Automated Test and Re-Test (ATRT)

Model-Based Testing (MBT) of Integrated Aviation Mission Systems

"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."

Shawn Kline, IDT
skline@idtus.com
609-313-3017

John Hudak, SEI
jhudak@sei.cmu.edu
412-268-5219

Andrew O'Neill, IDT
AO'Neill@idtus.com

Small Business Innovative Research (SBIR) Topic Number: A17-006

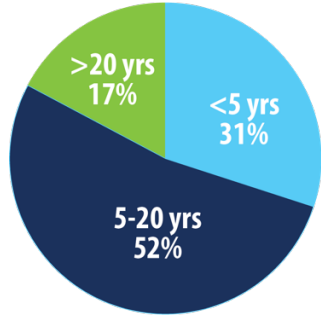Award/Contract Number: W911W6-18-C-0010

Agenda

- Background
  - IDT / Automated Test and Re-Test (ATRT)

- Technology Overview
  - Leveraged from ATRT SysML MBSE approach (TRL 8/9)
  - Currently being extended into AADL via A17-006 SBIR
  - AADL to ATRT interface

- Follow-Up Conversations & Discussion available upon request.
  - Demonstrations Available
    - AADL to ATRT feature list and accomplishments
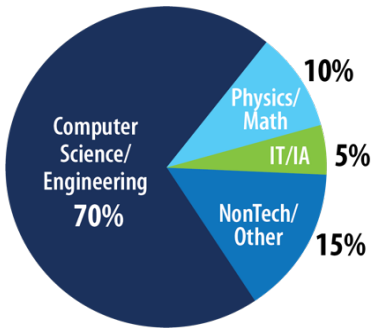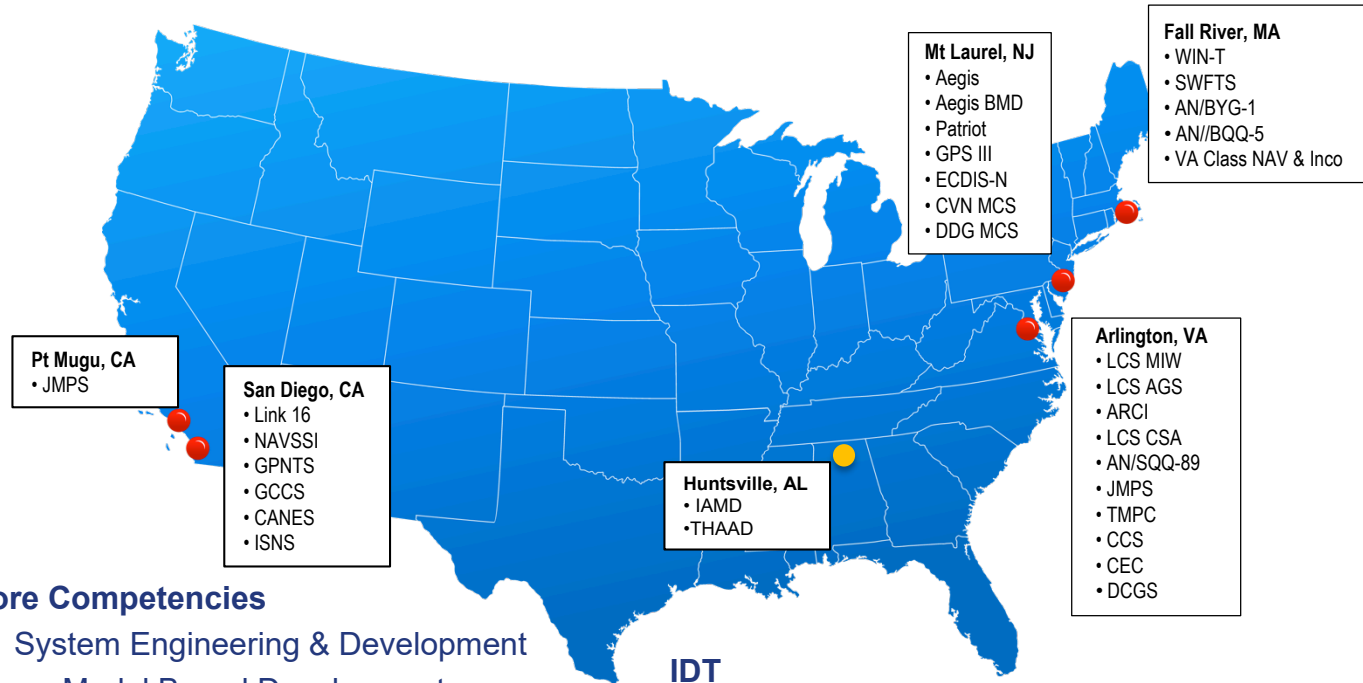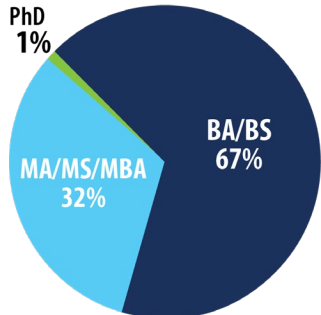   - Automated analysis from AADL model demonstration

