



# 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

DISTRIBUTION A: APPROVED FOR PUBLIC  
RELEASE; DISTRIBUTION UNLIMITED



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



**~9,553**  
FY18 Strength



**2,943**  
Civilian

**23**  
Military

**6,587**  
Contractor

## 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

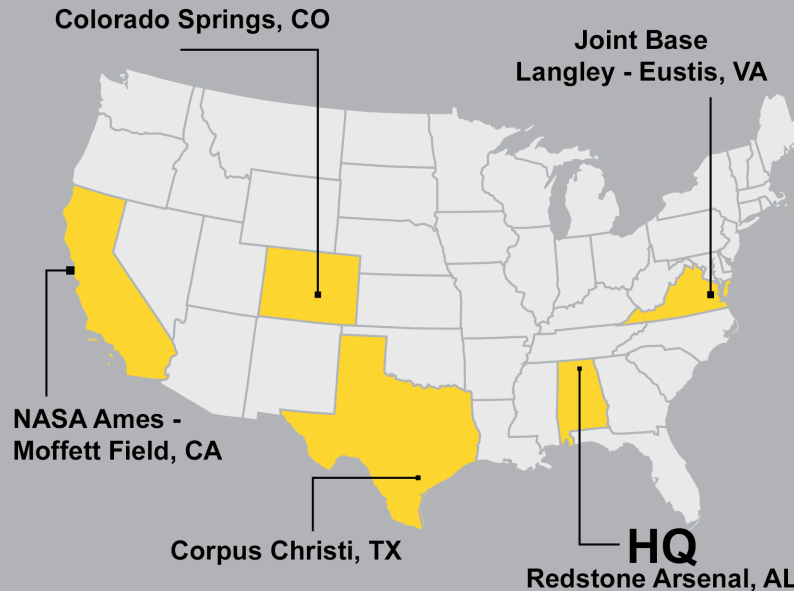
**FY18 Funding**  
**\$3.4B**

**7%**  
Aviation S&T

**8%**  
Missile S&T

**58%**  
Army

**27%**  
Other





## #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.



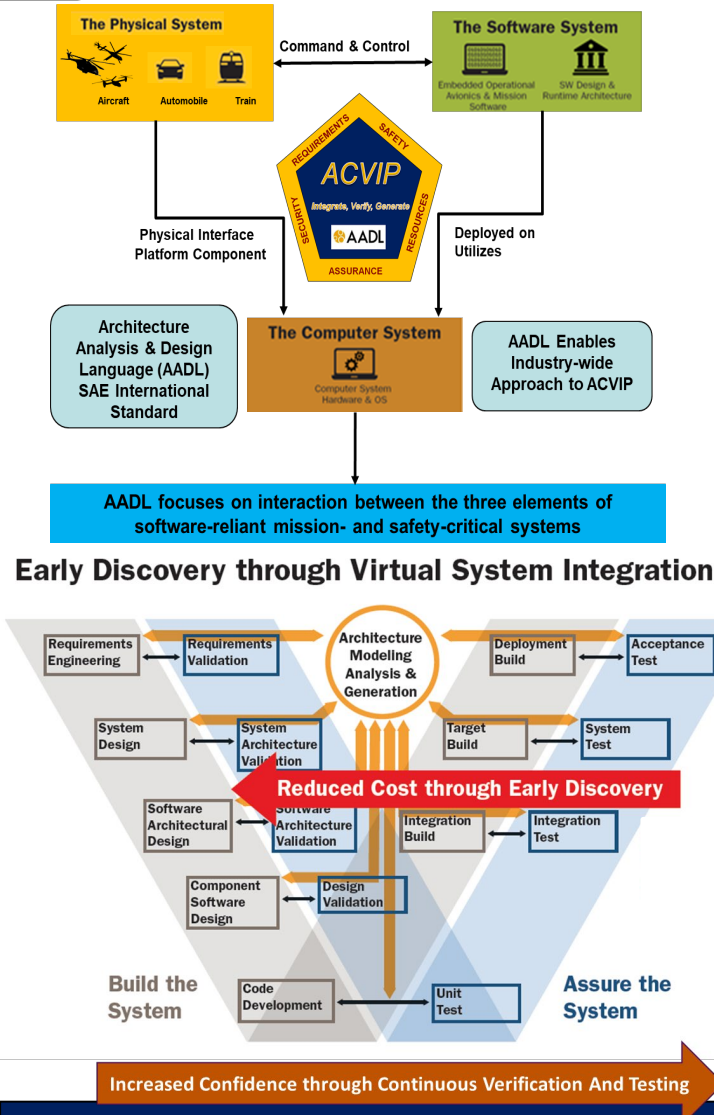
## #3: Soldiers and People

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]

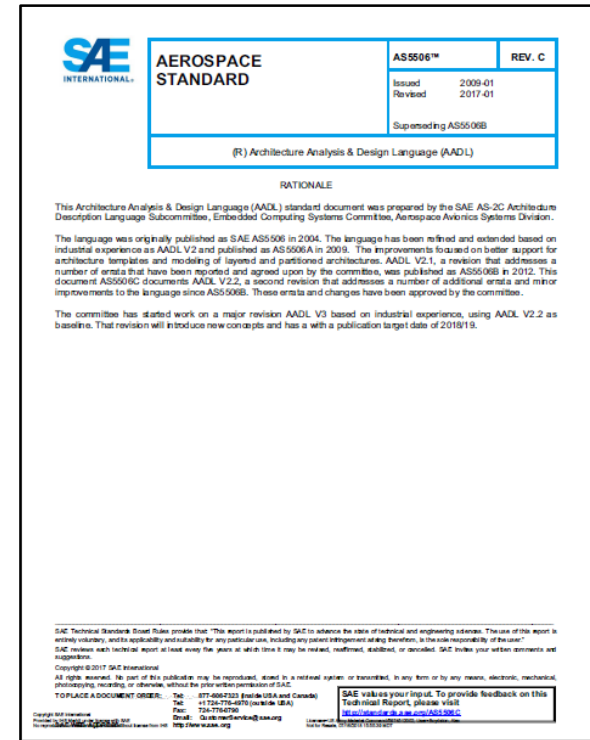
- Focused on embedded & cyberphysical software system modeling, analysis, and generation
- Strongly typed language with well-defined semantics for execution of threads, processes on partitions and processor, sampled/queued communication, modes, end-to-end flows
- Textual and graphical notation
- Revision 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





# MULTIPLE LANGUAGES NAD TOOLS TO MEET USERS NEEDS



Mission System  
Requirements



System Concepts  
& Functions

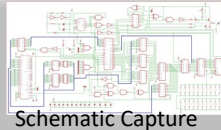


Computer  
Hardware

Embedded  
Software  
System

Physical  
System

Subsystem Architecture



Schematic Capture



Circuit board  
Hardware  
platform



Tasking &  
communication

Virtual System  
Integration

Component Design &  
Implementation



Circuit logic



Manufacture

Embedded Software System



OO SW design



Source code

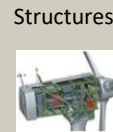
System Integration &  
Qualification



System Integration  
& Assembly



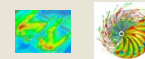
Model-driven  
Integration &  
Field Testing



