

Open Source AADL Workbench for Virtual System Integration

Peter Feiler

Oct 2015

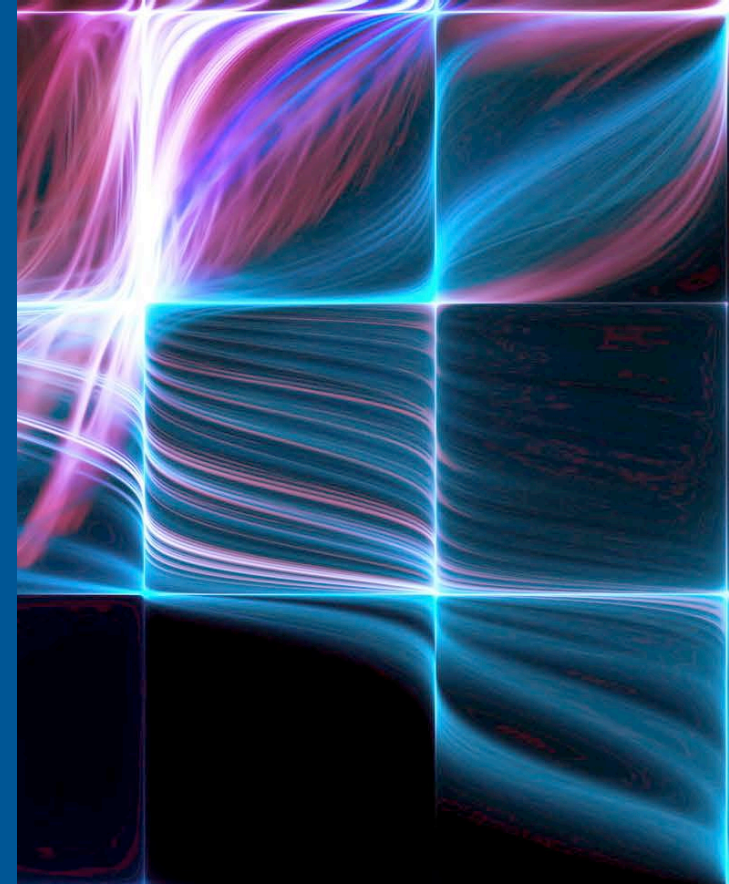
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Outline

Mission and Safety-Critical System Challenges

Virtual System Integration with SAE AADL

Samples of AADL Workbench Capabilities



We Rely on Software for Safe Aircraft Operation

Quantas Airbus A330-300 Forced to make Emergency Landing - 36 Injured

Written by htbw on Oct-7-08 1:48pm
From: soyawannaknow.blogspot.com



Thirty-six passengers were injured in a mid-air decompression emergency landing Tuesday.

The terrifying incident saw the Airbus A330-300 issue a mayday call when it suddenly changed altitude during a flight from Singapore to Perth, Qantas said.

Embedded software systems introduce a new class of problems not addressed by traditional system safety analysis

Oct. 15 (Bloomberg) -- Airbus SAS issued an alert to airlines after Australian investigators said a computer fault on a Qantas Ltd. flight switched off the autopilot and generated false data, causing the jet to nosedive.

The Airbus A330-300 was cruising at 37,000 feet (11,277 meters) when a computer fed incorrect information to the flight control system, the Australian Transport Safety Bureau said yesterday. The plane descended 650 feet within seconds, slamming passengers and crew against the ceiling, before the pilots regained control.




"This appears to be a unique event," the bureau said, adding that Toulouse, France-based Airbus, the world's largest maker of commercial aircraft, issued a telex late yesterday to airlines that fly A330-300s fitted with the same air-data computer. The advisory is aimed at minimizing the risk in the unlikely event of a similar occurrence.

FAA says software problem with Boeing 787s could be catastrophic

By Dan Catchpole
@dcatchpole

The Federal Aviation Administration says a software problem with Boeing 787 Dreamliners could lead to one of the most advanced jetliners losing electrical power in flight, which could lead to loss of control.

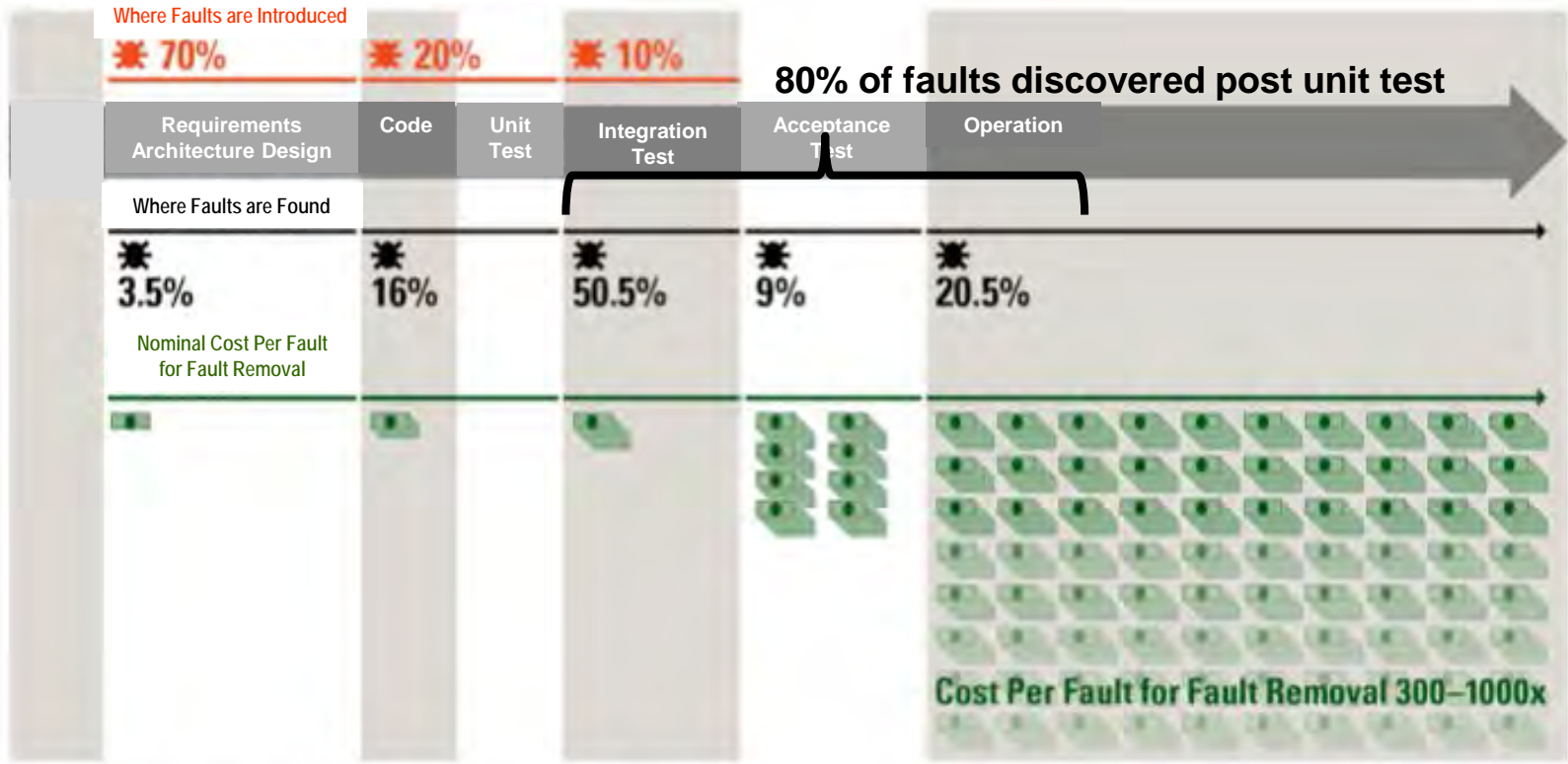
The FAA notified operators of the airplane Friday that if a 787 is powered continuously for 248 days, the plane will automatically shut down its alternating current (AC) electrical power.

