Model Based Engineering to System Architecture Virtual Integration (SAVI) with Demonstration

Bruce Lewis
US Army RDEC, SEI Affiliate

Peter H Feiler
Software Engineering Institute

Dave Redman
Aerospace Vehicle Systems Institute

FAA Software & Airborne Electronic Hardware Conf.
San Jose, CA
August 2009
Definitions

• **Architecture Analysis & Design Language** is a precise Architecture Description Language designed for Model-Based Engineering via virtual integration of the Embedded Software System.

• **Model-Based Engineering of the Embedded Software System Architecture** is the modeling and quantitative analysis of the integrated components, hardware and software, static and dynamic.

• **System Architecture Virtual Integration (SAVI)** is a project and an industrial process and infrastructure enabling virtual integration and analyses of the system and embedded software system from acquisition through development – “integrate then build”.

SAVI is being developed for next gen aircraft and upgrades to current aircraft but applicable to any ESS.
We Rely on Embedded Systems for Safe Vehicle Operation

Even with the autopilot off, flight control computers still command control surfaces to protect the aircraft from unsafe conditions such as a stall," the investigators said.

The unit continued to send false stall and speed warnings to the aircraft's primary computer and about 2 minutes after the initial fault "generated very high, random and incorrect values for the aircraft's angle of attack."

Embedded software systems introduce a new class of problems not addressed by traditional system modeling & analysis.
Mismatched Assumptions Cause Many Issues

Physical Plant Characteristics
- Lag, proximity

Data Stream Characteristics
- End-to-end latency (F16)

System Under Control

Control System

Hardware Engineer

Application Developer

System Engineer

SysML does not address Embedded Software System Architecture Issues

Precision Units
- Ariane 4/5

Compute Platform

Runtime Architecture

Application Software

Concurrency Communication
- ITunes crashes on dual-cores

Distribution & Redundancy
- Virtualization of HW (ARPA-Net split)

Embedded SW System Engineer

Why do system level failures still occur despite fault tolerance techniques being deployed in systems?
Architecture Level Software System Faults & Their Root Causes

Violation of data stream assumptions
• Stream miss rates, Mismatched data representation, Latency jitter & age

Partitions as Isolation Regions
• Space, time, and bandwidth partitioning
• Isolation not guaranteed due to undocumented resource sharing
• Fault containment, security levels, safety levels, distribution

Virtualization of time & resources
• Logical vs. physical redundancy
• Time stamping of data & asynchronous systems

Inconsistent System States & Interactions
• Modal systems with modal components
• Concurrency & redundancy management
• Application level interaction protocols

Performance impedance mismatches
• Processor, memory & network resources
• Compositional & replacement performance mismatches
• Unmanaged computer system resources

End-to-end latency analysis
Port connection consistency

Partitioned architecture models
Model compliance

Virtual processors & buses
Synchronization domains

Fault propagation
Security analysis
Architectural redundancy patterns

Resource budget analysis & task roll-up analysis
Resource allocation & deployment configurations
Cost & Time Reduction due to Early Fault Discovery

System & embedded software system faults

Where faults are introduced
Where faults are found
The estimated nominal cost for fault removal

SAVI applies MBE early From Acquisition

AADL: The Language

Designed for standardized incremental, composable, quantitative analysis and generative system integration

Precise semantics for components & interactions
• Thread, process, data, subprogram, system, processor, memory, bus, device, virtual processor, virtual bus, abstract
• Typed properties, properties with units and model reference values

Continuous control & event response processing
• Data and event flow, synchronous call/return, shared access
• End-to-End flow specifications, black box flow specs

Operational modes & fault tolerant configurations
• Modes & mode transition, mode specific properties & configurations

Modeling of large-scale systems
• Component variants, packaging of AADL models, public/private

Accommodation of diverse analysis needs
• Extension mechanism (property set, sublanguage) standardized
Modeling an Embedded System Architecture

Elements of an embedded system architecture

- **Application SW Architecture** (task & communication)
- **Computer platform architecture** (processors & networks)
- **Physical system/environment** (interface with embedded SW/HW)
- **Logical interface between software and physical system**
- **Physical interface between computer platform and physical system**
- **Deployment of software on computer platform**

SAE AADL supports modeling, analysis, and auto-generation of embedded system architectures.

