Architecture-Centric Virtual Integration Practice with AADL

Peter Feiler

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Copyright 2018 Carnegie Mellon University. All Rights Reserved.

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.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.


DM18-1319
BLUF

Architecture-Centric Virtual Integration Practice (ACVIP) addresses Cyber Physical Systems Issues: A Digital Engineering Contribution

Growth in complexity and late error discovery is driving affordability in embedded software systems.

Trial use has demonstrated that ACVIP and the SAE International Architecture Analysis and Description Language (AADL) standard suite dramatically reduce late error discovery, lowering cost and accelerating the deliver of trusted capability.

ACVIP can be a key contributor to the DoD Digital Engineering Strategy:

• Standard as a foundation for a commercial tool marketplace
• Maturation and commercialization of technology
• Contribution to authoritative source of truth
• Leverage infrastructure and environments
• Transformation of workforce and culture
DoD Digital Engineering Strategy: ACVIP as Key for Cyber Physical Systems
Architecture-Centric Virtual Integration Practice

ACVIP and Emerging Engineering Practice

Accelerating the Adoption of ACVIP
Problem: Growth in Complexity and Late Error Discovery in Cyber Physical Systems is Driving Affordability

Cyber Physical Systems, especially Aviation Systems, are reaching a software affordability limit, impacting the amount of new functionality we can integrate.

Software as % of total system development cost
1997: 45%  ➔ 2010: 66%  ➔ 2024: 88%

Post unit test software rework currently
~50% of total system development cost

This represents a significant opportunity for cost reduction and functional enhancement by discovering issues early through virtual integration and analysis of embedded software system models and synthesis of implementation from verified models.

F35 generated software had integration problems
Technical Challenges in Safety-Critical Embedded Software Systems

Why do system level failures still occur despite best safety practices?

Embedded software systems have become a major safety and cyber security risk
Assurance & Qualification Improvement Strategy

Assurance: **Sufficient evidence that a system implementation meets system requirements**

Architecture-Centric Virtual System Integration Practice (ACVIP)
Architecture Led Incremental System Assurance (ALISA)

Mission Requirements
Function
Behavior
Performance
Survivability
Requirements
Reliability
Safety
Security
Mixture

Model Repository
Architecture Model
Component Models
System Implementation
System configuration

Operational & failure modes
Resource, Timing & Performance Analysis
Reliability, Safety, Security Analysis

2010 SEI Study for AMRDEC
Aviation Engineering Directorate

Data-Driven
High Leverage
Cost Effective
Benefits of Virtual System Integration & Continuous Lifecycle Assurance

Reduced Cost through Early Discovery

Increased Confidence through Continuous Verification And Testing
AADL captures mission and safety critical embedded software system architectures in virtually integrated analyzable models to discover system level problems early and construct implementations from verified models.
Core AADL language standard [V1 2004, V2 2012, V2.2 2017]

- Focused on embedded 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 Annexes in Progress
- Network Specification Annex
- Cyber Security Annex
- FACE Annex
- Requirements Definition and Assurance Annex
- Synchronous System Specification Annex
Analysis of System Properties via Architecture Model
A Contribution to Single Source of Truth

Change of Encryption from 128 bit to 256 bit

One change drives multiple system issues

Higher CPU Demand

Resource Consumption

Bandwidth
CPU Time
Power Consumption

Security

Intrusion
Integrity
Confidentiality

Safety & Reliability

Hazard Analysis
FMEA
FTA
MTBF

Potential New Hazard

Single Source of Truth
Across Analysis Models

Real-Time Performance

Affects Temporal Correctness

Increased Latency

Data Quality

Temporal Correctness
Data Precision/ Accuracy
Confidence

Architectural Model

SAE AS5506 AADL

Architecture-Centric Virtual Integration Practice
with AADL Nov 2018
© 2018 Carnegie Mellon University
Demonstrations of Effectiveness in use of ACVIP with AADL

Finding Problems Early (AMRDEC/SEI)

• Summary: 6 Week Virtual Integration on CH47 using AADL
• Result: Identified 20 major integration issues early
• Benefit: Avoided 12-month delay on 24-month program

