Software Architecture Virtual Integration (SAVI)

Current
- Active – Boeing, Airbus, Lockheed Martin, BAE Systems, DoD (Army, Navy), FAA, GE Aviation, Rockwell Collins, SEI/Carnegie Mellon
- Joining – Dassault-Aviation, Honeywell, JPL/NASA

Potential
- Current AVSI members – DoD (Air Force), Goodrich, Hamilton Sundstrand (UTC) {Sikorsky, P&W}
- Potential new members – General Dynamics, Meggitt, Northrup Grumman, Raytheon, Thales, Woodworth
What’s Coming? More Complexity!

Airbus Code Growth. Boeing Numbers Similar.

Why AVSI?

Rapid technological advancement and obsolescence combined with increasingly complex hardware and software evolution present integration problems affecting all of us

◆ **It’s not going to get better, it’s only going to get worse**
  
  ❖ Boeing and Airbus have published data showing doubling of size and complexity every two years

◆ **We can’t afford to solve it alone**
◆ **We can’t afford to solve it multiple times**
◆ **We can’t afford not to solve it**
SAVI Project Objective

- **Overall Concept of Operations**
  - *Design and production based on early and continuous integration (virtual => physical)*
  - *Integrate, then build*

- **Objective**
  - *Shift architecting, design, and production activities to explicitly address integration issues early, reducing program execution risks, cycle time and cost*

- **Approach**
  - *Adopt/develop “virtual integration-based” software and system development processes with emphasis on integrating component-based, model-based and proof-based development*
Single Multi-Aspect Model Repository & Model Bus

Requirements

- Eclipse
- MatLab
- Simulink
- Rhapsody
- Esterel
- AADL

Verification

- OSATE
- SCADE
- TOPCASED
- DOORS

Model Repository

Design

Integration/Deployment

Aerospace Vehicle Systems Institute
Multi Phase Project

- Proof of Concept (POC) Project
  - 12 month project
  - As-is & To-Be process, ROI model, AADL-based demo
  - Initial POC demo within 2 months
  - Green light from Executive Board for pilot project

- Pilot Project
  - 3-4 years, $30+M
  - Establish practice infrastructure
  - Two phases: apply to production system, apply by product group
# PoC Prioritized Requirements

<table>
<thead>
<tr>
<th>#</th>
<th>Requirement</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Establish Model Bus infrastructure</td>
<td>Process</td>
</tr>
<tr>
<td>2</td>
<td>Establish Model Repository Infrastructure</td>
<td>Process</td>
</tr>
<tr>
<td>3</td>
<td>Inform RoI estimates through AFE58 performance &amp; results</td>
<td>Process</td>
</tr>
<tr>
<td>4</td>
<td>Analyses be conducted across the system</td>
<td>Analysis</td>
</tr>
<tr>
<td>5</td>
<td>Two or more analyses must be conducted</td>
<td>Analysis</td>
</tr>
<tr>
<td>6</td>
<td>Analyses be conducted at multiple levels of abstraction</td>
<td>Analysis</td>
</tr>
<tr>
<td>7</td>
<td>Analyses must validate system model consistency at multiple levels of abstraction</td>
<td>Analysis</td>
</tr>
<tr>
<td>8</td>
<td>Analyses must be conducted at the highest system level abstraction</td>
<td>Analysis</td>
</tr>
<tr>
<td>9</td>
<td>Model infrastructure must contain multiple model representations</td>
<td>Model</td>
</tr>
<tr>
<td>10</td>
<td>Model infrastructure must contain multiple communicating components</td>
<td>Model</td>
</tr>
</tbody>
</table>
Proof of Concept Demo

Aircraft system: (Tier 1)
- Engine, Landing, 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

- Multi-tier system & software system
- Integrator & subcontractor virtual integration
Tier 1 Analyses

Based on Top-level Aircraft Model

• Weight analysis
  • Weight limit, gross weight, net weight
  • Problem: Engine weight limit exceeded by gross weight
  • Correction: increase Engine weight limit within Aircraft weight limit
  • Analysis by plug-in
  • Analysis by MS Excel via CSV file
Tier 1 Analyses - 2