SAVI provides an industrial and acquisition process as well as the infrastructure for application of MBE.
System Engineering domain with SysML can be linked with AADL through SAVI model bus or tools that integrate both languages.

Tools are already being demonstrated that integrate Simulink, SCADE, SDL design and components to AADL specified Architecture and generation.
Potential Model-based Engineering Pitfalls

The Issues

- Inconsistency between independently developed analytical models

Potential Solution

- Architecture-centric model repository

- Generation from validated models

System models

System implementation

Confidence that model reflects implementation
Architecture-Centric Engineering Approach

**Virtual Integration & Validation of System Architecture**

- **Availability & Reliability**
  - MTBF
  - FMEA
  - Hazard analysis

- **Security**
  - Intrusion
  - Integrity
  - Confidentiality

- **Resource Consumption**
  - Bandwidth
  - CPU time
  - Power consumption

- **Data Quality**
  - Data precision/accuracy
  - Temporal correctness
  - Confidence

- **Real-time Performance**
  - Execution time/Deadline
  - Deadlock/starvation
  - Latency

- **SAE AADL Architecture Model**

- **Auto-generated analytical models**

SAVI Model Repository based on same concept, + Acquisition incremental refinement, Proprietary data, Distribution

© Texas Engineering Experiment Station

FAA SW & AEH Conference | San Jose, CA
Single-Model, Multi-Dimensional Analysis

SAVI Defined Analyses
Enhance Automation

SECURITY
- Intrusion
- Integrity
- Confidentiality

Increased confidentiality requirement
- change of encryption policy

Key exchange frequency changes
- Message size increases
  - increases bandwidth utilization
  - increases power consumption

Increased computational complexity
- increases WCET
- increases CPU utilization
- increases power consumption
- may increase latency

ARCHITECTURAL MODEL

RESOURCE CONSUMPTION
- Bandwidth
- CPU Time
- Power Consumption

REAL-TIME PERFORMANCE
- Deadlock/Starvation
- Latency
- Execution Time/Deadline

Confidence

Increased computational complexity
- increases WCET
- increases CPU utilization
- increases power consumption
- may increase latency
Key Benefits of AADL Standard

Consistent, rich, compiled model semantics for architecture modeling, analysis and generation of embedded systems.

Supporting incremental specification, multi-dimensional & multi-fidelity modeling & analysis based on a single-source architecture model.

Providing a semantically consistent interchange format, AADL XMI, for integration of models across supplier repositories & interfacing to analysis & generation tools.
Virtual Integration Throughout Life Cycle

From Prediction to Validation

- Sensitivity analysis for uncertainty
  - Requirements Engineering
  - System Design
  - Software Architectural Design
  - Component Software Design
  - Detailed AADL Model
- Confidence in implementation
  - High-level AADL Model
  - Specify Model-Code Interfaces
  - Code Development
  - Unit Test
  - Integration Test
  - System Test
  - Acceptance Test

SAVI defines Analyses Fidelity and Timing

→ generation of test cases
← updating models with actual data

© Texas Engineering Experiment Station
FAA SW & AEH Conference | San Jose, CA
Benefits of Virtual Integration

• Reduce the risks
  – Analyze system early and throughout life cycle
  – Understand system wide impact
  – Validate assumptions across system

• Increase the confidence
  – Validate models to complement integration testing
  – Validate model assumptions in operational system
  – Evolve system models in increasing fidelity

• Reduce the cost
  – Fewer system integration problems
  – Fewer validation steps through use of validated generators
The Aerospace Vehicle Systems Institute

**AVSI** is a global cooperative of aerospace companies, government organizations, and academic institutions.

Past AVSI projects have covered the breadth of aerospace systems and current research includes projects in the areas of reliability, certification, and virtual integration.

The System Architecture Virtual Integration program is an AVSI program addressing virtual integration of systems.