Carnegie Mellon University
Software Engineering Institute

IDT at a Glance

Relevant Experience

- >20 yrs 17%
- <5 yrs 31%
- 5-20 yrs 52%

Degrees

- Computer Science/Engineering 70%
- Physics/Math 10%
- IT/IA 5%
- NonTech/Other 15%

Advanced Degree

- PhD 1%
- MA/MS/MBA 32%
- BA/BS 67%

Pt Mugu, CA
- JMPS

San Diego, CA
- Link 16
- NAVSSI
- GPNTS
- GCCS
- CANES
- ISNS

Huntsville, AL
- IAMD
- THAAD

Mt Laurel, NJ
- Aegis
- Aegis BMD
- Patriot
- GPS III
- ECDIS-N
- CVN MCS
- DDG MCS

Fall River, MA
- WIN-T
- SWFTS
- AN/BYG-1
- AN//BQQ-5
- VA Class NAV & Inco

Arlington, VA
- LCS MIW
- LCS AGS
- ARCI
- LCS CSA
- AN/SQQ-89
- JMPS
- TMPC
- CCS
- CEC
- DCGS

Core Competencies

- System Engineering & Development
  - Model Based Development
  - Automated Analysis/Assessment
  - Continuous Integration
  - Virtualization
  - SoS Interoperability
- Software Re-Engineering
  - Automated SW Re-Architecture
- Test & Integration
  - Automated Test & Analysis
- Cyber Hardening, Threat Detection & Recovery

IDT

- CMMI Level 3
- Recognition
  - Contractor of the Year (Small & Emerging Contractor Advisory Forum—SECAF)
  - 50 Fastest Growing Businesses (Washington Business Journal)
  - 20 Most Promising Solution Providers (CIO Defense Technology Review)

3

Carnegie Mellon University
Software Engineering Institute

Automated Test and Re-Test (ATRT)

ATRT is a TRL 8/9 technology developed under the SBIR program to achieve the following goals:

- Provide a methodology and capabilities able to used across an Enterprise and throughout the lifecycle
- Significantly reduce the time and manpower required for testing and in particular regression testing
- Support the efficient characterization of a system's performance envelope
- Provide significant objective quality evidence across the development and test lifecycle for system certification
- Provide collaborative test strategy capability for test planning, test status, and test reporting across distributed test resources

Enable Capability to be Delivered Faster and More Efficiently to the Warfighter

Carnegie Mellon University
Software Engineering Institute

IDT

What is Different With ATRT ?

