
# Execution-based Verification of AADL Models

John Hudak

Software Engineering Institute (SEI)
PWP: 6-396H7

Innovative Defense Technologies, LLC
Contract Number: W911W6-19-C-0008

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

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.



DM19-1177

# Project Overview - Execution-based Verification of AADL Models

A multi-phase SBIR - Objective

*"Reduce aviation mission system integration testing time and effort and increase assurance by developing testing tools that support a model-based system development process.*
***Develop a software tool that will check instrumentation data collected from an integrated mission system to see if the observed system behaviors of an integrated mission system conform to required and allowed behaviors defined in an Architectural Analysis and Design Language (AADL) model of the integrated aviation software and hardware mission system**."*

Teaming With Industry: SEI & Innovative Defense Technologies (IDT)

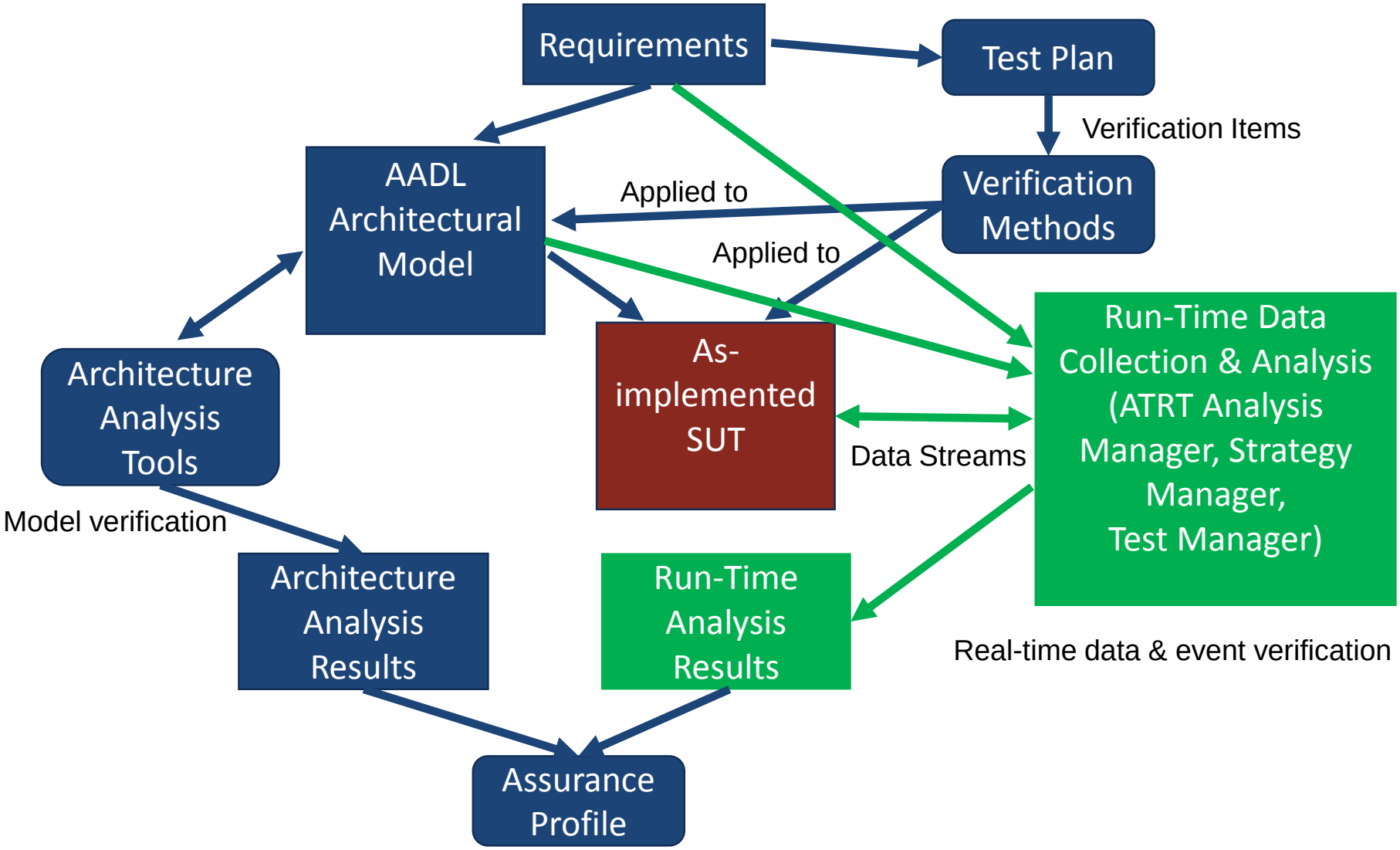
- SEI – AADL and Model-based engineering SME's
- IDT- Tool environment: Automated Test and Re-Test (ATRT) test environment.
  - Translate AADL models into a format usable by ATRT
  - Leverage ATRT existing infrastructure capabilities (based on SysML)

