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

Architecture Centric Virtual Integration Process Overview

Alex Boydston, MSEE

JMR MSAD S&T Project Engineer

Aviation Development Directorate JMR Technology Demonstrator (TD) Program

APPROVED FOR PUBLIC RELEASE


Deliver collaborative and innovative aviation and missile capabilities for responsive and cost-effective research, development and life cycle engineering solutions.

~9,553	2,943	23	6,587
FY18 Strength	Civilian	Military	Contractor

FY18 Funding

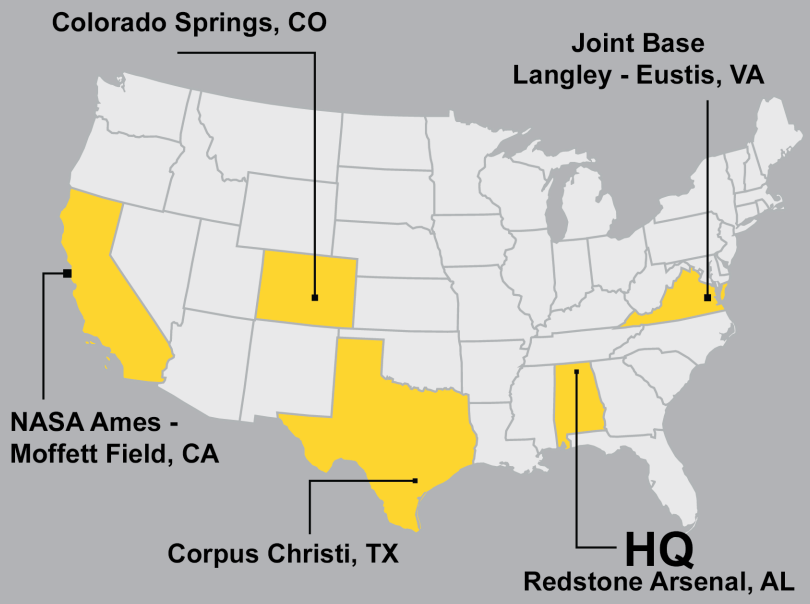
$3.4B

7% Aviation S&T

8% Missile S&T

58% Army

27% Other



Core Competencies

- Life Cycle Engineering
- Research, Technology Development and Demonstration
- Design and Modification
- Software Engineering
- Systems Integration
- Test and Evaluation
- Qualification
- Aerodynamics/ Aeromechanics
- Structures
- Propulsion
- Guidance/Navigation
- Autonomy and Teaming
- Radio Frequency (RF) Technology
- Fire Control Radar Technology
- Image Processing
- Models and Simulation
- Cyber Security

#1: Readiness

Provide aviation and missile systems solutions to ensure victory on the battlefield today.