SFI was selected as the contractor to help work the Proof of Concept Effort.
Systems Grow Increasingly Complex

Estimated Onboard SLOC Growth

- Slope = 0.17718
- Intercept = -338.5
- Curve implies SLOC doubles about every 4 years

Year | Onboard SLOC
--- | ---
1960 | INS: 0.8K
1970 | B757, B767: 190K
1980 | A300B: 4.6K
2000 | 8M, 27M
2010 | 61M, 134M
2020 | 299M

Software Base Cost COCOMO II

- Boeing: $160 B
- Airbus: $7.8 B
- Unaffordable: $290 M, $81 M, $38 M

The line fit is pegged at 27M SLOC because the projected SLOC sizes for 2010 through 2020 are unaffordable. The COCOMO II estimated costs to develop that much software are in excess of $10B.


Leading to Schedule and Cost Overruns

Errors Discovered Late in the Development Cycle Contribute More Cost Growth
New Development of Safe Aircraft is Reaching the Limits of Affordability

- Complexity will continue to increase and the situation will get worse, not better
- Individual companies cannot afford to solve this issue alone
- The industry cannot afford to solve it multiple times
- We can’t afford not to solve it

A coordinated, industry-wide effort is needed to solve this issue.
The SAVI Approach: Integrate, Then Build

SAVI is

- A changed acquisition paradigm to facilitate systems integration
- A research effort to define the standards and technologies needed to effect virtual integration
- Built on the three-legged stool of
  - Model-based,
  - Proof-Based, and
  - Component-Based Design
- Structured/transformable data interfaces
- A global collaboration

SAVI is not

- A software tool or a design tool
- A continuation of current system development practices
SAVI Aims to Enable Virtual Systems Integration to Discover Issues Earlier in Development

Standardized architecture language with strong semantics, the **Model Bus** and **Model Repository** concepts in SAVI enable...

... early validation of system and embedded software system behavior to reduce integration errors.

© Texas Engineering Experiment Station

FAA SW & AEH Conference | San Jose, CA
Modified Business Model

System Integrator defines a new product using internal repository of virtual “parts”
Specifications for virtual subsystems sent to suppliers
Proposed and developed subsystem models incrementally provided to integrator

System Integrator

- Repository
- Parse & Process
- Modify
- Create
- Virtually Integrate

Suppliers

- Repository
- Parse & Process
- Modify
- Create
- Virtually Integrate

Model Exchange Via Standardized Interchange Formats

© Texas Engineering Experiment Station
Virtual Integration
Return on Investment Estimate

**NPV (Cost avoidance with SAVI discounted @ 10%)**

\[ ROI = \frac{NPV(\text{cost to develop SAVI discounted at 10%})}{\text{Years}} \]

**Conservative Assumptions**
- SLOC less than now experienced by OEMs (yet SLOC growth exp)
- Low assumes 33% requirements error leakage downstream is prevented by SAVI, expected 66%, high 100%
- Other drivers - % new dev, % rework of dev, % rework req errors
- Assumed Cost to develop SAVI 2X current best estimate
- ROI on a single large aircraft development - 2010-2018

**Positive results**

$100 million – low; $400 million – expected; $1 billion – high
## Architecture Design Language Requirements for SAVI
### Supporting Embedded Software System (ESS) Analyses

<table>
<thead>
<tr>
<th>Desired quality</th>
<th>Reason</th>
<th>AADL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESS architecture concepts with precise semantics</td>
<td>Standardized analysis quantitative assessment</td>
<td>ESS abstractions as language primitives</td>
</tr>
<tr>
<td></td>
<td>Use of formal methods</td>
<td>Semantics well-documented for each component &amp; interaction category</td>
</tr>
<tr>
<td>Checkable consistency of architecture formation</td>
<td>Incremental change impact detectable</td>
<td>Compilable strongly typed language with standard legality &amp; consistency rules</td>
</tr>
<tr>
<td></td>
<td>Impact analysis across quality attributes</td>
<td>EMF-based meta model drives XMI standard</td>
</tr>
<tr>
<td>Component-based fidelity multi-dimensional modeling</td>
<td>Consistency &amp; quantitative analysis early &amp; throughout development life cycle</td>
<td>Hierarchical composable SW/HW/physical components with interaction behavior &amp; timing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Explicit support for templates, patterns, incomplete models</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Standard extensions via property sets &amp; annex sublanguages to core</td>
</tr>
<tr>
<td>Model scalability, variability &amp; management</td>
<td>Large scale system modeling &amp; subcontractor management</td>
<td>Spec/instance separation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type/implementation variation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dynamic re-configurability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Packages to manage model space</td>
</tr>
</tbody>
</table>
SAVI Proof of Concept Demonstration
Incremental, Multi-Fidelity, Multi-Dimensional, Multi-Layered Architectural Modeling & Analysis