Structures



Behavior



Thermodynamics  
Fluid Dynamics  
Electrical Controls  
& Signals



Manufacture

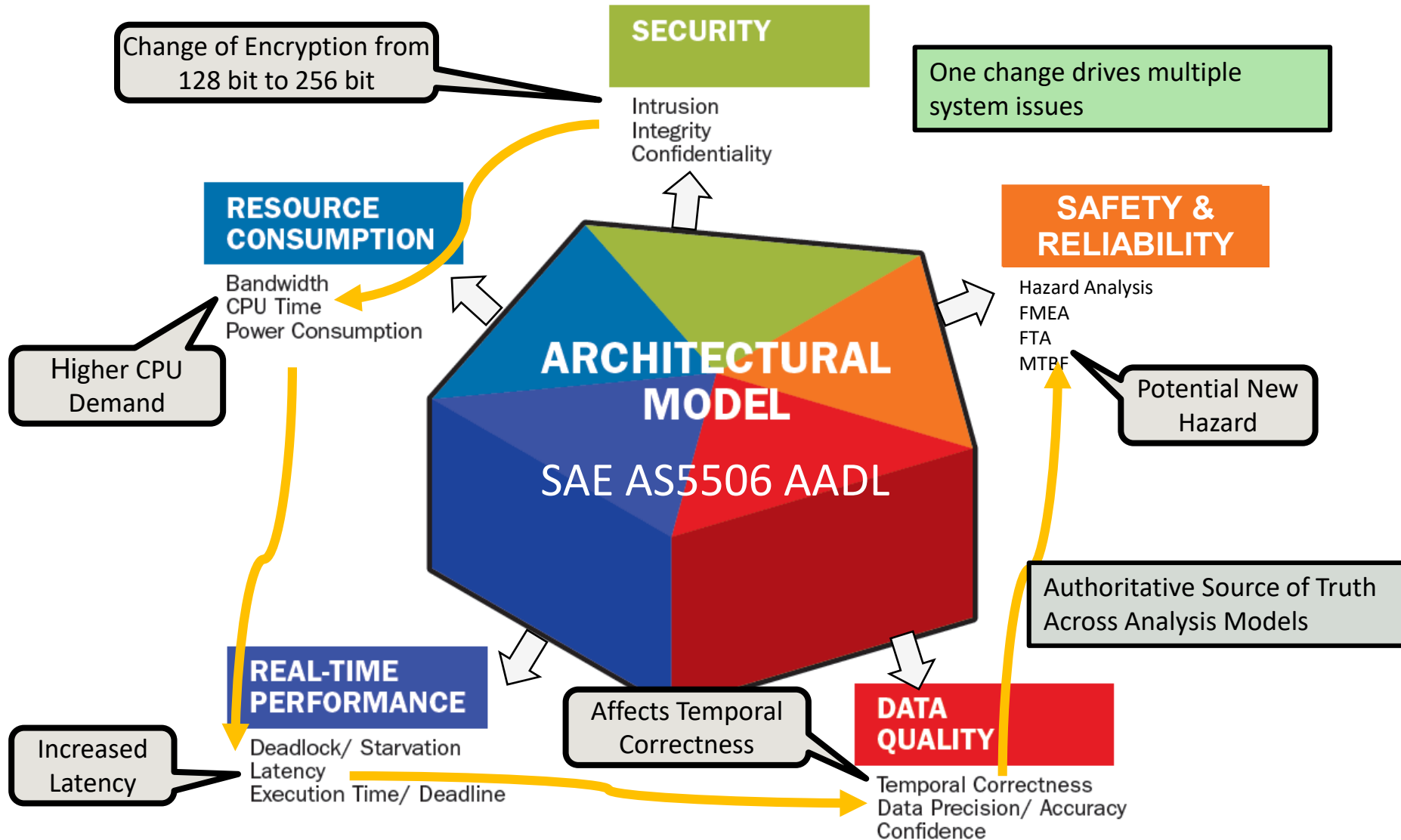


System Integration  
& Assembly

## 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 Handbooks Updated for Capstone Demo & DEWG



Overview



OSD DEWG  
Overview



Modeling &  
Analysis



Acquisition  
Management

# ACVIP GUIDANCE & TOOLS



## 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>



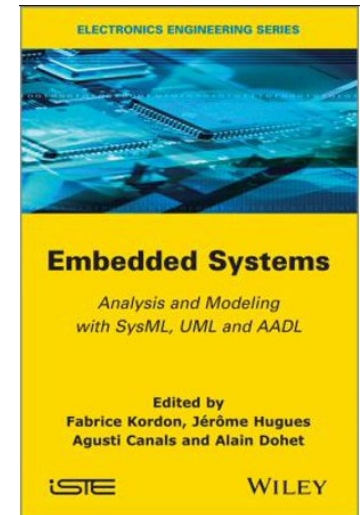
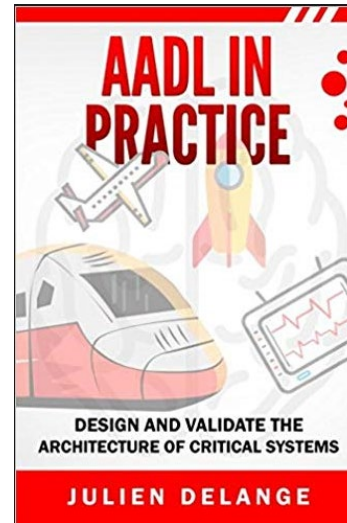
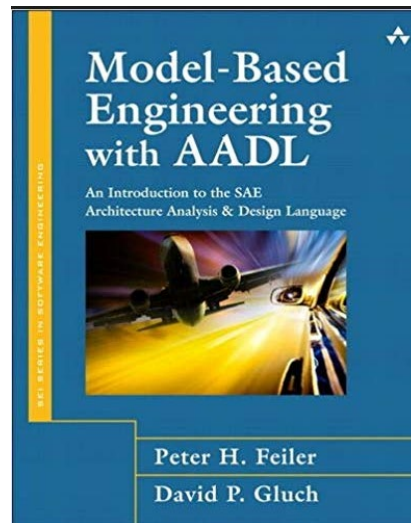
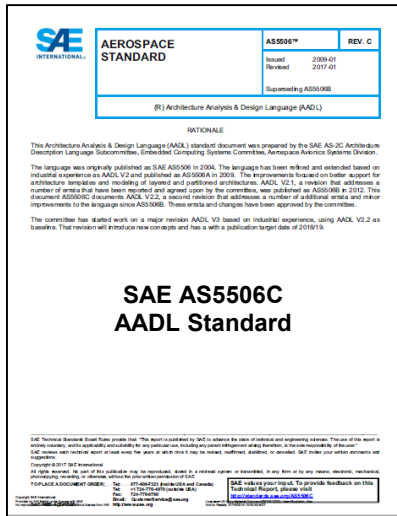
<https://www.adventiumlabs.com/our-work/products-services/model-based-engineering-mbe-tools>

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



**Carnegie Mellon University** Search for

## Software Engineering Institute

About Research and Capabilities Publications News and Events Education and Outreach Careers

SEI > Education and Outreach > Courses > Modeling System Architectures Using the ...

### Modeling System Architectures Using the Architecture Analysis and Design Language (AADL)

Modeling and validating of quality attributes for real-time, embedded systems is often done with low-fidelity software models and disjointed architectural specifications by various engineers using their own specialized notations. These models are typically not maintained or documented throughout the life cycle, making it difficult to predict the impact of change on attributes that cut across system functionality. The unanticipated effects of design approaches or changes are discovered only late in the life cycle, when they are much more expensive to resolve.