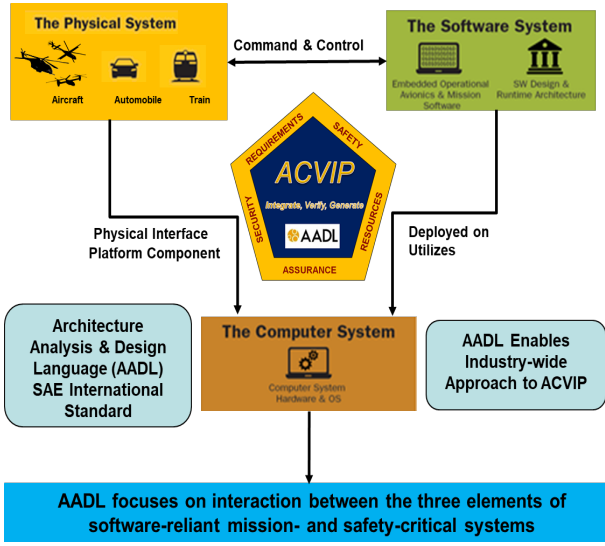
#2: Future Force

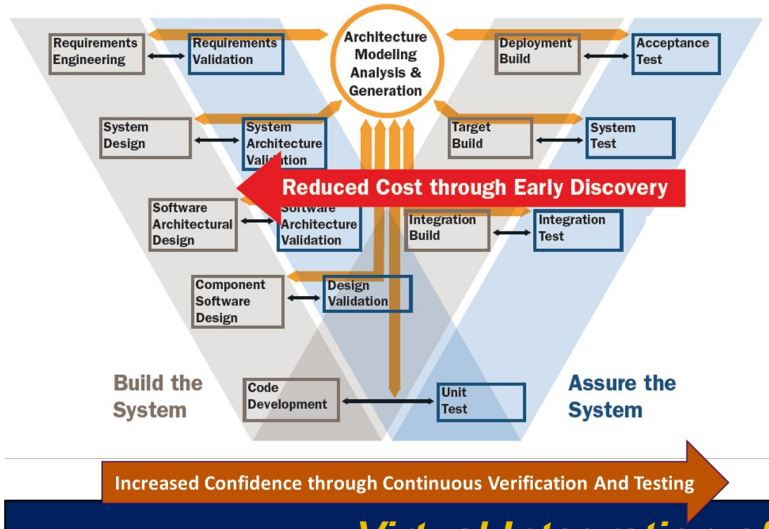
Develop and mature Science and Technology to provide technical capability to our Army's (and nation's) aviation and missile systems.

#3: Soldiers and People

Develop the engineering talent to support both Science and Technology and the aviation and missile materiel enterprise

ARCHITECTURE CENTRIC VIRTUAL INTEGRATION PROCESS (ACVIP)





- **Leverages** research from the AVSI **SAVI** consortium which used virtual integration to draw down costs in commercial aviation systems
- **Utilizes architecture models** to perform **virtual integration focusing on** software-intensive parts of real-time safety- and security-critical **computing systems** to **identify issues early before integration**
- **Process** (from ACVIP Modeling & Analysis Handbook)
  1) **Develop ACVIP Management Plan**
  2) **Establish Model Structure**
  3) **Define Model Content Needed for Analysis**
  4) **Incrementally Execute Analyses, Resolve**
  5) **Build System in Conformance to Models**
  6) **Support Certification and Readiness Reviews**
- **Supports architecture-based compositional modeling and analysis** of computing system properties
- Analytical results support **increasing assurance confidence** and compliments testing
- Provides an **"Authoritative Source of Truth"** embedded systems architectural model

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

AADL (AS-5002) STANDARD SUITE

Core AADL language standard [V1 2004, V2 2012, V2.2 2017]

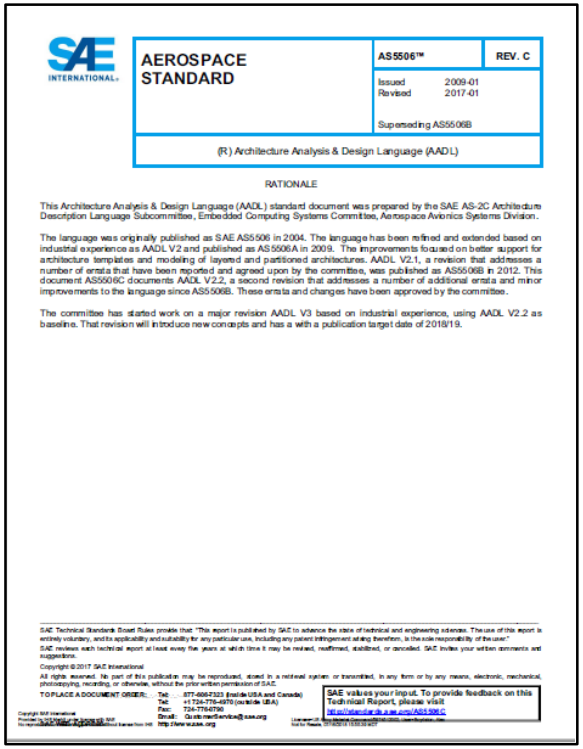
- Focused on embedded & cyberphysical software system modeling, analysis, and generation
- Strongly typed language with well-defined semantics for execution of threads, processes on partitions and processor, sampled/queued communication, modes, end-to-end flows
- Textual and graphical notation
- Revision V3 in progress: interface composition, system configuration, binding, type system unification