-  The Buzz: Hipster's dilemma
-  Boeing & aerospace news
-  Aerospace blog

Safety-Critical System Challenges



70% of faults introduce in requirements and architecture design
80% of faults discovered post unit test



Sources: Critical Code; NIST, NASA, INCOSE, and Aircraft Industry Studies

Total System Cost
 Boeing 777 \$12B F-35 \$59B

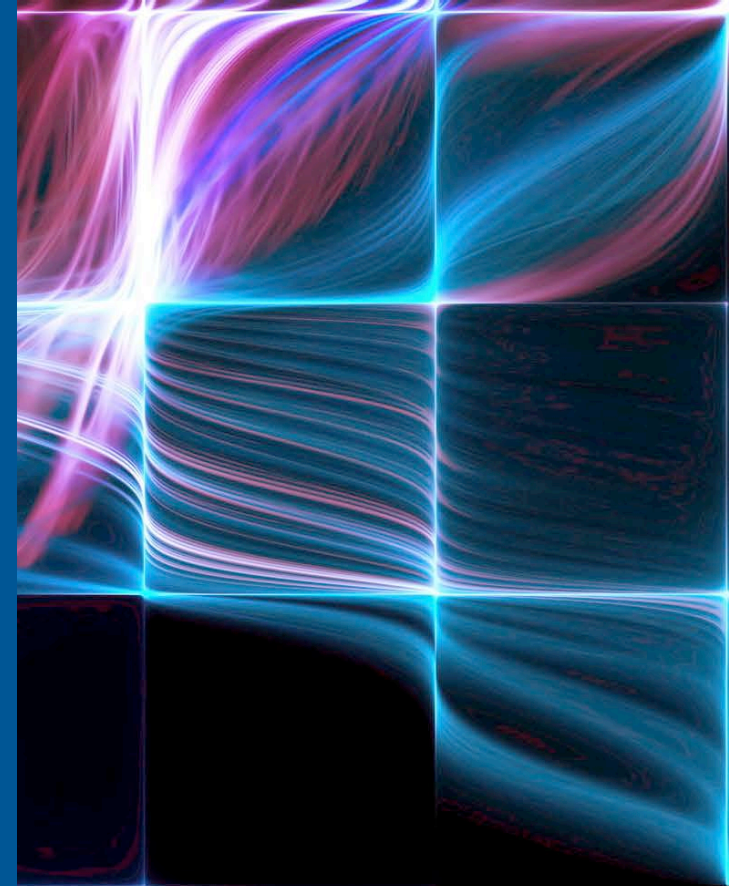
Software as % of total system development cost
 1997: 45% → 2010: 66% → 2024: 88%