- Aircraft system: (Tier 1)
  Engine, Landing Gear, Cockpit, ...
  Weight, Electrical, Fuel, Hydraulics, ...

- IMA System: (Tier 2)
  Hardware platform, software partitions
  Power, MIPS, RAM capacity & budgets
  End-to-end flow latency

- Subcontracted software subsystem: (Tier 3)
  Tasks, periods, execution time
  Software allocation, schedulability
  Generated executables

OEM & Subcontractor:
- Subsystem proposal validation
- Functional integration consistency
- ARINC 429 protocol mappings

Additional Opportunities:
- Safety & security analysis
- Fault modeling & impact analysis
- What-if trade studies

- Multi-tier system & software architecture
- Integrator & subcontractor virtual integration

© Texas Engineering Experiment Station
FAA SW & AEH Conference | San Jose, CA
Contacts

Dr. Donald Ward – SAVI
Program Manager
savipm@dishmail.net

Bruce Lewis – AADL chair
bruce.a.lewis@us.army.mil

Dr. David Redman –
Director, AVSI
dredman@tamu.edu

Peter Feiler – AADL
Technical Lead
phf@sei.cmu.edu

SAVI next phase is being planned – it's your opportunity to join – contact Dr. Ward for more information
Model Based Engineering to System Architecture Virtual Integration (SAVI) with Demonstration

FAA Software & Airborne Electronic Hardware Conf.
Bruce Lewis (US Army RDEC), Peter H Feiler (SEI), D. Redman (AVSI)
San Jose, CA August 2009
Modeling an Embedded System Architecture

Elements of an embedded system architecture

- Application SW Architecture (task & communication) PLUS
- Computer platform architecture (processors & networks) PLUS
- Physical system/environment (interface with embedded SW/HW) PLUS
- Logical interface between software and physical system PLUS
- Physical interface between computer platform and physical system PLUS
- Deployment of software on computer platform

SAE AADL supports modeling, analysis, and auto-generation of embedded system architectures.

SAVI provides an industrial and acquisition process as well as the infrastructure for application of MBE
The Systems and Supply Chain Are Hierarchical

We should expect the same of the tools and processes employed in their development.
Model-Based Engineering (MBE) for Computer Based System Architecture

- Ensure embedded, real-time system performance, reliability, and safety *prior* to system integration, test, or upgrade
- Prediction through quantitative analysis & simulation of system operation based on architecture models
- System validation through model verification and implementation compliance checking

UML-based model-driven development emphasizes data modeling and component interface, but lacks analytic power.
Industrial Embedded Systems Initiatives

OMG MARTE 2005-2009

EAST ADL Consortium AutoSAR

TOPCASED
Open Source Embedded Systems Tool Framework
28 partners €20+M 2005-2008

ITEA SPICES
Model-Driven Embedded Systems Engineering
15 partners €16M 2006-2009

AVSI SAVI
Analysis-based System Validation
8+ partners $40+M 2008-2011

EC ASSERT
Proof-based Satellite Architectures
ESA + 30 partners €15M 2004-2007

OpenGroup
Real-Time Forum
EU + US partners

IST ARTIST2
Embedded Systems Center of Excellence
-2011

FAA SW & AEH Conference | San Jose, CA
AADL & Other Standards

• AADL & OMG MARTE
  – Joint AADL UML profile effort
  – AADL subprofile appendix in MARTE Document (in ballot 2009)