Standardized AADL Annex Extensions

- Error Model language for safety, reliability, security analysis [2006, 2015]
- ARINC653 extension for partitioned architectures [2011, 2015]
- Behavior Specification Language for modes and interaction behavior [2011, 2017]
- Data Modeling extension for interfacing with data models (UML, ASN.1, …) [2011]
- AADL Runtime System & Code Generation [2006, 2015]

AADL Annexes in Progress

- Cyber Security Annex
- **FACE Annex**
- Network Specification Annex
- Requirements Definition and Assurance Annex
- Synchronous System Specification Annex



SAE INTERNATIONAL

AEROSPACE STANDARD

AS5506™ REV. C

Issued 2009-01
Revised 2017-01

Superseding AS5506B

(R) Architecture Analysis & Design Language (AADL)

RATIONALE

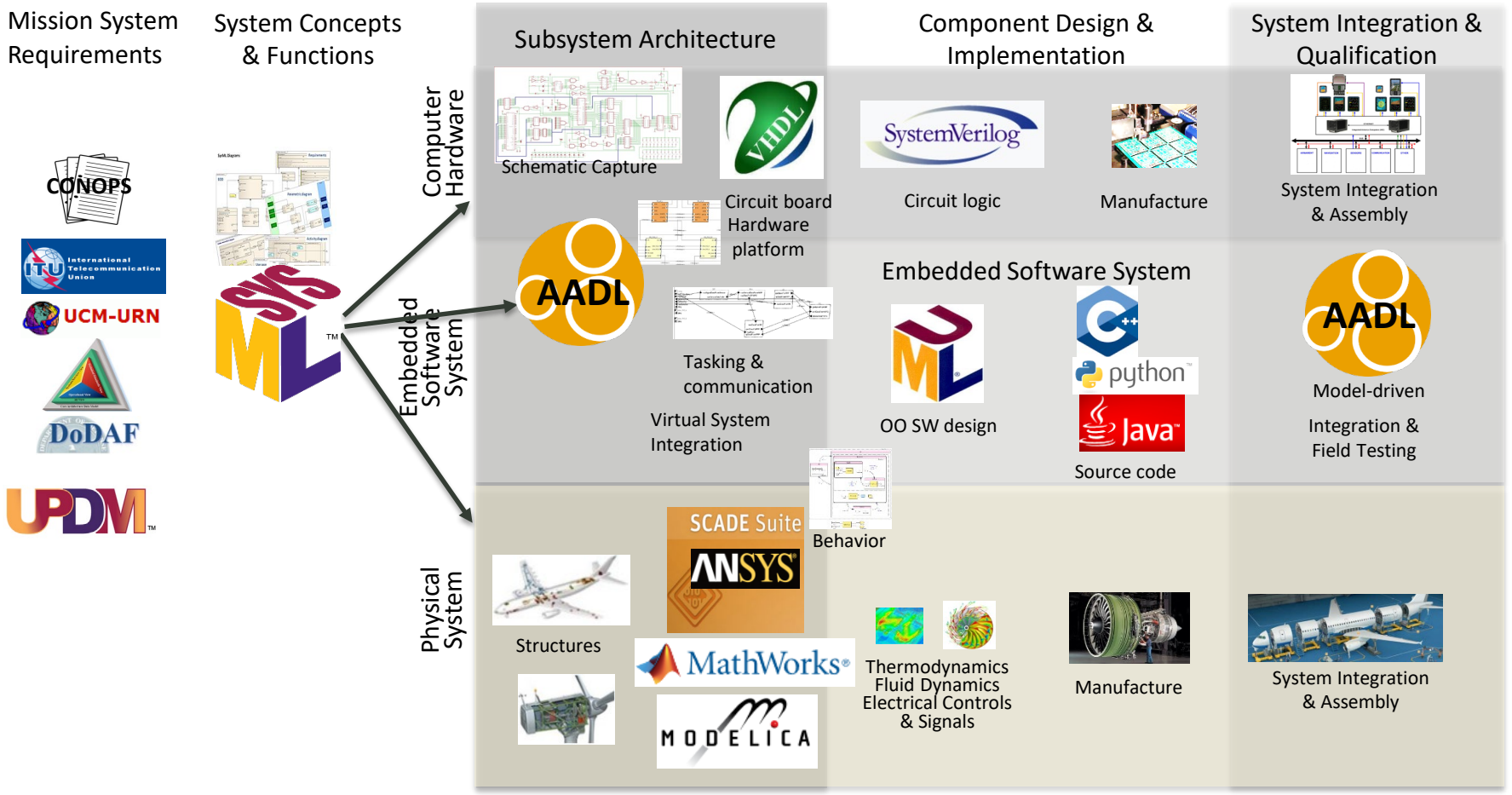
This Architecture Analysis & Design Language (AADL) standard document was prepared by the SAE AS-2C Architecture Description Language Subcommittee, Embedded Computing Systems Committee, Aerospace Avionics Systems Division.