Outline

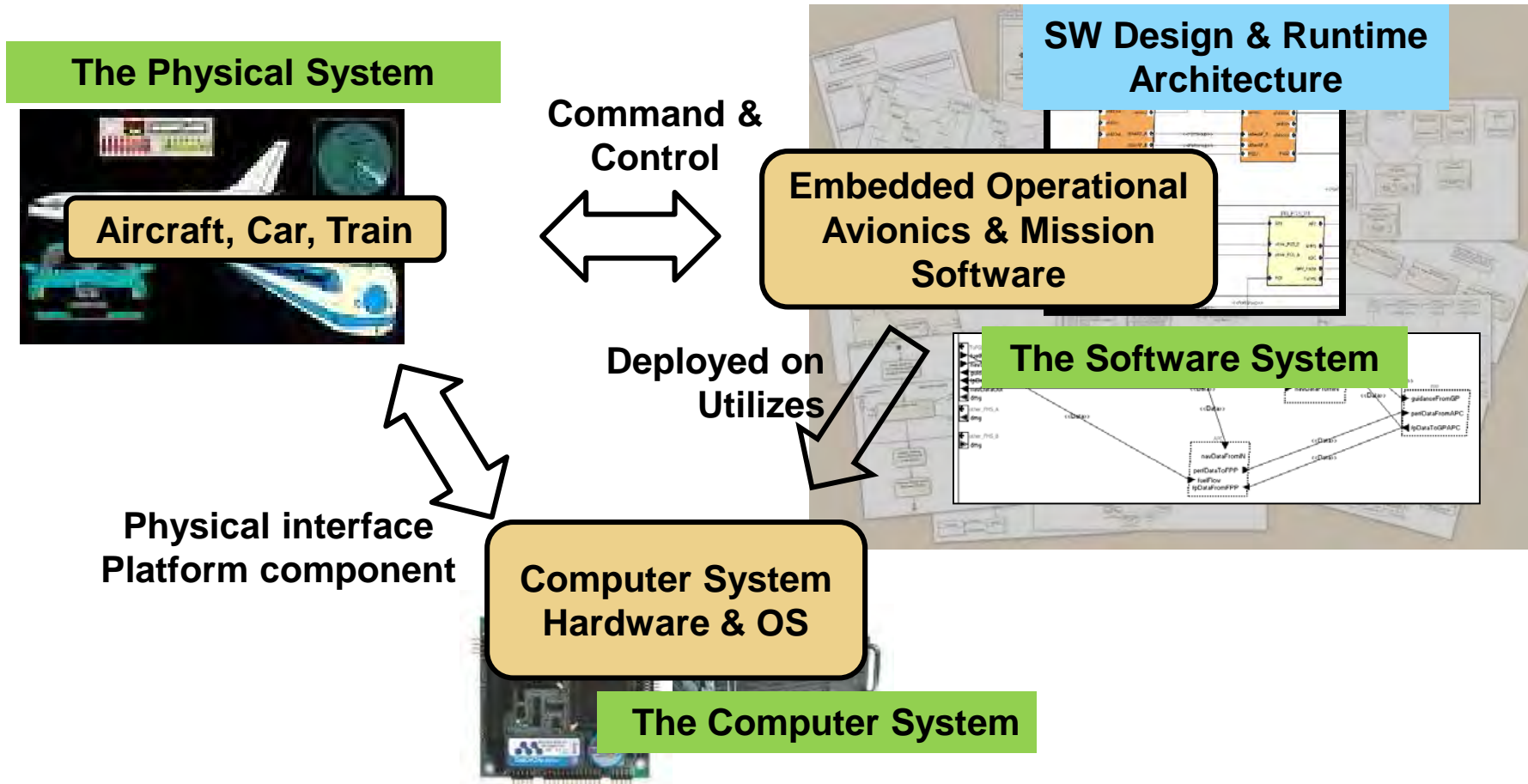
Mission and Safety-Critical System Challenges

Virtual System Integration with SAE AADL

Samples of AADL Workbench Capabilities



SAE Architecture Analysis & Design Language (AADL) Standard to the Rescue



AADL focuses on interaction between the three elements of a software-reliant mission and safety-critical systems.



Analysis of Virtually Integrated Software Systems

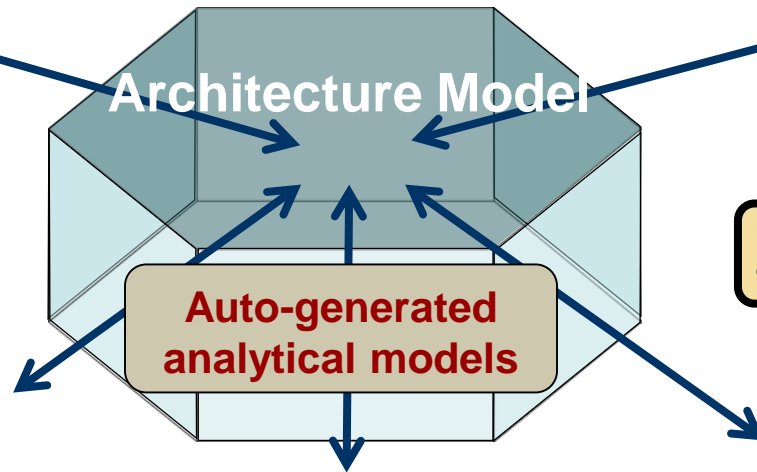
Single Annotated Architecture Model Addresses Impact Across Operational Quality Attributes