Phase 1 – Proof of concept

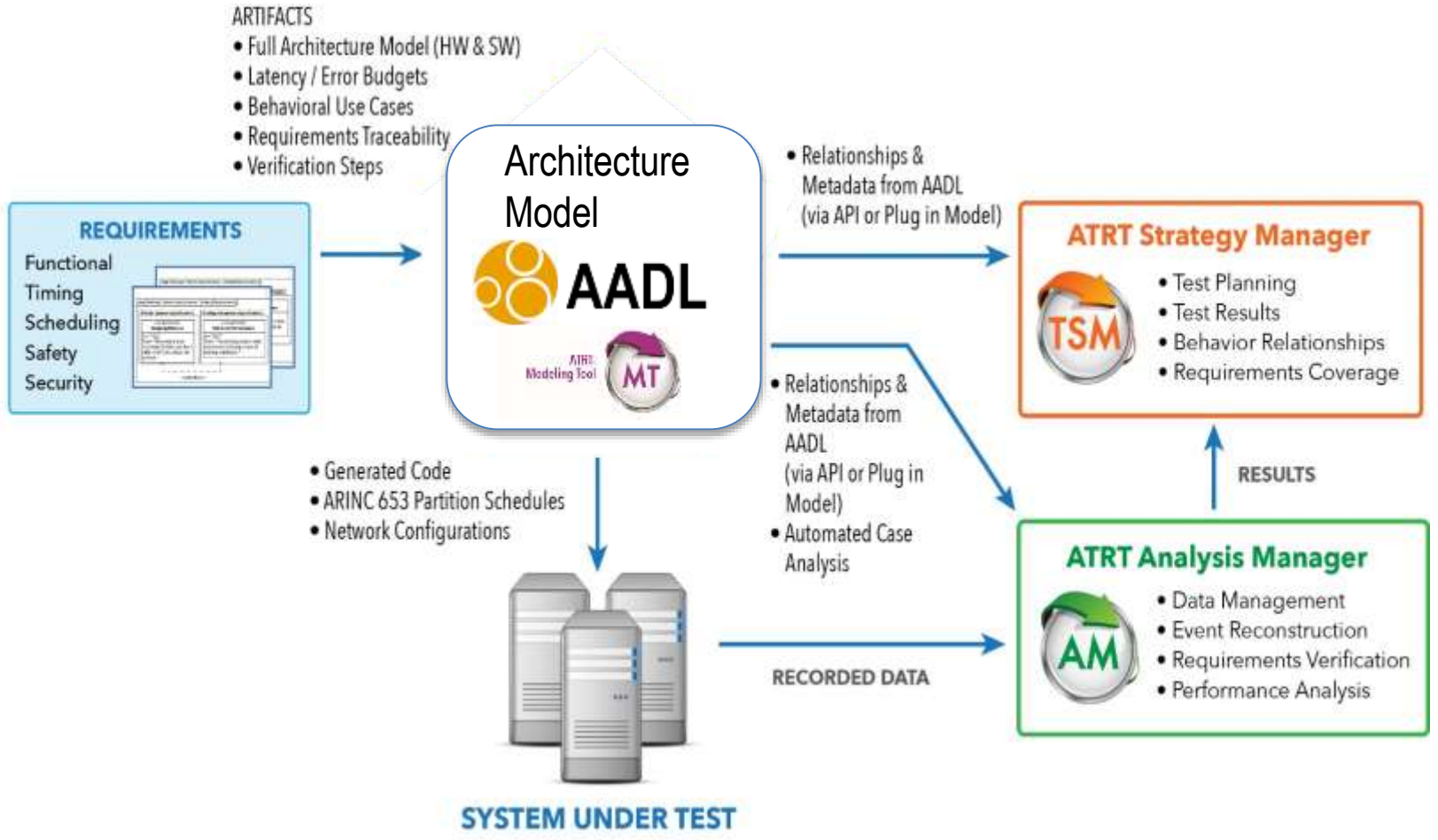
Phase 2 – Expanded development, produce TRL 6 capable tooling

Phase 3 – Improving TRL level, application on real systems

# Requirements Verification Flow



# Overall Approach




Model-Based Automated V&V Concept for Model Driven Testing (MDT)

# Capabilities Demonstrated – Phase 1

Demonstrated  automated verification of system properties from (simulated) system execution data to the same properties in the corresponding architecture model and analysis, in three key areas

Performance

- End to End flow paths with latency analysis – multiple views
- Modal behavior of runtime elements (process, threads)
- Bus communication bandwidth analysis
- System physical analysis (Power bus capacity, weigh analysis)
- Resource utilization computing loads (memory, CPU)
- Error flow (detection, handling)