The language was originally published as SAE AS5506 in 2004. The language has been refined and extended based on industrial experience as AADL V2 and published as AS5506A in 2009. The improvements focused on better support for architecture templates and modeling of layered and partitioned architectures. AADL V2.1, a revision that addresses a number of errata that have been reported and agreed upon by the committee, was published as AS5506B in 2012. This document AS5506C documents AADL V2.2, a second revision that addresses a number of additional errata and minor improvements to the language since AS5506B. These errata and changes have been approved by the committee.

The committee has started work on a major revision AADL V3 based on industrial experience, using AADL V2.2 as baseline. That revision will introduce new concepts and has a with a publication target date of 2018/19.

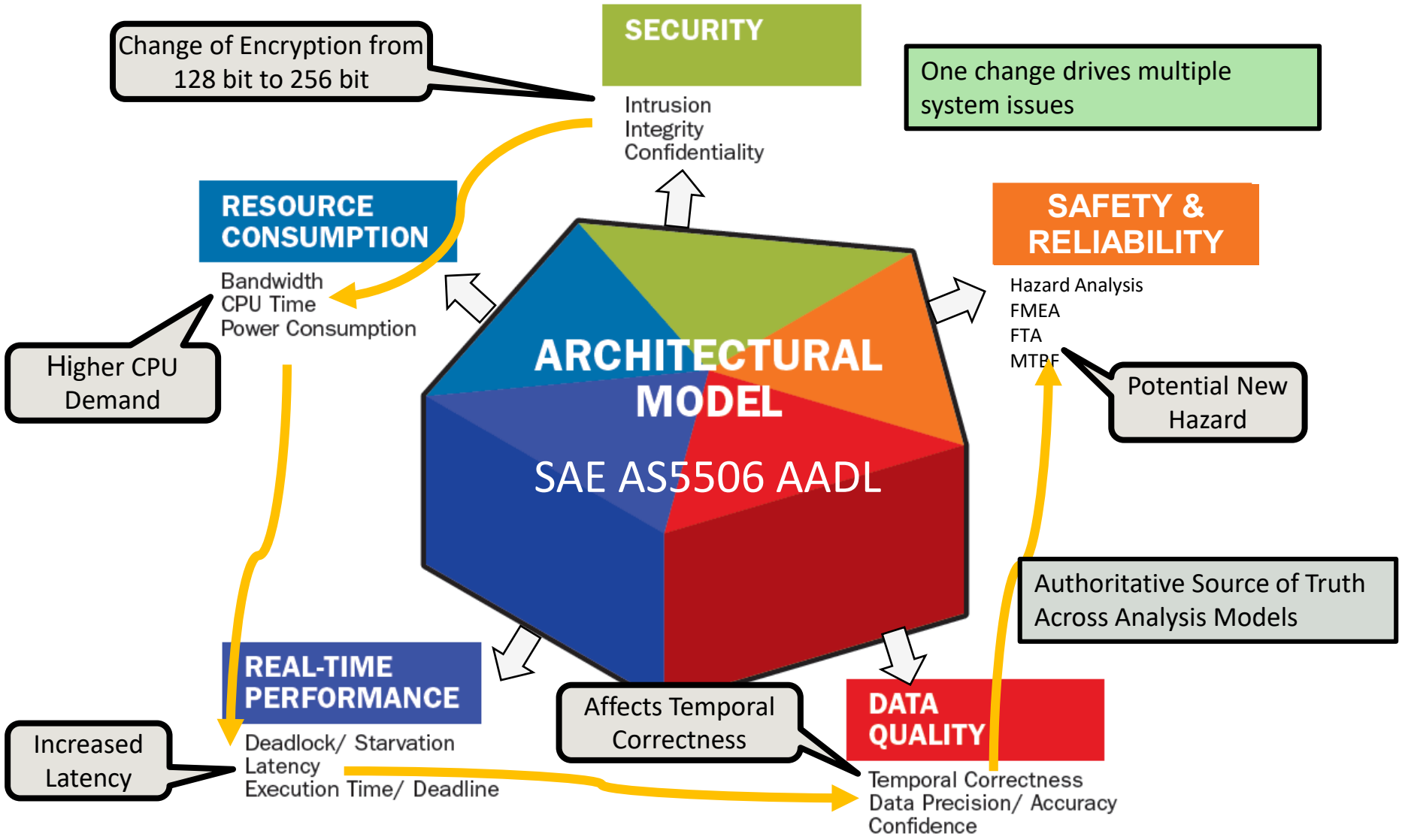
SAE values your input. To provide feedback on this Technical Report, please visit

MULTIPLE LANGUAGES NAD TOOLS TO MEET USERS NEEDS