Safety & Reliability

- MTBF
- FMEA
- Hazard analysis

Security

- Intrusion
- Integrity
- Confidentiality



Potential new hazard

Change of Encryption from 128 bit to 256 bit

Data Quality

- Data precision/accuracy
- Temporal correctness
- Confidence

Affects temporal correctness

Real-time Performance

- Execution time/Deadline
- Deadlock/starvation
- Latency

Increased latency

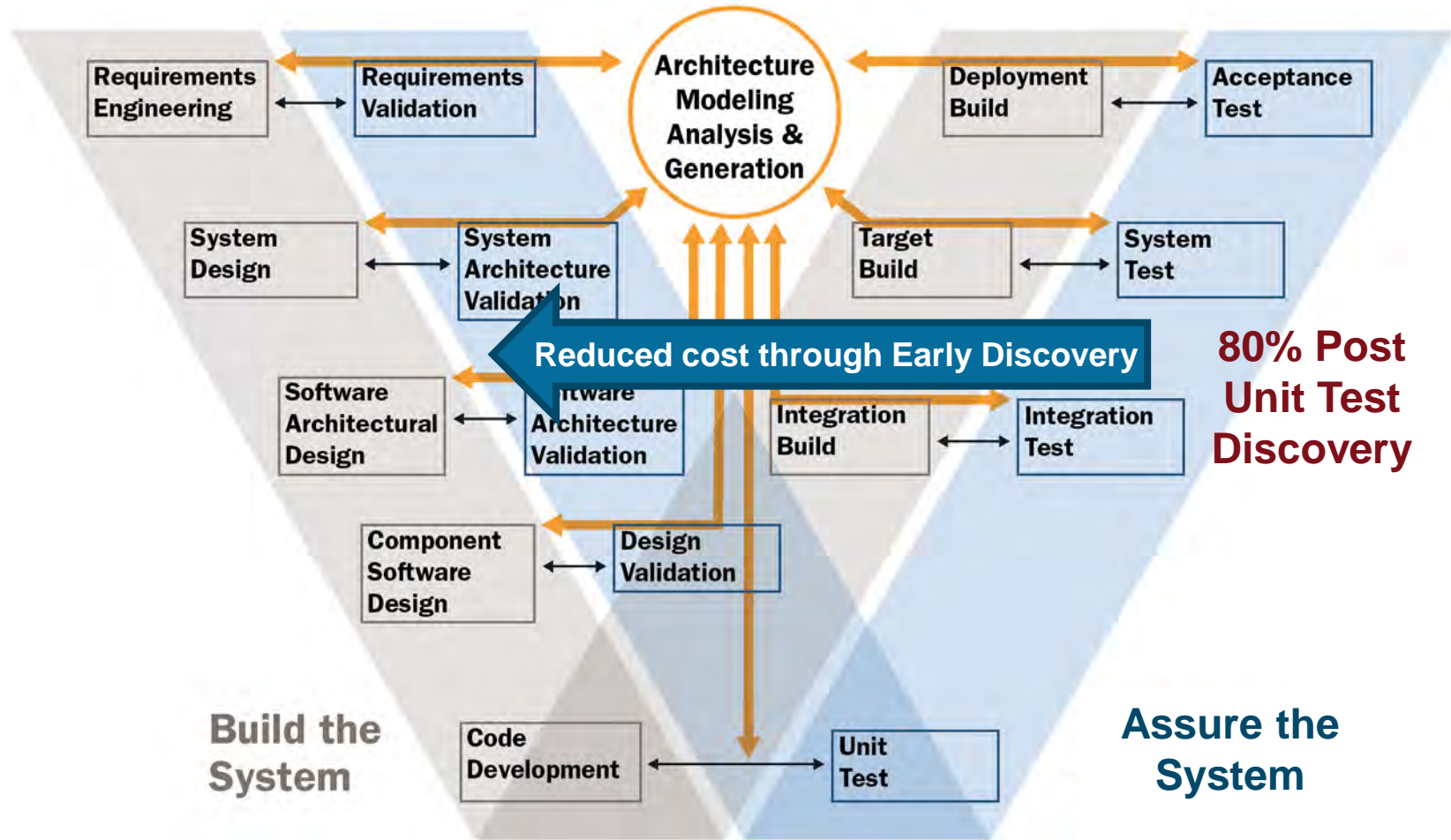
Resource Consumption

- Bandwidth
- CPU time
- Power consumption

Higher CPU demand



Early Discovery through Virtual System Integration

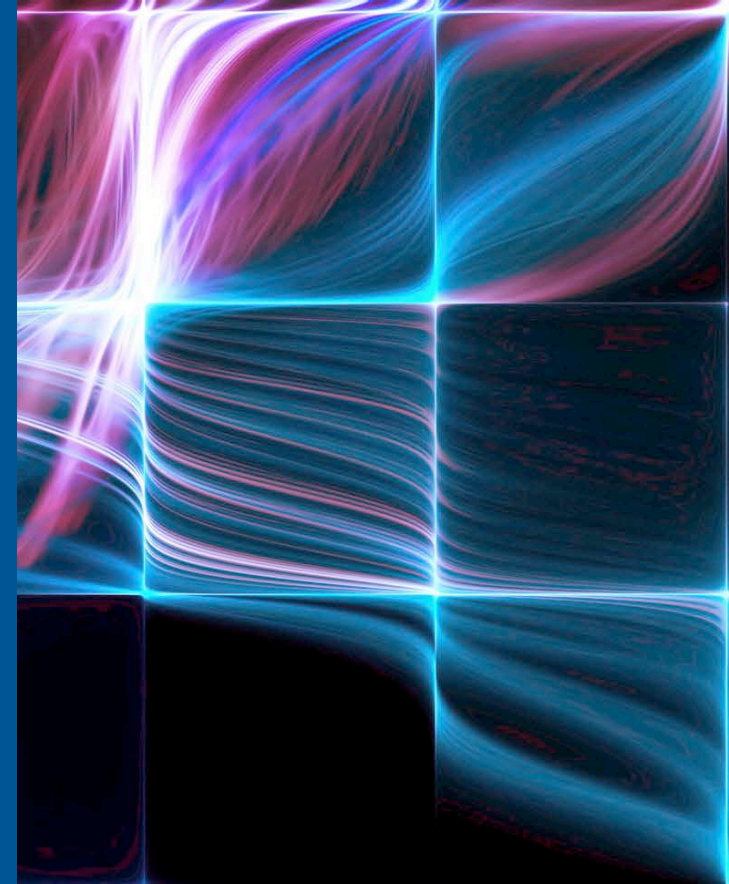


Outline

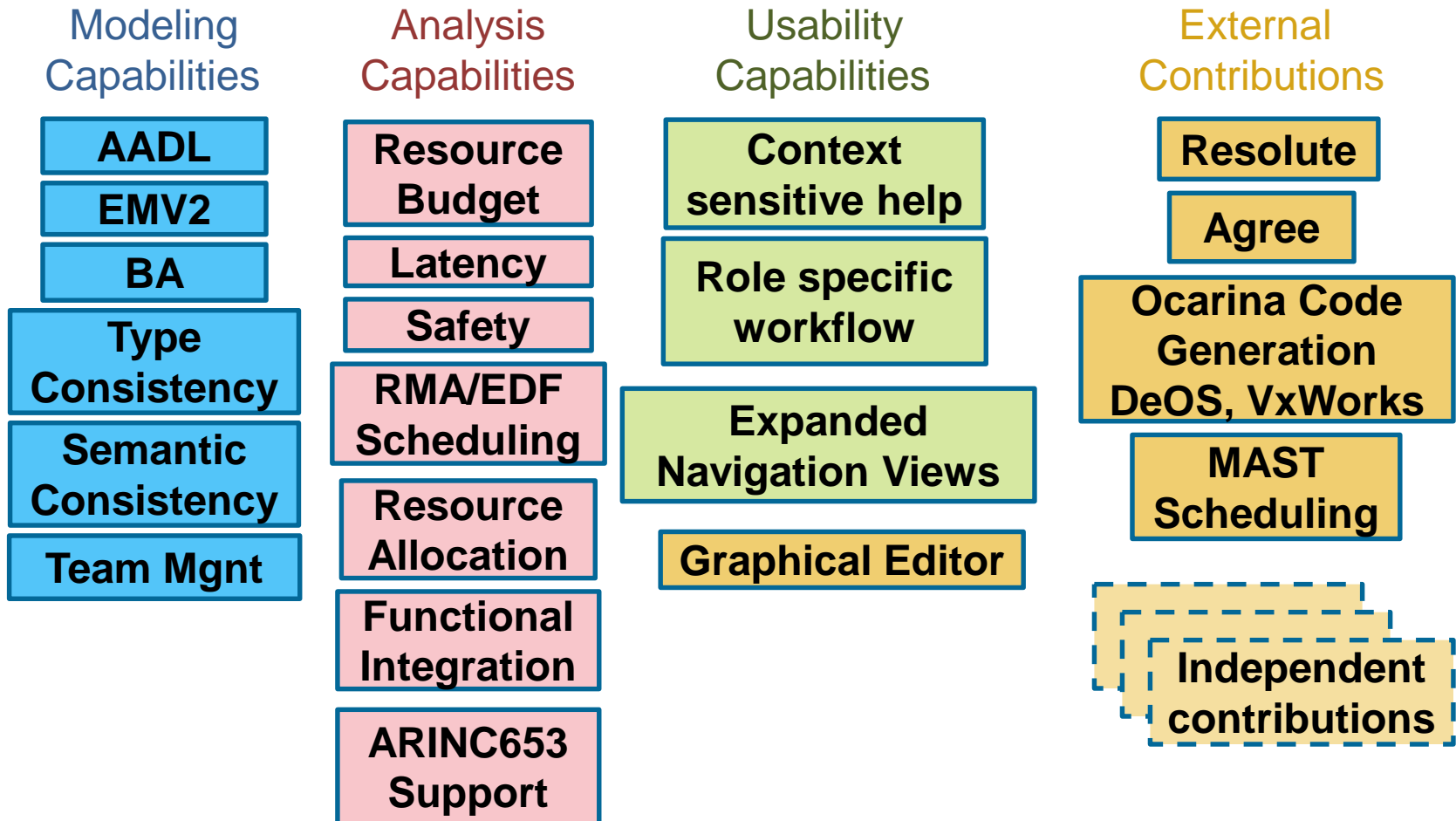
Mission and Safety-Critical System Challenges