Power analysis

- Power capacity of bus
- Power supply by Engine and Aux Power Unit
- Power budget by power consumers
- Problem: total power supply over capacity
- Correction: reduce supply by APU
- Resulting problem: budgets exceed available power
- Options:
  - Reduce power consumption
  - Replace with higher capacity power bus => increased weight
Tier 2: Flight Guidance IMA Architecture

Physical view

Logical view
Tier 2 Analyses: Subsystem partitions

Flight Guidance IMA elaborated
  - Subsystems to be contracted out
  - Weight analysis revisited
  - Power analysis revisited
  - MIPS, RAM, ROM Resource Analysis
    - MIPS capacity & budget, RAM/ROM capacity & budget
    - Initial result ok, but incomplete budget assignments
  - End to end latency analysis
    - Direct mode & IMA mode for stick to surface
Tier 2 Flow Latency Analysis Results

Two end-to-end flows

- Direct stick to surface
- Stick to surface via IMA

Analysis utilizes partition rate of subsystems & any flow space latency on application components

- Lower bound of worst case end-to-end latency
- Can be extended to determine latency jitter

IMA path exceeds requirement

- For synchronous system (all processors on same clock)
- For asynchronous systems (all processors)
Flow Use Scenario through Subsystem Architecture

- Display -> IOProcessor
- Command -> Comm -> Nav
- IOProcessor -> Modem
- IOProcessor -> Nav -> Comm
- Command -> Display

Latency = Partition hops + processing + transfer
Independent clock per processor

Multiple rates and processors with independent clocks
SAVI Model Repository Requirements

Support development process & system structure
Support integrator & subcontractor
Support versioning, configurations, development branches of systems and models
Support development teams within integrator & subcontractor
Support different views & representations

- AADL Packages, refinement via extends, multiple variants
- Versioning of AADL models & fragments
Integrator – Subcontractor Negotiations

Three phase process

- Phase 1: System architecture specification & RFP
  - Integrator defines top-level system architecture & subsystem specification

- Phase 2: Joint subsystem specification refinement
  - Subcontractor evaluates subsystem specification & proposes subsystem specification refinements
  - Integrator evaluates impact of proposed changes

- Phase 3: Subsystem development & delivery
  - Subcontractor develops subsystem architecture
  - Subcontractor repeatedly delivers intermediate models for iterative virtual integration testing by integrator
Integrator – Subcontractor Repositories

Development Repositories

Integrator
- Internal
  - Integrator data dictionary
  - System Impl
  - System configurations
  - Integration test harness

Public
- Common data dictionary
- Subsys Spec
- Subsys ICD
- Possibly Public Per Sub

Subcontractor1
- Public
  - Subsys ICD rev
  - Subsys Spec rev
  - Subsys Impl
- Internal
  - Subcontractor data dictionary
  - Subsys Impl
  - Test harness

Subcontractor2
- Public

Public For Aircraft customer
- Aircraft System spec

For Aircraft System spec
Distributed POC Model Development

John G/M (Rockwell Collins)
Peter, Lutz (SEI)
Keith (BAE Systems)
Jean-Jacques (Airbus)

Subversion Model Repository at TEES
Phase 2 Proposal Evaluation

Subcontractor delivers
- Refined ICD (port group types)
- Port specifications
- Data types with base type, size, rate & measurement units
- Proposed mapping to ARINC 429 protocol in some cases

Integrator evaluates proposals
- Integrates subcontractor subsystem specs into flight guidance & aircraft model
- Integrator runs functional integration analysis
- Integrator resolves conflicts between subcontractor specs

ARINC429 Connection Consistency Marker (6 items)
- Source number bits 3 and destination number bits 4 differ
- Source number bits 4 and destination number bits 3 differ
- Source number bits 4 and destination number bits 3 differ
- Source Word ID 13 and Word ID 12 differ
- Source Word ID 13 and Word ID 12 differ

Instantiation Marker (20 items)
- Source base type uint3 and destination base type uint4 differ
- Source base type uint3 and destination base type uint4 differ

Port Connection Consistency Marker (14 items)
- Source base type uint3 and destination base type uint4 differ
- Source base type uint3 and destination base type uint4 differ
Tier 3: Subcontractor Analyses

Subcontractor models analyzed stand-alone
  • Two IMA subsystem models down to thread level
  • One subcontractor model of black box subsystem
    – Own hardware & software elaborated to executable Ada code