Filling the Modeling and Analysis Gap for Embedded Software System

ANALYSIS OF SYSTEM PROPERTIES VIA ARCHITECTURE MODEL
A CONTRIBUTION TO AUTHORITATIVE SOURCE OF TRUTH



ACVIP GUIDANCE & TOOLS

ACVIP Handbooks Updated for Capstone Demo & DEWG

- Overview
- OSD DEWG Overview
- Modeling & Analysis
- Acquisition Management

AADL Based Tools Available for Capstone Demo as ACVIPv1.0

- Architecture Led Integrated System Assurance (ALISA)
- Architecture Topology Analysis
- ARINC 653 Analysis & Generation Tools
- Behavior Analysis
- Computer Resource Analysis
- Continuous Virtual Integration Test
- Functional Integration Analysis
- Model Based Testing
- Open Source AADL Tool Environment (OSATE)
- Security Analysis (MILS, RMF)
- Safety Analysis Support (MIL-STD-882, SAE ARP 4761 & STPA)
- Structural, Compositional and Formal Method Analyses
- System of Systems Simulation
- Translators and Translation Guidance (FACE-AADL, SysML-AADL)
- Timing, Latency and Scheduling Analysis

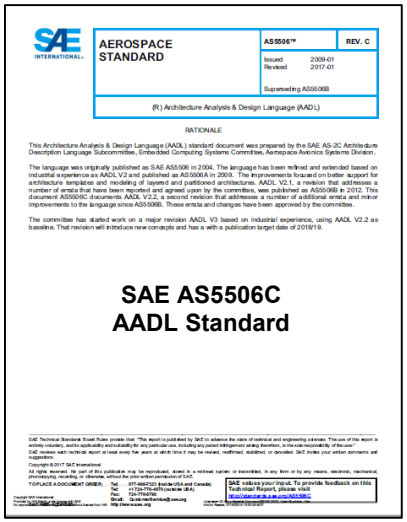
https://osate.org

https://www.adventiumlabs.com/our-work/products-services/model-based-engineering-mbe-tools