Virtual System Integration with SAE AADL

Samples of AADL Workbench Capabilities

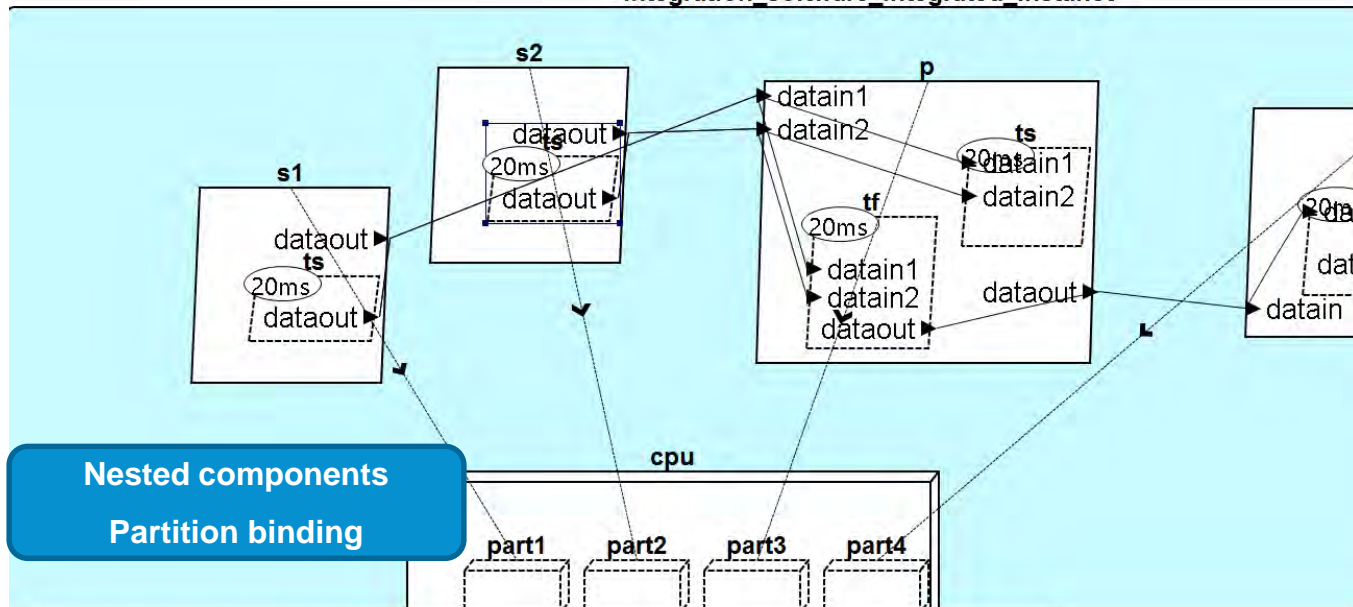


AADL Workbench* @ www.aadl.info/wiki



* aka Open Source AADL Tool Environment (OSATE)

Graphical Editing and Deployment View



Problems Properties AADL Property Values Search

Property	Value
Deployment_Properties	
Actual_Processor_Binding	(integration_software_integrated_Instance.cpu.part2)
Thread_Properties	
1 Active_Thread_Handling_Protocol	abort
1 Active_Thread_Queue_Handling_Protocol	flush
1 Deactivation_Policy	inactive
1 Dispatch_Protocol	Periodic
1 Synchronized_Component	true
Timing_Properties	
1 Compute_Execution_Time	1 ms .. 2 ms
1 Deadline	Period
1 Period	20 ms

Table-based property view and editing of selected component

Context-Sensitive Editing



Type-sensitive Data Entry

The screenshot shows an IDE window with the following components:

- Problems/Properties/AADL Property Values:** A table with a search bar and a dropdown menu set to "All".
- Property Table:**

Property	Status
Timing_Properties	
1 Compute_Execution_Time	local
1 Period	local
- Code Editor:** Contains AADL code for a process implementation. The line `Period => 8;` is highlighted. A red squiggly line under the number 8 indicates an error.
- QuickFix Popup:** A yellow box titled "Number value is missing a unit" with 7 quick fixes:
 - [Add units 'hr' to number](#)
 - [Add units 'min' to number](#)
 - [Add units 'ms' to number](#)
 - [Add units 'ns' to number](#)
 - [Add units 'ps' to number](#)
 - [Add units 'sec' to number](#)
 - [Add units 'us' to number](#)At the bottom, it says "Press 'F2' for focus".