Historical Testing Approach

- Repetitive human labor intensive testing
- Time constrained
- Limited permutation testing
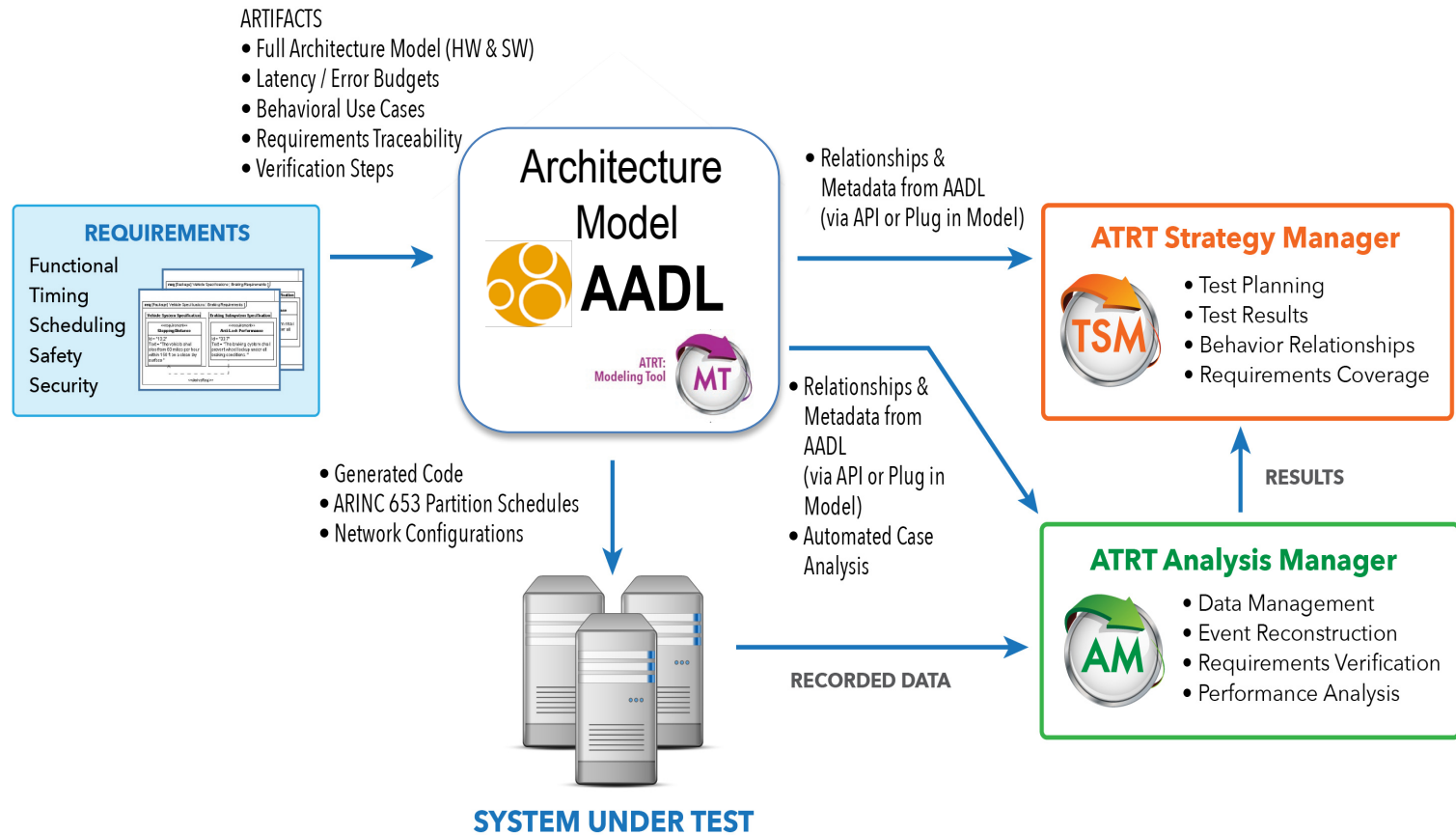- Evaluation of test results requires days or weeks