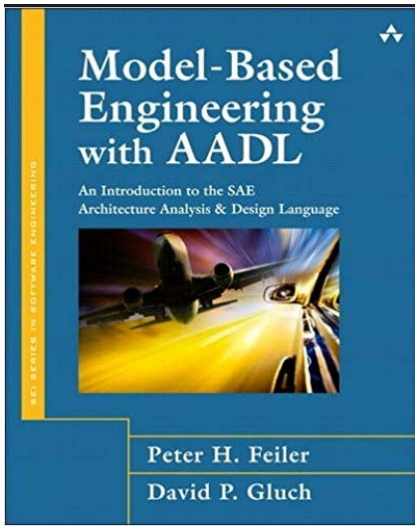
Plus new tools from multiple Sources:
*** SBIRs**
*** DARPA**
*** Europe**
*** etc..**

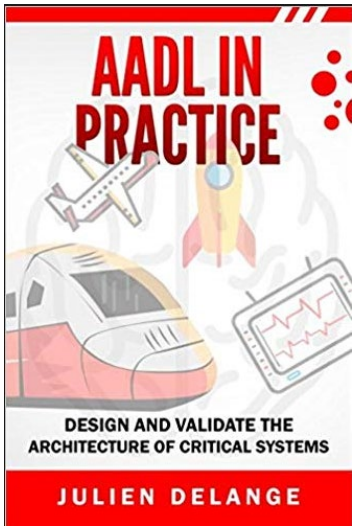
ACVIP guidance and tools are exercised, evaluated and matured on JMR MSAD to support legacy and future aviation systems