# Capabilities Demonstrated – Phase 1

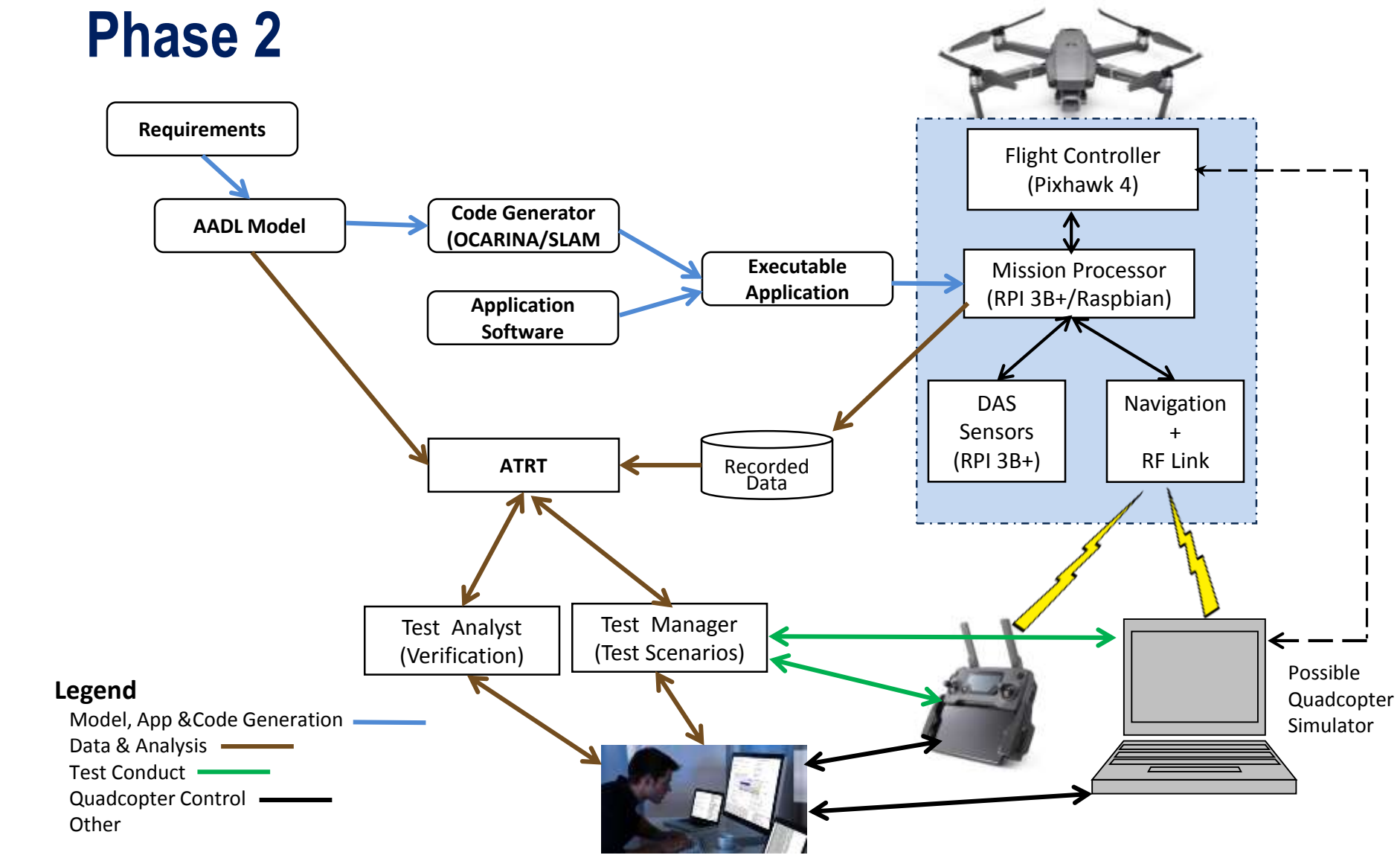
Safety

- Verification of properties in support of ARP-4761 safety assessment ( Functional Hazard Analysis, Prelim System Safety Assessment, System Safety Assessment)

Security

- Verification of properties based on Microsoft STRIDE threat model (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege) – Demonstrated verification of Information Disclosure, Elevation of Privilege, remaining properties currently in development.

# Model-based Verification – Testbed Flows Phase 2



# Technical Challenges Being Addressed

AADL component, characteristics, and semantics translated to ATRT

- Supporting multiple architecture views (logical, runtime, deployment)
- Concepts of data and event flows through views

System under test instance

- Data acquisition & recording
  - What to record, how to record, loading effects
- System and variable state and temporal consistency
  - AADL code generation approaches
  - Implementation of runtime services to ensure thread semantics
  - Quantifying/bounding temporal measurements

# Impact to DoD

This approach ‘closes the verification loop’ established by using a model-based architecture-centric approach to embedded systems development.

The method provides a way to verify system requirements, in an automated way, beginning with the architectural model and continuing to verify the same requirements from the corresponding executing system of the model.

Ramifications on overall system requirements verification process

- Requirements verified  from architectural model analysis to system implementation
- Automation of this approach can reduce system testing time, repeatable, precise
- Produces Objective Quality Evidence (OQE) of results based on test conducts and instrumented system execution –
- Architectural analysis finds errors early in lifecycle (reduces cost), execution testing ensures translation integrity, reduced testing time.

# Contact Information


## John J. Hudak

Senior Member of the Technical Staff
Telephone: +1 412.268.5219
Email: jhudak@sei.cmu.edu

**U.S. Mail:**
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890