurity certification

1ST JMR MSAD DEMONSTRATION: JCA DEMO LESSONS LEARNED



- **Use of AADL for virtual integration and analysis identified** >85 issues

- **ACVIP analyses identified errors prior to system integration (as early as during the kickoff meetings with DCFM suppliers)**

Functional integration analysis (SAVI 2008)
Latency jitter analysis (Line 2006)
Fault taxonomy and analysis (Line 2012)
Incremental assurance (ALISA 2015-16)

FACE->AADL and SysML->AADL would have been beneficial to automate and reduce human error

- **ACVIP training proved beneficial**
  - Boeing used AADL to extend their demo for timing and control stability analysis and found issues

Architecture analysis is critical for the successful and affordable integration of systems

DEMOS OF EFFECTIVENESS IN USE OF ACVIP & AADL

Finding Problems Early Using AADL (CCDC/SEI)

- Summary: 6 Week Virtual Integration of HUMS on CH47F using AADL
- Result: Identified 20 major integration issues early
- Benefit: Avoided 12-month delay on 24-month program

Decreased fielding time

CH47F Chinook

Architecture Concurrency (Line 2005)
Virtual Upgrade Validation Method (2012)
Error Model V2 Annex (2015)

Discovering Performance Issues Early Using AADL (UH-60V)

- Summary: Applied AADL anal
- Result:. Predicted multicore p
- Benefit: Provided early performance insight and risk reduction

Multi-core Scheduling (Line 2014)

UH60V Blackhawk

Early Risk Reduction

Improving System Security (DARPA / AFRL)

- AADL applied to Unmanned Aerial Vehicles & Autonomous Truck using formal methods analysis and trusted system generation
- Result: AADL models enforced security policies and were used to auto build the trusted system
- Benefit: Combined with formal methods verification, prevented security intrusion by a red team

AADL strong typing (2002)
MILS Security (Line 2009)
ARINC653 Annex (2011)
Resolute extension (2014)

TARDEC Autonomous Truck

Unmanned Little Bird

Increased Cybersecurity

Transforming procurement supporting MBE and ACVIP (JMR MSAD)

- Summary: Increasingly complex onstrations using Model Based Engineering
- Result: Pre-integration fault ider
- Benefit: ~3x increase to require ~10x reduction on test and integratior

MDS Reference Architecture (2010)
Reliability Validation & Improvement (2014)
AADL Workbench (Line 2015)
Incremental Assurance (ALISA 2015-16)
Integrated Safety & Security (ISSE 2018-20)

Decreased development costs, supports MOSA & certification

Makes complex capabilities possible through Agile analytic and virtual integration of real-time safety and security critical cyber physical embedded systems

Web Site
https://www.avmc.army.mil/

Facebook
www.facebook.com/ccdc.avm

Instagram
www.instagram.com/CCDC_AVM

Twitter
@CCDC_AVM

Public Affairs
usarmy.redstone.ccdc-avmc.mbx.pao@mail.mil

Integrated Safety and Security Engineering

Looking Ahead

Looking Ahead

NEAR	MID	FAR
What other assumptions underlying various emerging technologies (e.g., ML / AI, DevOps, formal verification of behavior) would be beneficial in architectural models?	How can models be used at runtime? What data do we need to more effectively let systems autonomously use models of themselves?	To what extent can we use ML / AI to help develop models, rather than the other way around?

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