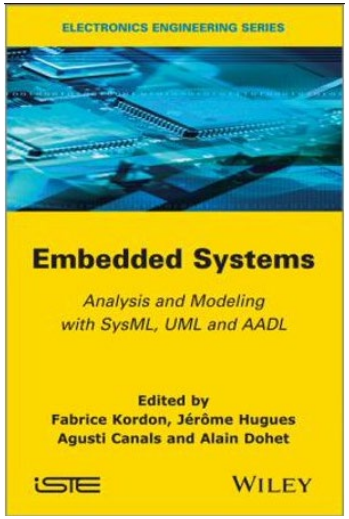
AADL REFERENCES & AVAILABLE TRAINING

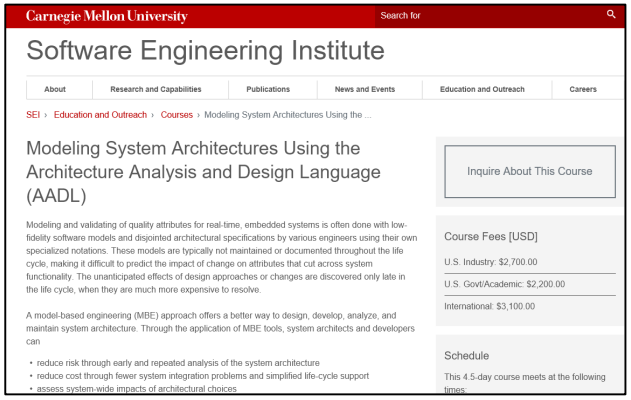


SAE AS5506C
AADL Standard

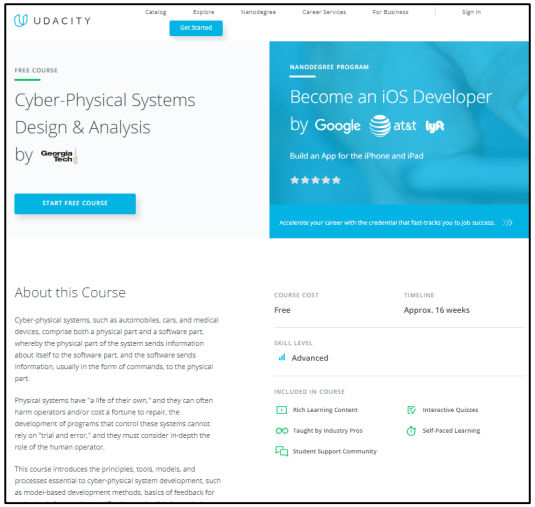




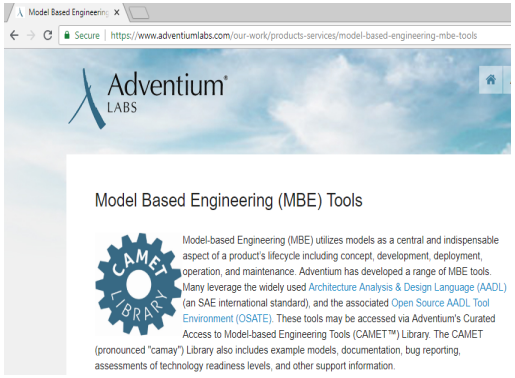




SEI Course on AADL:
https://www.sei.cmu.edu/education-outreach/courses/course.cfm?courseCode=P72



Georgia Tech Class on Cyber Physical Systems
https://www.udacity.com/course/cyber-physical-systems-design-analysis--ud9876

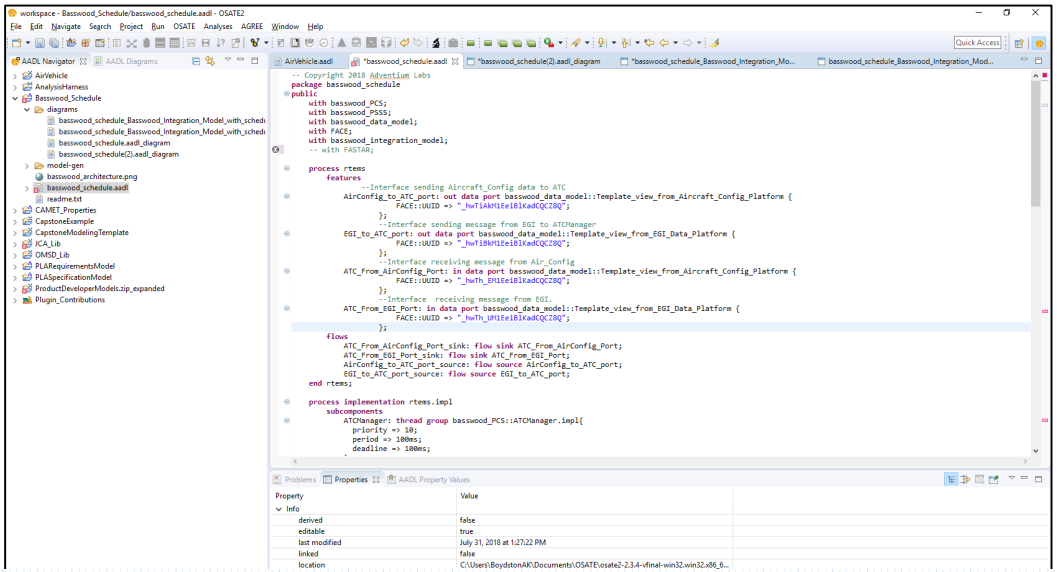


Adventium CAMET SITE
www.camet-library.com

BACKUP CHARTS

OPEN SOURCE AADL TOOL ENVIRONMENT (OSATE)