Transforming procurement (Joint Multi-Role)

• Summary: Industry/DoD mission system architecture demonstrations using ACVIP
• Result: Pre-integration fault identification
• Benefit: 10X reduction integration test cost

Improving System Security (DARPA / AFRL)

AADL applied to Unmanned Aerial Vehicles & Autonomous Truck

Result: AADL models enforced security policies and were used to auto build the system

Benefit: Combined with formal methods verification, prevented security intrusion by a red team
Synthesize & Verify High-Assurance Systems

High Assurance Cyber Military Systems

Researchers Challenges
- Synthesis of attack-resilient control systems
- Synthesis of operating systems code
- Specification languages: function, environment, hardware, resources
- Composition/Proof engineering
- Scaling
- Attack/fault response
- V&V of complete system
Multiple Languages and Tools to Meet Users Needs

Filling the Modeling and Analysis Gap for Embedded Software System
## SysML & AADL Comparison

### Modeling Language

<table>
<thead>
<tr>
<th>Modeling Language</th>
<th>UML</th>
<th>SysML</th>
<th>AADL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standards Org</td>
<td>OMG</td>
<td>OMG</td>
<td>SAE International AS 5506</td>
</tr>
<tr>
<td>Purpose</td>
<td>Object Oriented Program Modeling</td>
<td>Larger Systems Modeling &amp; Analysis</td>
<td>Embedded Software Systems Modeling &amp; Analysis</td>
</tr>
<tr>
<td>Constructs/ Views</td>
<td>Class Diagram, Block Diagram, Sequence, Activity, State Machine</td>
<td>Use-Case, Block Diagrams, Internal Block Diagrams, Reqmts, Sequence, Activity, State Machine, Parametric</td>
<td>RT Components (Abstract, Processor, Memory, Bus, System, Threads...) State Machines (Modes, Behavior, Error) Flows, Bindings, connections</td>
</tr>
<tr>
<td>Practice / Methodology</td>
<td>Object Oriented</td>
<td>OOSEM</td>
<td>Virtual System Integration, ACVIP</td>
</tr>
<tr>
<td>Tools (Examples)</td>
<td>Rhapsody, SCADE, Sparx EA, MagicDraw, etc.</td>
<td>Rhapsody, SCADE, Sparx EA, MagicDraw, etc.</td>
<td>OSATE, Adventium, ANSYS SCADE, ElliDiss, Dassault, WW Technology Group</td>
</tr>
<tr>
<td>Practitioners</td>
<td>Commercialized</td>
<td>Commercialized</td>
<td>R&amp;D platform, S&amp;T, commercial tools available</td>
</tr>
</tbody>
</table>

### Diagram

- **USE**: SysML, UML
- **STD**: AADL
- **Common Area**: AADL, UML

---

**Translation and Integration Process**

1. **Use**: SysML
2. **Translate**: AADL
3. **Integrate Then Build**: AADL

**Common Area**

- **Use**: Analysis, Modeling, Intended Use
- **Translate**: Incremental Refinement, Analysis & Integration (SW/HW/SYS)

---

**Tools**

- **SysML**:
  - Rhapsody, SCADE, Sparx EA, MagicDraw, etc.
- **AADL**:
  - OSATE, Adventium, ANSYS SCADE, ElliDiss, Dassault, WW Technology Group

---

**Practitioners**

- **Commercialized**

---

**Notes**

- Distribution Statement A: This material has been approved for public release and unlimited distribution

---

**Software Engineering Institute**

Carnegie Mellon University

November 2018
Architecture-Centric Virtual Integration Practice

ACVIP and Emerging Engineering Practice

Accelerating the Adoption of ACVIP
Emerging Engineering Practice and ACVIP

Emerging engineering practices emphasize:

- Rapid delivery of capability using iterative, incremental development paradigms, e.g. MBE, Agile-DevOps and Software Factory.
- Multi-discipline MBE and modular open systems concepts, e.g. Digital Engineering Strategy.
- Synthesis of complex systems from unambiguous descriptions that support analysis and proof/evidence of correctness, e.g. HACMS

ACVIP delivers many of these capabilities for embedded systems.