ATRT Testing Approach

- Automated computer based testing
- 24/7/365 testing
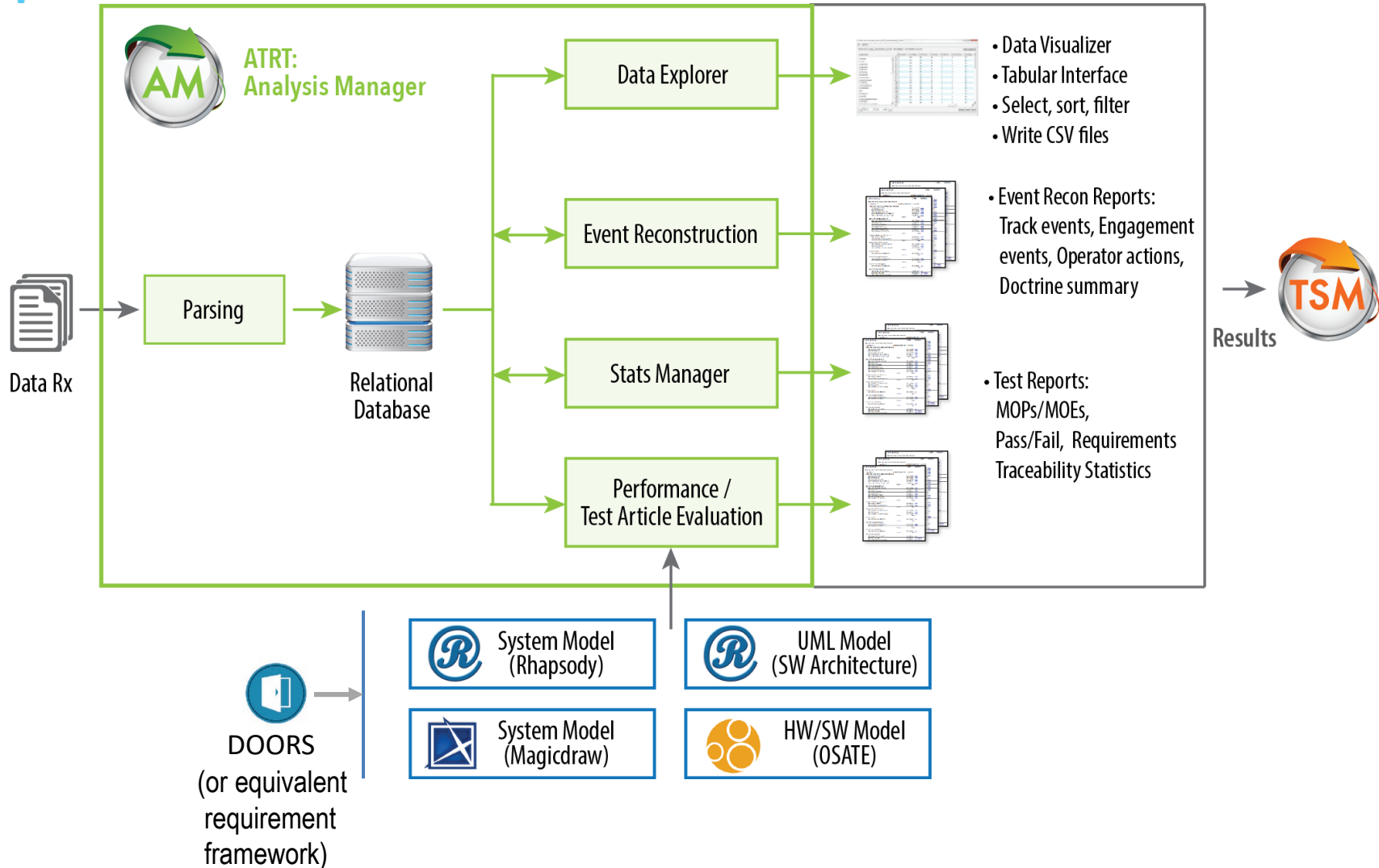- Expansive permutation testing
- Test results available in minutes