Content Assist & QuickFix



End to End Latency Analysis



Latency analysis throughout life cycle

- Functional & system architecture: latency budgets
- Task & communication architecture: processing, sampling, transfer
- Platform architecture: partitions, protocols, computer hardware

Latency contributors

- Systems: processing, sampling, queuing latency
- Connections: protocol overhead, physical transfer, sampling
- Partitions: sampling, window schedule

Trade studies

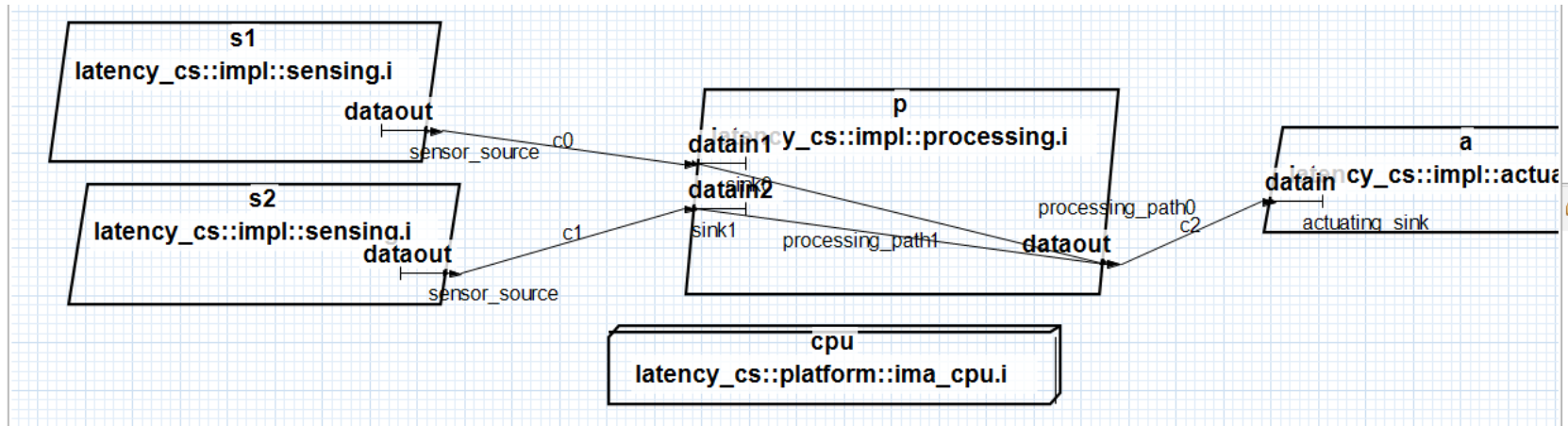
- Best-case & worst-case, latency jitter
- Mid-frame and frame-delayed communication
- Synchronous and asynchronous systems
- Partition end and major frame output policy
- Empty & full queue

Top-down & bottom-up

- Latency budgets & rate, size, time based actuals

Utilizes end-to-end flows
Incremental refinement
Interprets deployment bindings
Operational mode specific analysis

Latency Analysis Views and Results



Contributor	Min Specified	Min Value	Min Method	Max Specified	Max Value	Max Method	Comments
Partition cpu.part1		0.0ms	partition offset		0.0ms	partition offset	Initial 200.0ms partition latency not added
thread s1.ts		0.0ms	first sampling		0.0ms	first sampling	Initial 20.0ms sampling latency not added
thread s1.ts		1.0ms	processing time		2.0ms	processing time	
Partition cpu.part1		199.0ms	partition output (MF)		198.0ms	partition output (MF)	Output at 200.0ms major frame
Connection		0.0ms	no latency		0.0ms	no latency	
Partition cpu.part3		100.0ms	partition offset		100.0ms	partition offset	Synchronous communication on same platform
thread p.tf		0.0ms	sampling		0.0ms	sampling	Task period smaller than partition period
thread p.tf		2.0ms	processing time		3.0ms	processing time	
Partition cpu.part3		98.0ms	partition output (MF)		97.0ms	partition output (MF)	Output at 200.0ms major frame
Connection		0.0ms	no latency		0.0ms	no latency	
Partition cpu.part4		150.0ms	partition offset		150.0ms	partition offset	Synchronous communication on same platform
thread a.tc		0.0ms	sampling		0.0ms	sampling	Task period smaller than partition period
thread a.tc		1.0ms	processing time		3.0ms	processing time	
Immediate Connection		0.0ms	no latency		0.0ms	no latency	
thread a.td		0.0ms	no latency		0.0ms	no latency	
thread a.td		1.0ms	processing time		2.0ms	processing time	
Latency Total	0.0ms	552.0ms		0.0ms	555.0ms		
End to End Latency		20.0ms			30.0ms		
End to end Latency Summary							
ERROR	Minimum actual latency total 552.0 ms exceeds expected maximum end to end latency 30.0ms						
ERROR	Maximum actual latency 555.0ms exceeds expected end to end latency 30.0ms						