• Embedded systems & System engineering
  – Meeting of minds: technical leads of AADL & SysML (Dec 2008)
  – Coordination: AADL, MARTE, SysML (April 2009)
  – Collaboration: AADL, MARTE, SysML, INCOSE cross membership & joint meetings
  – STEP AP-233 Standards
  – DO-178 / DO-254
  – ARP 2754
Proof of Concept Effort

Objectives

• Develop a credible RoI estimate
• Define a roadmap for development of SAVI
• Develop a Proof of Concept Modeling environment:
  – Establish a prototype Model Bus
  – Establish a prototype Model Repository
  – Define a sample model that captures targeted systems properties
  – Perform system analyses across multiple levels of abstraction
Accomplishments

First Feasibility Demonstration Completed
- Investigation of Acquisition Model showed significant changes
- Proof of Concept demonstration suggested SAVI is technically feasible
- Laid out Road Map to evolve new paradigm
- Estimated Return on Investment (RoI) from software requirements alone is very favorable

Acquisition Model Changes Significantly
- “Integrate, then Build”
  - Static view – complete information model from requirements phase onward
  - Formal definition of proof-based models and data structure
  - Dynamic view – explicit linkage of models across tiers
- Demonstrated concept of tool neutrality through the Model Bus
Proof of Concept Demonstration

Distributed PoC Model Development

Rockwell Collins

BAE Systems

SEI

Airbus

Subversion Model Repository at AVSI
Proof of Concept Demonstration (2)

• Three Models (Tiers 1, 2, and 3) Analyzed
  – Tier 1 (Aircraft level)
  – Tier 2 (Aircraft system level)
  – Tier 3 (Sub-system/LRU level)

• Analysis and Demonstration
  – Propagate requirements and constraints
  – Higher level model down to suppliers' lower level models
  – Verification of lower level models satisfies higher level requirements and constraints

• Evaluation Based on Quality Factors
  – Started with 19 (Criticality, Frequency, Difficulty, Cost, Breadth...)
  – Video demonstrations available
Spiral Development Planned

- Three Iterations to Reach TRL 9

- Schedule Roadmap Is Next

June 09
# SAVI Development Roadmap

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVI PROJECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVI Process description v1.0</td>
<td></td>
<td>Model Bus &amp; Model Repository Specs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Bus &amp; Model Repository Specs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADL Selected</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVI DEPLOYMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVI Tools pre-implementation</td>
<td></td>
<td>Encapsulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVI Tools &amp; Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partial Supply Chain integration (SAVI partners)</td>
<td></td>
<td>Full data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partial Supply Chain integration (SAVI partners)</td>
<td></td>
<td>Full data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full supply chain integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools Vendors (partners)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools Vendors Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMMUNICATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional at model level</td>
<td></td>
<td>Functional &amp; types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aircraft signals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMMUNICATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional at model level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aircraft signals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIRCRAFT APPLICATIONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partial Supply Chain integration (SAVI partners)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partial Supply Chain integration (SAVI partners)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full supply chain integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIRCRAFT APPLICATIONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aircraft program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All suppliers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Texas Engineering Experiment Station

FAA SW & AEH Conference | San Jose, CA
Predictability through Virtual Integration

• Reduce the risks
  – Analyze system early and throughout life cycle
  – Understand system wide impact
  – Validate assumptions across system

• Increase the confidence
  – Validate models to complement integration testing
  – Validate model assumptions in operational system
  – Evolve system models in increasing fidelity

• Reduce the cost
  – Fewer system integration problems
  – Fewer validation steps through use of validated generators
Questions?

David Redman, Ph.D.
Director
Aerospace Vehicle Systems Institute
746C HR Bright Bldg 3141 TAMU
College Station, TX  77843-3141
p:    979-862-2316
f:     979-845-1696
m:     979-218-2272
e:  dredman@tamu.edu

Dr. Don Ward
p:  254-842-5021
f:  979-845-1696
m:  979-218-2272
e:  SAVIPM@dishmail.net