- Rapid delivery with continuous virtual integration
- Single source of truth HW/SW architecture model repository
- Proof of correctness through semantically precise architecture descriptions
- Automated (Trusted) build capability for embedded systems architecture

Yields => Rapid integration of components into verified open architecture with automated prototyping and testing to final trusted embedded system build
DoD Digital Engineering Strategy and ACVIP Alignment

Goals
1. Formalize the development, integration, and use of models to inform enterprise and program decisions
2. Provide and enduring, and authoritative source of truth
3. Incorporate technological innovation to improve the engineering practice
4. Establish a support infrastructure and environments to perform activities, collaborate, and communicate across stakeholders
5. Transform the culture and workforce to adopt and support digital engineering across the lifecycle

✓ “…model based engineering… to facilitate systems engineering and decision making across the lifecycle”
✓ “…authoritative source of truth … to access, manage, protect, and analyze…data and models”
✓ “…DoD should develop, mature, and use digital engineering methodologies …“
  ✓ “methods and processes to support digital engineering”
  ✓ “…tools…data and interface standards…”
  ✓ “…visualization, analysis, model management and interoperability, workflow, collaboration, and extensions/customization…”

Existing and planned ACVIP capabilities address many Digital Engineering Strategy elements for embedded systems.
DoD Digital Engineering Strategy: Cyber Physical Systems as Key Component
ACVIP and DevOps must be/are Part of the DoD Digital Engineering Maturation and Transition

ACVIP and DevOps are key to addressing Cyber Physical System Issues

ACVIP and DevOps must leverage efforts in changing culture and workforce

ACVIP and DevOps contribute to benefits specifically for cyber physical systems
Architecture-Centric Virtual Integration Practice

ACVIP and Emerging Engineering Practice

Accelerating the Adoption of ACVIP
ACVIP Maturation and Adoption Strategy

1. Research & Development
Mature and Extend ACVIP Capabilities including:
- Continuous Model Integration and Exchange for SSOT
- Model consistency verification framework
- Behavioral integration analysis
- Parallelized analysis to support scalability
- Cybersecurity analysis, etc.

2. Adapt/Apply
Adapt and apply ACVIP on selected legacy aviation systems

3. Community of Practice
Build an ACVIP Community of Practice with participants from Academia, Gov’t, Industry, International Research agreements.

4. ACVIP Lab
Establish ACVIP lab and center of excellence to support adoption within the AMRDEC.

5. Workforce Development
Develop and support ACVIP/AADL training

6. AADL Standards
Extend SAE AADL standard and tool support

7. Assess Cost/Benefit
Assess ACVIP/AADL Cost/Benefit
Additional High Leverage Research Opportunities

Schedulability and Latency
- End to end scheduling and latency
- Multi-core scheduling and latency
- AADL Guidance aligned with Multicore Association open standards

Safety and Security
- Safety engineering of embedded software systems
- Integrated safety and security engineering approach
- AADL Security annex, revision of Error Model (safety) annex

Design Space Exploration for Embedded Software Systems
- Impact of architecture design choices
  - sampled and message-driven task and communication architectures
  - hardware, partitioning, encryption
  - logical and physical redundancy
- AADL V3 addresses configurability of multiple dimensions

Single Source of Truth

See Synthesis and Verification of High Assurance Systems slide for other research challenges
Other S&T Research involving AADL Maturation

SAE AADL Standards Work
- Committee chair and standard suite architect
- Authoring of AADL V3 and various Annex extensions
- Prototype reference implementation

JMR Mission Systems Architecture Demonstration
- JCA Demonstration
- Architecture Implementation Process Demonstrations (AIPD)
- Capstone Demonstration

DARPA 6.2 S&T Programs using AADL
- HACMS (High Assurance Cyber Military Systems)
- CASE (Cyber Assured Systems Engineering)

Other Army Aviation 6.3-6.3 S&T Programs for potential use
- Integrated Mission Equipment (IME)
- Synergistic Unmanned Manned Intelligent Teaming (SUMIT)
- Degraded Vision Environment Mitigation (DVE-M)
- Legacy aircraft systems and subsystems