A model-based engineering (MBE) approach offers a better way to design, develop, analyze, and maintain system architecture. Through the application of MBE tools, system architects and developers can

- reduce risk through early and repeated analysis of the system architecture
- reduce cost through fewer system integration problems and simplified life-cycle support
- assess system-wide impacts of architectural choices

Inquire About This Course

Course Fees [USD]
U.S. Industry: \$2,700.00
U.S. Govt/Academic: \$2,200.00
International: \$3,100.00

**Schedule**  
This 4.5-day course meets at the following times:

## SEI Course on AADL:

<https://www.sei.cmu.edu/education-outreach/courses/course.cfm?courseCode=P72>

**UDACITY** Create Explore Nanodegree Career Services For Business Sign in

**FREE COURSE**

### Cyber-Physical Systems Design & Analysis

by Georgia Tech

**START FREE COURSE**

**INNOVATION PROGRAM**  
Become an iOS Developer by Google at&tat

Build an App for the iPhone and iPad

★★★★★

Accelerate your career with the credential that top-track you to job success >>>

COURSE COST	TIMELINE
Free	Approx. 16 weeks

**SKILL LEVEL**  
Advanced

**INCLUDED IN COURSE**

- Rich Learning Content
- Interactive Quizzes
- Taught by Industry Pros
- Self-Paced Learning
- Student Support Community

**About this Course**

Cyber-physical systems, such as automobiles, cars, and medical devices, comprise both a physical part and a software part, whereby the physical part of the system sends information about itself to the software part, and the software sends information, usually in the form of commands, to the physical part.