Technical Approach

ARTIFACTS
- Full Architecture Model (HW & SW)
- Latency / Error Budgets
- Behavioral Use Cases
- Requirements Traceability
- Verification Steps

REQUIREMENTS

Functional
Timing
Scheduling
Safety
Security

Architecture Model
AADL

ATRT: Modeling Tool
MT

- Relationships & Metadata from AADL (via API or Plug in Model)

ATRT Strategy Manager

TSM
- Test Planning
- Test Results
- Behavior Relationships
- Requirements Coverage

- Generated Code
- ARINC 653 Partition Schedules
- Network Configurations

- Relationships & Metadata from AADL (via API or Plug in Model)
- Automated Case Analysis

RESULTS

SYSTEM UNDER TEST

RECORDED DATA

ATRT Analysis Manager

AM
- Data Management
- Event Reconstruction
- Requirements Verification
- Performance Analysis

ATRT/AADL Model-Based Testing (MBT)

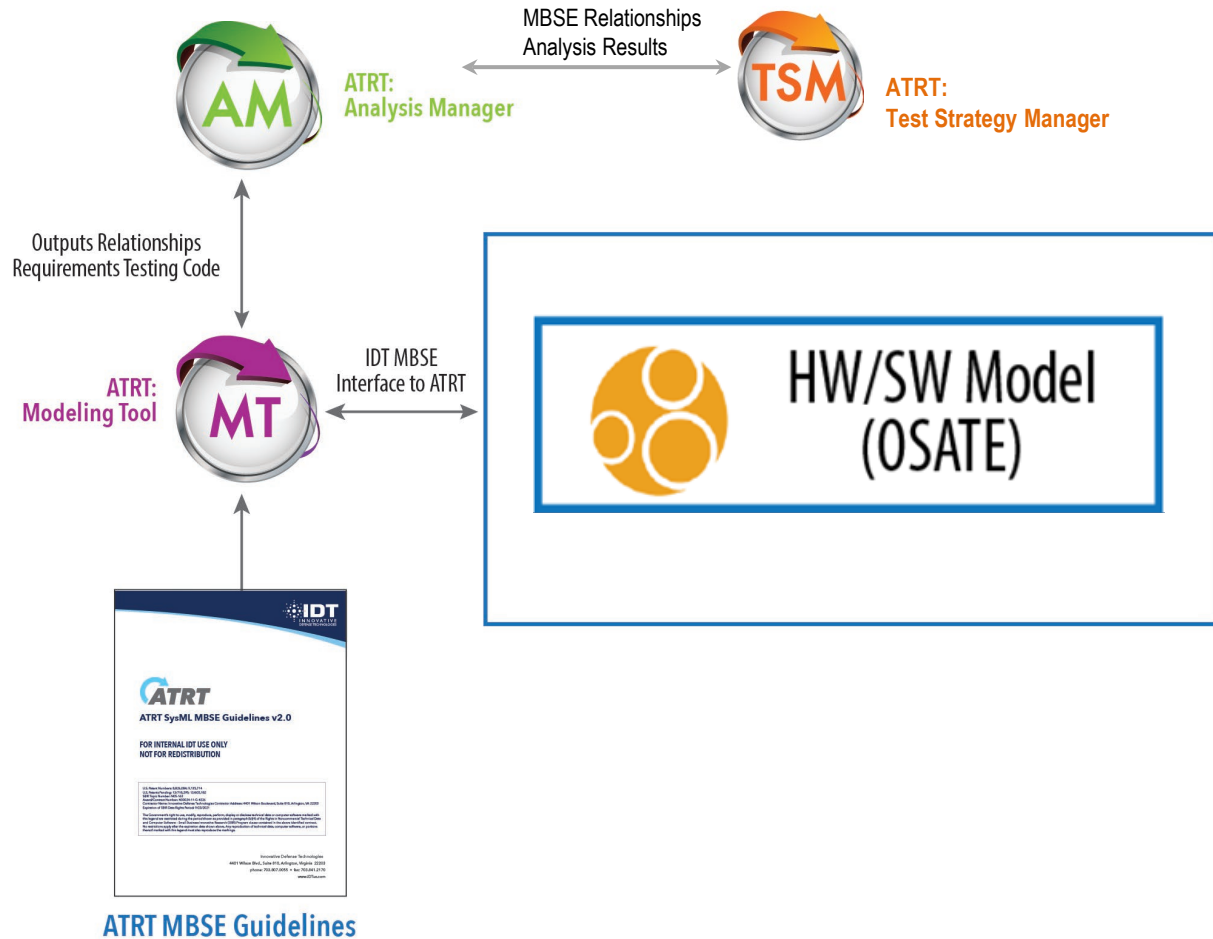**Software Engineering Institute** | **Carnegie Mellon**