SBIR funded projects
- Incremental Partitioning to Minimize Change Impact
- Mixed Critical Cyber Physical Systems & Advanced Integrity and Safety Assurance (WWTG)
- Model Based Testing of Integrated Aviation Mission Systems (IDT)
- Rapid Configuration of Heterogeneous Collaborative System-of-Systems Simulations (Physical Optics Corp)
- State Linked Interface Compliance Engine for Data (SLICED) (Adventus)
- Unified Behavior Descriptions for AADL Architecture Models (Adventus, POC)
- Security & Safety Co-Analysis Tool Environment (SSCATE) (Adventus)
- Virtual MBSE Platform to Enable Agile Development of Secure FACE Software (DornerWorks)
Summary

Cyber Physical Systems are facing exponential growth in software development cost exceeding 70% of total system development cost

ACVIP is a set of technologies and practices that specifically have been designed to provide early detection and continuous verification throughout the life cycle

ACVIP is a key contributor to the DoD Digital Engineering Strategy
Backup Charts
Systems Design & Challenges with Computing Systems Integration

Systems Design
- Involves multiple engineering disciplines
- Each requires different languages/tools/methods
- Most functionality deployed in software
- Software V&V does not begin until integration

Design Challenges
- Maintaining consistency across design elements (units, data, messages, etc.)
- Inability to detect emergent side effects of limited resources, timing, scheduling
- Predicting interaction of requirements change during development & sustainment
- Qualifying and certifying the system

The growth in system complexity is being partially addressed by various MBSE tools and methods, but there remains challenges in integration and qualification of integration with software.
Cost Reduction Potential through Virtual Integration of Embedded Software Systems

AT Kearney “Software: The Brains Behind U.S. Defenses Systems”

Nominal development cost reduction of 26.1% ($2.391B out of $9.186B) for a 27 MSLOC system

Source: ROI Analysis of the System Architecture Virtual Integration Initiative
CMU/SEI-2018-TR-002
Previous/On-going Investment into AADL & Virtual System Integration

SAE AADL standard suite (1999-now)
- Committee work B. Lewis: chair, P. Feiler: Tech lead [AMRDEC]
- Standard document content authoring [AMRDEC, other*]
- AADL and annex concept development [SEI, AMRDEC, other*]

AADL Tooling (2004-now)
- Open Source AADL Tool Set (OSATE) development [SEI, AMRDEC**]
- AADL FACE translation tools [AMRDEC]
- Graphical OSATE support by UAH [AMRDEC]
- SBIR projects by Adventium Labs and WW Technology Group [AMRDEC]

Education and Training (2004-now)
- Model-based Engineering with AADL Book [SEI]
- Virtual System Integration methods [SEI, AMRDEC]
- 5 day course on AADL, 2 day workshop [SEI]
- Joint Multi Role (JMR) ACVIP Engineering and Acquisition handbooks [AMRDEC]

Research into Architecture Analysis and Verification (2005-now)
- SEI, DARPA, European Union funded projects in US and Europe
- Current SEI research
  - Integrated safety and security engineering (3 year) [SEI]
  - New concepts for AADL V3 and feasibility prototyping [SEI***]

* European funding sources for European committee members
** Development SEI funded. Maturation and maintenance by AMRDEC JMR
*** Peter Feiler SEI Fellow project funds
Model Based DevOps for Embedded Systems Using AADL/ACVIP

AADL Architecture Analysis Verification

AADL Partial Component Specification

Component Source
- Developed manually
- Generated: SCADE, Simulink, etc.
- Reused component

Correct; verified components

Trusted Compile and Build

Platform Specific Binaries

Airborne Tactical Future Cloud Terrestrial
Note About AADL and DevOps

AADL has been designed to encompass embedded system architectural design and analysis, and later generation of complex systems.

System generation relies on code generation techniques, coupling the AADL data and code generation annexes to core language.

AADL supports the many of the central pillars of DevOps philosophy:

• coding (both through modeling and code generation)
• building the generated code
• testing it at model or code level
• packaging/releasing through regular mechanisms and configuring the running infrastructures

AADL does not address the last DevOps stack, “monitoring the deployment”, to update and improve the system overall performance, and would need to be extended to support this capability.