Physical systems have "a life of their own," and they can often harm operators and/or cost a fortune to repair; the development of programs that control these systems cannot rely on "trial and error," and they must consider in-depth the role of the human operator.

This course introduces the principles, tools, models, and processes essential to cyber-physical system development, such as model-based development methods, basics of feedback for

## Georgia Tech Class on Cyber Physical Systems

<https://www.udacity.com/course/cyber-physical-systems-design-analysis--ud9876>

Model Based Engineering (MBE) Tools

**Adventum LABS**

Model-based Engineering (MBE) utilizes models as a central and indispensable aspect of a product's lifecycle including concept, development, deployment, operation, and maintenance. Adventum has developed a range of MBE tools. Many leverage the widely used *Architecture Analysis & Design Language (AADL)* (an SAE international standard), and the associated *Open Source AADL Tool Environment (OSATE)*. These tools may be accessed via Adventum's Curated Access to Model-based Engineering Tools (CAMET™) Library. The CAMET (pronounced "camay") Library also includes example models, documentation, bug reporting, assessments of technology readiness levels, and other support information.

**Adventum CAMET SITE**  
[www.camet-library.com](http://www.camet-library.com)



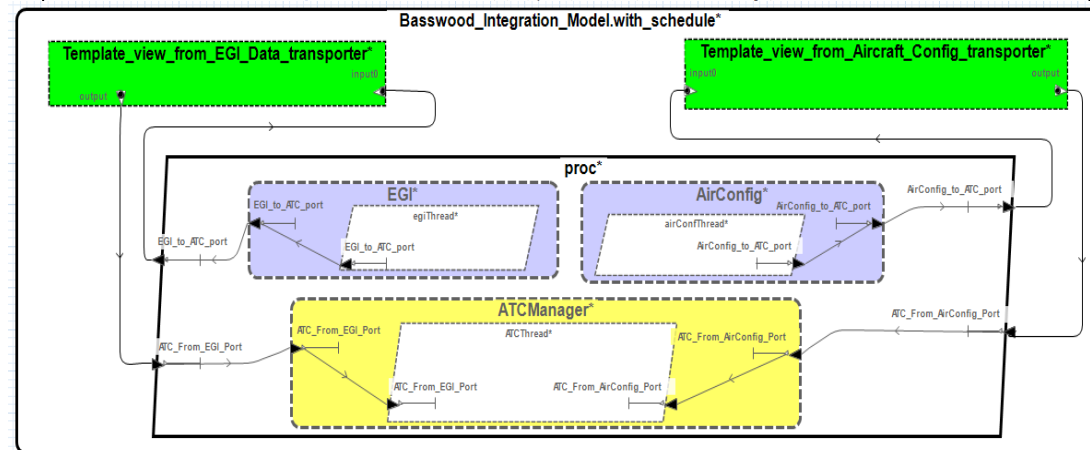
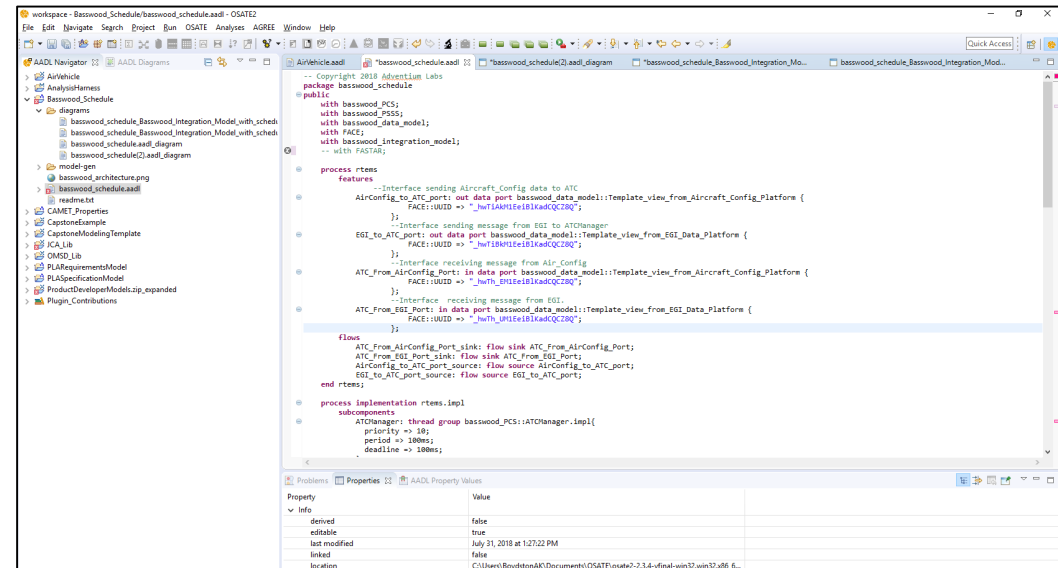
## BACKUP CHARTS