Advanced Scheduling Capabilities



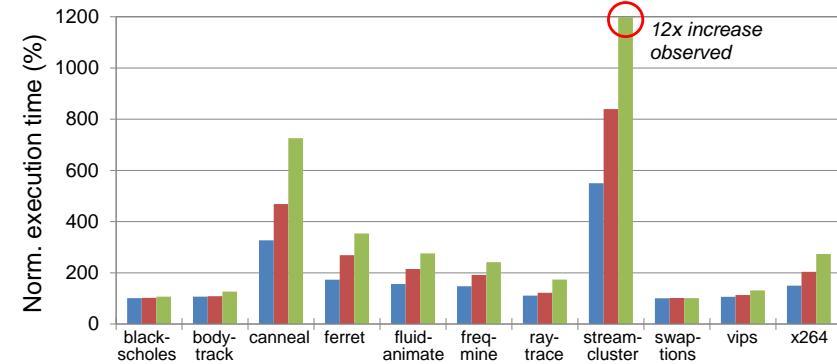
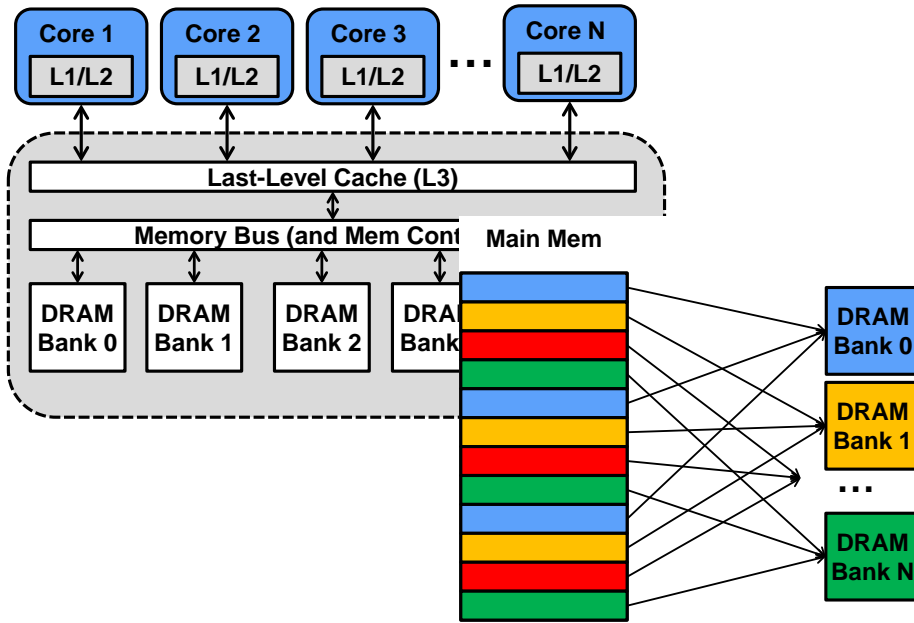
Multicore Schedulers

- Rate-Monotonic with Memory Partitioning
- Global Earliest-Deadline-First (GEDF) Scheduler for Parallelized Tasks
- Memory Profiler for Multicore Processors
- GEDF for Parallelized Task with Memory Partitioning

Mixed-Criticality Scheduling (Zero-Slack Rate Monotonic)

- Asymmetric protection: protect high-criticality tasks from lower-criticality but allow higher-criticality to steal CPU cycles from lower-criticality

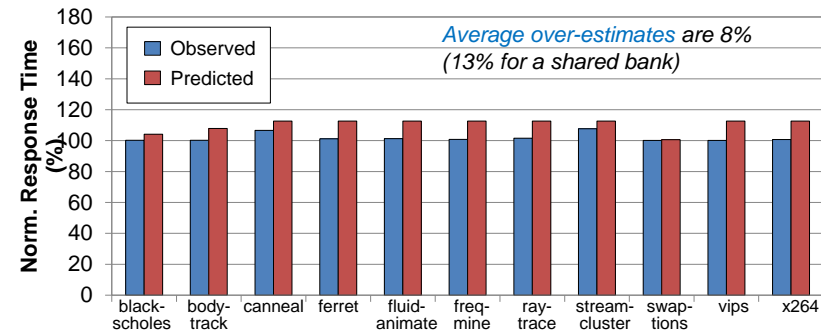
Rate Monotonic with Memory Partitioning



$$R_i^{k+1} = C_i + \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot C_j \quad \text{Classical iterative response-time test}$$

$$+ \min \left\{ \begin{array}{l} H_i \cdot RD_p + \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot H_j \cdot RD_p, \quad JD_p(R_i^k) \end{array} \right\}$$

Request-Driven (RD) Approach Job-Driven (JD) Approach



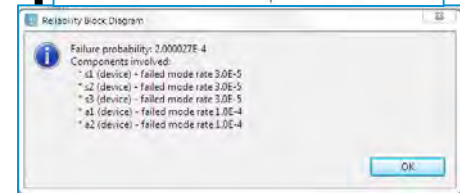
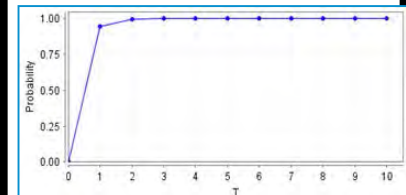
Support of SAE ARP4761 System Safety Assessment Practice



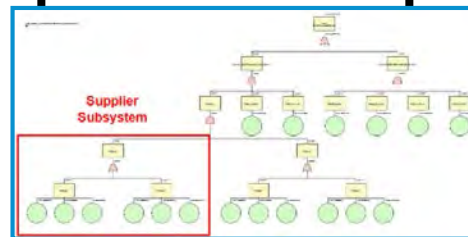
Functional Hazard Assessment (FHA)

AADL & Error Model V2

Probabilistic Reliability & Availability Analysis



Fault Tree Analysis (FTA)



Failure Mode & Effects Analysis (FMEA) Common Cause Analysis (CCA)

Item	Initial State	Initial Failure	Top Level Effect	Transition	Top Level Effect
CPU_1_cpu	ErrorFree	CPU_Failure	PermanentError	cpu_cpu_failed	PermanentError
PR_AP_L	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
CPU_2_cpu	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
PR_AP_R	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
PR_FGS_L1	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
CPU_3_cpu	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
PR_FGS_R1	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
CPU_1_cpu	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
PR_AP_L	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
CPU_2_cpu	ErrorFree	CPU_Failure	PermanentError	cpu_cpu_failed	PermanentError
PR_AP_R	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
PR_FGS_L1	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
CPU_3_cpu	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError
PR_FGS_R1	ErrorFree	ErrorFree	ErrorFree	CPU_failed(N)	PermanentError

Component	Error	Hazard Description	Crossrefer	Functional Failure	Operational P
StabilatorPositionSe	"ServiceOmission of	"No stabilator position readings due to s	"1.1.3"	"Loss of sensor readings"	"all"
StabAct1	"ServiceOmission of	"Failure to move stabilator into desired ;	"1.1.2"	"Loss of actuator functionalit	"all"
StabAct2	"ServiceOmission of	"Failure to move stabilator into desired ;	"1.1.2"	"Loss of actuator functionalit	"all"
StabilatorController	"null on ActCmd"	"Absence of computed data should signa	"1.1.1"	"Loss of guidance values"	"Approach"
StabilatorController	"null on ActCmd"	"Absence of computed data should signa	"1.1.1"	"Loss of guidance values"	"Approach"
StabilatorController	"null on ActCmd"	"Absence of computed data should signa	"1.1.1"	"Loss of guidance values"	"Approach"