Process & thread allocation analysis
  • Allocation constraints
    – Not co-located
  • Scheduling analysis: RMS vs. EDF, others
  • Recording of allocation decisions

Use of alternate scheduling analysis tool
  • Explicitly recorded allocations
  • Same results
Subcontractor models integrated with Aircraft model

- Weight analysis revisited
- Power analysis revisited
  - Reduced power consumption
  - Subsystem below power budget
  - IMA power supply subsystem analysis with rollup
- MIPS, RAM, ROM Resource Analysis revisited
  - Budget rollup based on thread data
  - Reduced total based on more accurate data

```
process ImplementedAircraftSystem_AircraftSystem_FlowSubSystemServiceImpl_Instance.FGS.PR_FGS.L1 total 33.3 MIPS below budget 50.0 MIPS (33.3 % slack)
process ImplementedAircraftSystem_AircraftSystem_FlowSubSystemServiceImpl_Instance.FGS.PR_FGS.R1 total 33.3 MIPS below budget 50.0 MIPS (33.3 % slack)
process ImplementedAircraftSystem_AircraftSystem_FlowSubSystemServiceImpl_Instance.FGS.PR_FMS_i total 111.3 MIPS below budget 120.0 MIPS (7.2 % slack)
process ImplementedAircraftSystem_AircraftSystem_FlowSubSystemServiceImpl_Instance.FGS.PR_FMS_2 total 111.3 MIPS below budget 120.0 MIPS (7.2 % slack)
```

- MIPS capacity 800.000 MIPS : MIPS budget 544.333 MIPS
- RAM capacity 2.048 GB : RAM budget 140.028 MB
- ROM capacity 4.096 GB : ROM budget 632.280 MB
Virtual Integration Analyses - 2

Subcontractor models integrated with Aircraft model

- Latency analysis revisited
  - Latency has increased
  - Configure in one subsystem at a time to determine contributing subsystem task model