# 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).



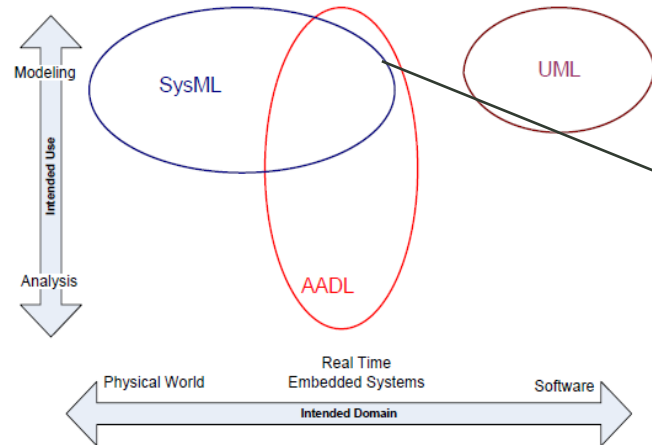
**AADL Model Text and Graphics auto-generated from  
FACE 3.0 model by FACE-AADL Translator**



# SYSML & AADL COMPARISON

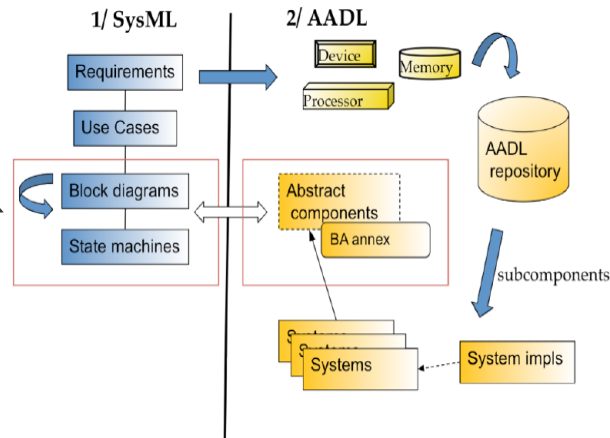


## USE



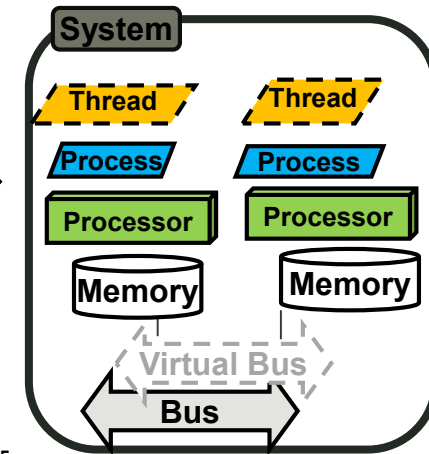
Reference: Rockwell Collins "SysML to AADL Translator User's Manual" v1.1

## 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