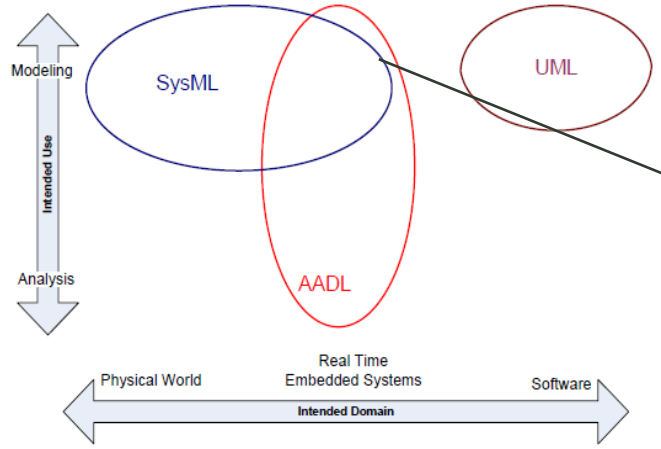
- Intended for open source environment for R&D purposes
- Allows creation of AADL models using a syntax-aware text editor and synchronized graphical editor.
- AADL models are organized in separate projects in a workspace.
- The tool supports validation of AADL models according to all naming and legality rules defined in the AADL standard.
- The text editor provides code templates, real-time syntax checking, code completion, and proposals to fix errors.
- In addition to the core language, models may contain elements from the AADL Standard Annexes (e.g., error model, requirements definition and analysis, ARINC653, and behavior annexes).



AADL Model Text and Graphics auto-generated from FACE 3.0 model by FACE-AADL Translator

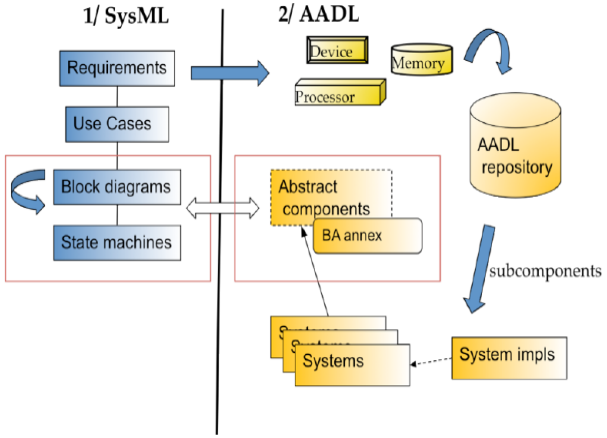
SYSML & AADL COMPARISON

USE



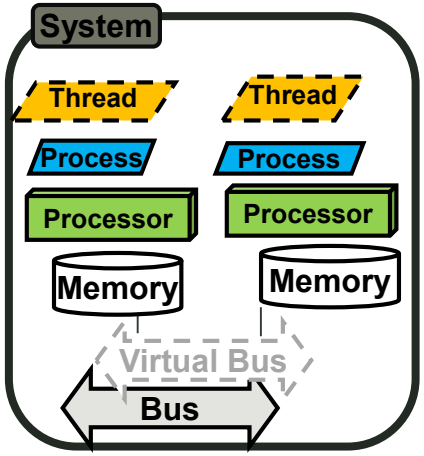
Reference: Rockwell Collins "SysML to AADL Translator User's Manual" v1.1

TRANSLATE



Reference: https://hal.archives-ouvertes.fr/hal-00669391/document page 5

INTEGRATE THEN BUILD



Modeling Language	SysML	AADL
Standards Org	OMG	SAE International AS 5506
Purpose	Larger Systems Modeling & Analysis	Embedded Software Systems Modeling & Analysis
Constructs/ Views	Use-Case, Block Diagrams, Internal Block Diagrams, Rqmts, Sequence, Activity, State Machine, Parametric	RT Components (Abstract, Processor, Memory, Bus, System, Threads…), State Machines (Modes, Behavior, Error) Flows, Bindings, connections
Practice / Methodology	Object Oriented Systems Engineering Methodology (OOSEM)	Architecture Centric Virtual Integration Practice (ACVIP)
Tools (Examples)	Rhapsody, SCADE, Sparx EA, MagicDraw, Papyrus, Modelio, etc.	OSATE, Adventium, ANSYS SCADE, ElliDiss, Dassault, WW Technology Group, SBIR produced tools
Practitioners	Commercial and Government Users	Commercial R&D, Government S&T, commercial tools becoming available