-- Examine latency computation trace as “Report” CSV file

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>owner</td>
<td>flow</td>
<td>model elem name</td>
<td>deadline or sampling delay</td>
<td>partition delay</td>
<td>flow spec</td>
<td>additional</td>
<td>total (ms)</td>
<td>expected</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Subcomponent DE_YOKES</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>500.0 us</td>
<td>500.0 us</td>
<td>500.0 us</td>
<td>25.0 ms</td>
</tr>
<tr>
<td>2</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Connection Implemented Aircraft System</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>8.5 ms</td>
</tr>
<tr>
<td>3</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Subcomponent THR_FCP:YokesToFGS1</td>
<td>3000.0 us</td>
<td>5.0 ms</td>
<td>0.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>8.5 ms</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Connection Implemented Aircraft System</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>8.5 ms</td>
</tr>
<tr>
<td>5</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Subcomponent THR_FGSMain:PFCLAWS1</td>
<td>50.0 ms</td>
<td>100.0 ms</td>
<td>0.0 us</td>
<td>50.0 ms</td>
<td>50.0 ms</td>
<td>50.0 ms</td>
<td>155.5 ms</td>
<td>25.0 ms</td>
</tr>
<tr>
<td>6</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Connection Implemented Aircraft System</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>158.5 ms</td>
</tr>
<tr>
<td>7</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Subcomponent THR_AP:APCLAWS1</td>
<td>3000.0 us</td>
<td>5.0 ms</td>
<td>0.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>158.5 ms</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Inst End2End</td>
<td>fStickToSurface_Norm Connection Implemented Aircraft System</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>0.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>3000.0 us</td>
<td>158.5 ms</td>
</tr>
</tbody>
</table>
```
Virtual Integration Analyses - 3

Integrator performs analyses on Flight Guidance IMA

- **Scheduling analysis**

  Scheduling Analysis Marker (3 items)
  - FlightGuidanceIMAConfiguration_FlightGuidance_a4_fullybound_Instance.CPU_1.cpu is schedulable with 55.7 % utilization
  - FlightGuidanceIMAConfiguration_FlightGuidance_a4_fullybound_Instance.CPU_2.cpu is schedulable with 75.0 % utilization
  - FlightGuidanceIMAConfiguration_FlightGuidance_a4_fullybound_Instance.CPU_3.cpu is schedulable with 54.0 % utilization

- **Network bandwidth analysis**

  | Total Bus bandwidth budget 22.0 Kbps of bound tasks with loopback within bandwidth capacity 10000.0 Kbps of net_16 |
  | Total Bus bandwidth budget 22.0 Kbps of bound tasks within bandwidth capacity 10000.0 Kbps of net_12 |
  | Total Bus bandwidth budget 22.0 Kbps of bound tasks within bandwidth capacity 10000.0 Kbps of net_16 |
  | Total Bus bandwidth budget 3.0 Kbps of bound tasks with loopback within bandwidth capacity 13.3 Kbps of ar429_10 |
  | Total Bus bandwidth budget 3.0 Kbps of bound tasks with loopback within bandwidth capacity 13.3 Kbps of ar429_9 |
  | Total Bus bandwidth budget 3.0 Kbps of bound tasks within bandwidth capacity 13.3 Kbps of ar429_10 |
  | Total Bus bandwidth budget 3.0 Kbps of bound tasks within bandwidth capacity 13.3 Kbps of ar429_9 |
  | Total Bus bandwidth budget 30.0 Kbps of bound tasks with loopback within bandwidth capacity 100.0 Kbps of ar429_3 |
  | Total Bus bandwidth budget 30.0 Kbps of bound tasks with loopback within bandwidth capacity 100.0 Kbps of ar429_4 |
Possible Other Demos

Architecture consistency checks

- Application data types, measurement units, base types
- Required connections & hardware connectivity
- System Tier consistency
- Consistency between logical and physical connections

Security analysis

Safety criticality analysis

Fault propagation & isolation analysis

Model checking of redundancy mode logic

Auto generation of runtime system
Recent SEI Customer Projects

AVSI SAVI
- Proof of Concept pilot of multi-tier modeling and analysis of aircraft architecture including integrator/subcontractor support

NASA IV&V
- Case study of JPL Mission Data System reference architecture and Validation & Verification Framework

US Army AMRDEC
- Comparative modeling of six CAAS IMA helicopter architectures

PEO Aviation
- Modeling and analysis of Apache helicopter IMA architecture in context of SEI ATAM

Automotive supplier
- Multi-dimensional variation of embedded software subsystem
Towards Architecture Centric Engineering

Build on architecture tradeoff analysis (e.g., SEI ATAM)
• Provides focused evaluation method
• MBE/AADL provides quantitative analysis & starter models to build on

Project reviews & root cause analysis
• Identify systemic risks in problem systems & in technology migration
• AADL provides semantic framework to identify issues and potential mitigation strategies

Architecture documentation of existing systems
• Leverage existing design data bases
• Challenge: abstract away from design details ("what" instead of "how")

System and software assurance
• Provides structured approach to safety/dependability assurance
• MBE/AADL provides evidence based on validated models
AADL & Other Standards

AADL & OMG MARTE

- MARTE met with SAE AADL during RFP in 2004
- Joint AADL UML profile effort
- AADL sub-profile 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
Increased Confidence through Virtual Integration

Predictive

Sensitivity analysis for uncertainty

Top-Level Verification Items

High-level AADL Model

Detailed AADL Model

Specify Model-Code Interfaces

Component Software Design

Software Architectural Design

System Design

Requirements Engineering

Validated

Confidence in implementation

Acceptance Test

System Test

Integration Test

Unit Test

Code Development

Component Software Design

Specify Model-Code Interfaces

Detailed AADL Model

High-level AADL Model

Top-Level Verification Items

Requirements Engineering

→ generation of test cases

← updating models with actual data

Model-driven artifact generation

Conformance of models and systems

AVSI SAVI POC Demo
Feiler, Apr 2009
© 2009 Carnegie Mellon University
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
POC Demonstrates Early Identification of System-level Errors

- Requirements Engineering: 20.5% (30x)
- System Design: 0%, 9% (15x)
- Software Architectural Design: 70%, 3.5% (1x)
- Component Software Design: 10%, 50.5% (5x)
- Code Development: 20%, 16% (5x)
- Integration Test: Insert your data here
- Unit Test: Insert your data here


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

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation 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.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.