IDT

ATRT MBSE Block Diagram

AADL / ATRT Interfaces

- MT plugs into an existing AADL model through OSATE and will traverse the current model in the OSATE workspace

- Walks the Instance Model to pull out information from the flows and the AADL objects contained in each flow

- From the Instance Model information MT automatically generates C++ files and database files AM can consume

- These auto generated files serve as a map for Analysis Manager's analysis engine to perform analysis on the system's recorded data

MBSE Relationships Analysis Results

AM ATRT: **Analysis Manager**

TSM ATRT: **Test Strategy Manager**

Outputs Relationships Requirements Testing Code

ATRT: **Modeling Tool** **MT**

IDT MBSE Interface to ATRT

HW/SW Model (OSATE)

ATRT
ATRT SysML MBSE Guidelines v2.0

FOR INTERNAL IDT USE ONLY
NOT FOR REDISTRIBUTION

ATRT MBSE Guidelines

8

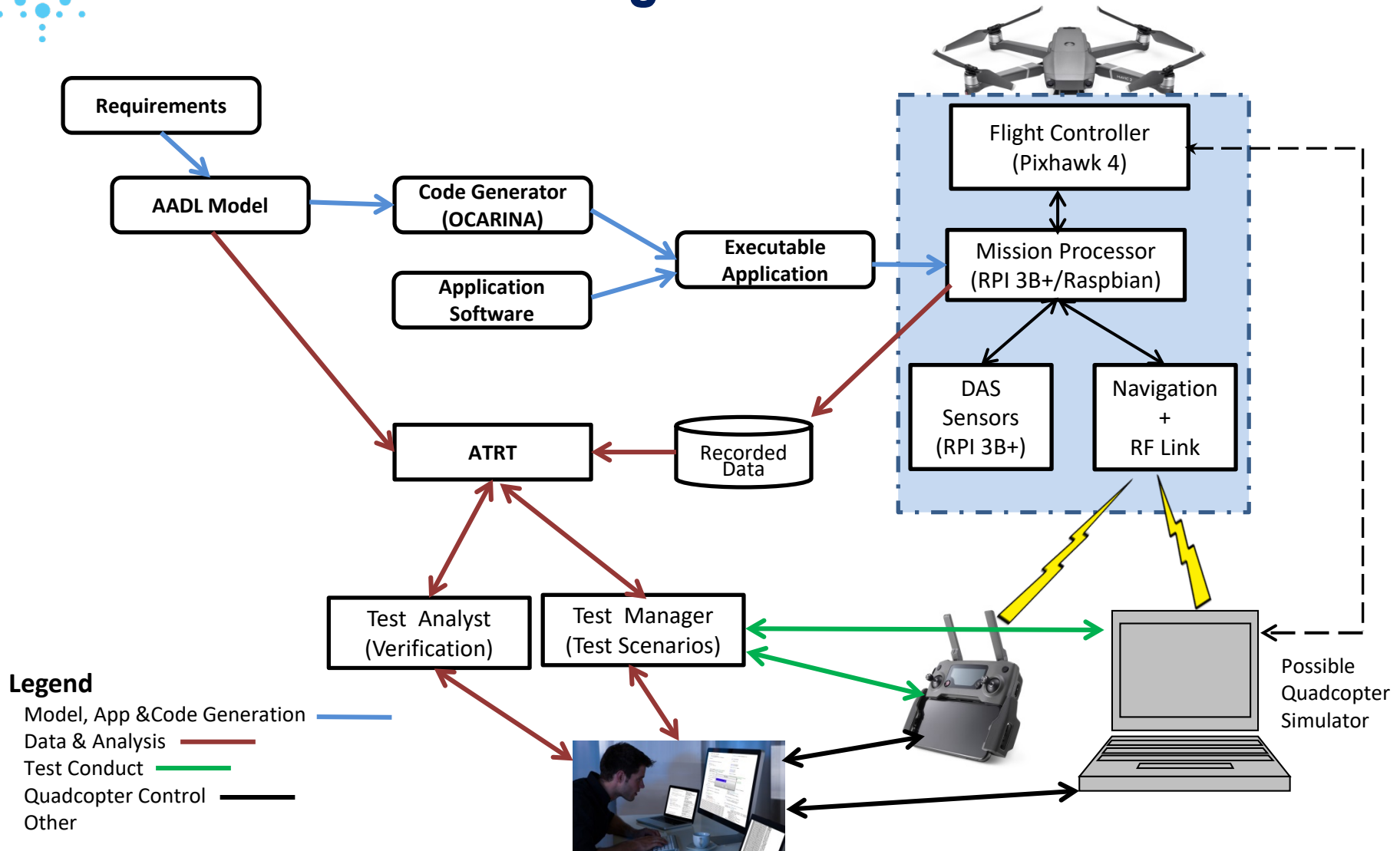**Carnegie Mellon University**
Software Engineering Institute

Technical Status

- Initially identified and prototyped the following as characteristics that can be leveraged from AADL to ATRT elements:

  ✓ a) End to End flow (of data, events, or both)
  ✓ b) Latency (between/through logical components, execution of threads)
  ✓ c) Modes attached to threads
  ✓ d) Communication bus bandwidth (worst case loads, scheduled loads)
  ✓ e) Power bus capacity (power)
  ✓ f) Resource utilization of bound loads (memory, CPU)
  ✓ g) Error flow (ensure error types are handled/mitigated)
  ✓ h) Functional hazard analysis
  ✓ i) Fault tree analysis
  ✓ j) Security (partial)
  ✓ k) Data access & Subprograms*

Carnegie Mellon University
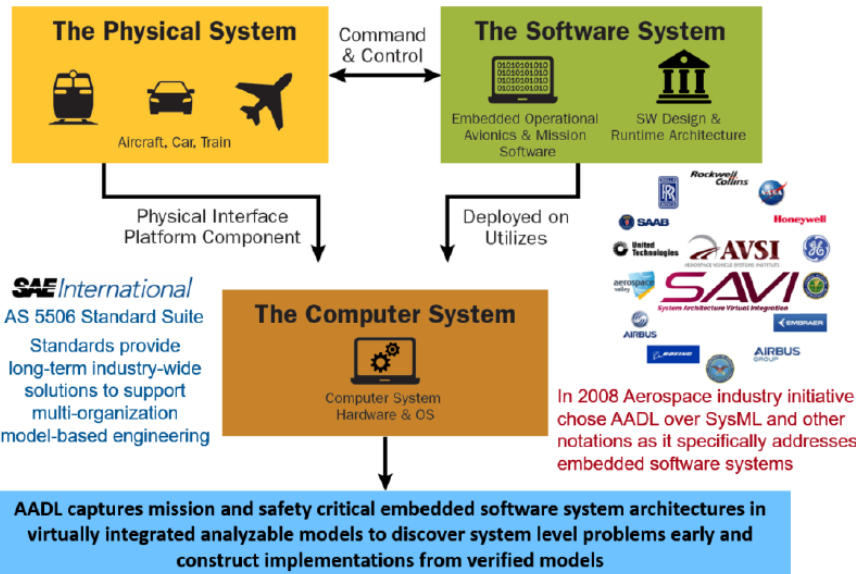Software Engineering Institute

Model-based Testing – Platform/Testbed Flows

Legend
- Model, App &Code Generation ——— (blue)
- Data & Analysis ——— (red)
- Test Conduct ——— (green)
- Quadcopter Control ——— (black)
- Other

Carnegie Mellon University
Software Engineering Institute

ATRT Demonstration (Available Upon Request)

- Walkthrough created AADL model

- Ocarina generating PolyOrb-HI-C application software

- Executable application running on processor with software probe collecting data

- ATRT analysis on recorded data with Objective Quality Evidence (OQE) reports

Increased Confidence through Continuous Verification and Testing

Development Workflow

Requirements

Properties for Verification

Analysis: Latency, Execution Time. Etc.

Develop AADL Model

OSATE & OCARINA

Model Analysis Reports

Generated Skeleton Code Wrappers

Application Code in C

Geany, CDT, or CodeBlocks

GNU Compiler Collection In IDE

Combine Code Wrappers + Functional Code

Executable Image

Deploy and Run on Target Computer

Data of Interest

Recorded Data

Data Files

ATRT

Runtime Verification Reports

12

Collaboration

- IDT and SEI seek to collaborate with aviation platform providers in facilitating ATRT pilot efforts

  - Demonstrate value of ATRT/MBSE approach via Automated Test & Assessment utilizing post test data

  - Understand design issues discovered in test on overall system capability

  - Provide assessment of system behavior relative to requirements

  - Increase efficiency in failure resolution process

  - Produce quantitative Objective Quality Evidence (OQE) to support system Sell-Off, Certification and Caps and Limits

Carnegie Mellon University
Software Engineering Institute

Summary

- Phase I&II SBIR - demonstrated model-based testing applicability to all analysis in AADL/OSATE

- Demonstrated an end-to-end automated tool chain from models to runtime implementation on a representative UAV system

- Provides verification support that requirements are met by the runtime system as captured and analyzed in the architectural
  - More than just requirements tracing!

- Tooling supports ACVIP through system implementation

- Successfully leveraged a TRL 8/9 verification tool (ATRT) that is currently being used in multiple DoD Programs of Record

- As part of Phase 3 of SBIR, looking for partners to exercise & expand capabilities to address program specific needs

Carnegie Mellon University
Software Engineering Institute

IDT

Questions

Carnegie Mellon University
Software Engineering Institute

Acronyms

- AADL – Architecture Analysis and Design Language

- AM – Analysis Manager

- ATRT – Automated Test and Re-Test

- CPU – Computer Processing Unit

- DAS – Digital Acquisition System

- DoD – Department of Defense

- GNU – GNU's Not Unix

- IDE – Integrated Development Environment

- IDT – Innovative Defense Technologies

- MBSE – Model Based Systems Engineering

- MBT – Model Based Testing

- MT – Modeling Tool

- OSATE – Open Source AADL Tool Environment

- OQE – Objective Quality Evidence

- RPI – Raspian Pi Imager

- SEI – Software Engineering Institute

- SBIR – Small Business Innovative Research

- TRL – Technology Readiness Level

- TSM – Test Strategy Manager

- UAV – Unmanned Aerial Vehicle

Carnegie Mellon University
Software Engineering Institute