Architectural Security Verification



AADL Model of QuadCopter Software System

```
only_receive_decrypt(x : component) <=
  ** "The component " x " only receives messages that pass Decrypt" **
  forall (c : connection).
    (parent(destination(c)) = x) =>
      is_sensor_data(c) or only_receive_decrypt_connection(c)

only_receive_decrypt_connection(c : connection) <=
  ** "The connection " c " only carries messages that pass Decrypt" **
  let src : component = parent(source(c));
  unalterable_connection(c) and (is_decrypt(src) or only_receive_decrypt(src))
```

```
▲ [!] only_receive_gs(ML : SOFTWARE::Main_Loop.Impl)
  ▲ [!] 'MC : SOFTWARE::Motor_Control' only receives messages from the Ground Station
    ▸ [✓] Only the Ground Station can send messages that pass Decrypt
    ▲ [!] The component 'MC : SOFTWARE::Motor_Control' only receives messages that pass Decrypt
      ▲ [!] The connection 'SN.motor_commands -> MC.motor_commands' only carries messages that pass Decrypt
        ▸ [✓] The connection 'SN.motor_commands -> MC.motor_commands' delivers data without alteration
        ▲ [!] The component 'SN : SOFTWARE::Stability_Navigation' only receives messages that pass Decrypt
          ▸ [✓] The connection 'CCT.mavlink_out -> SN.mavlink' only carries messages that pass Decrypt
          ▲ [!] The connection 'RC.commands_out -> SN.rc_commands' only carries messages that pass Decrypt
            ▸ [✓] The connection 'RC.commands_out -> SN.rc_commands' delivers data without alteration
            ▸ [!] The component 'RC : SOFTWARE::Radio_Control' only receives messages that pass Decrypt
```

Logical Connection to SW

Logical Connection to Ground Stations

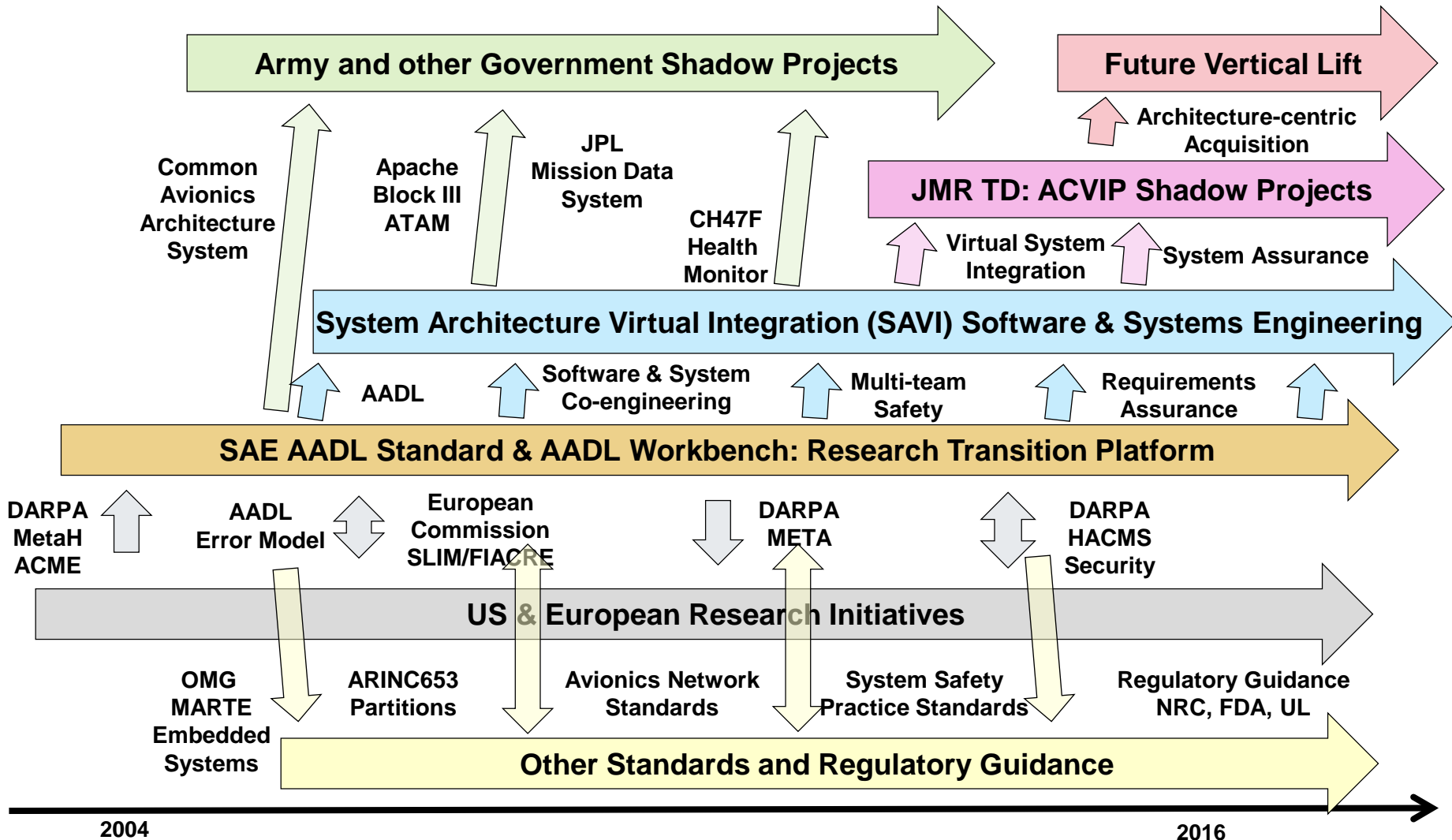
DARPA High-Assurance Cyber Military Systems (HACMS)



Secure Mathematically-Assured Composition of Control Models (SMACCM) Project



Towards an Architecture-Centric Virtual Integration Practice (ACVIP)



Contact Information



Peter Feiler/Lutz Wrage

Software Solutions Division

Email: phf/lwrage@sei.cmu.edu

U.S. Mail

Software Engineering Institute

Customer Relations

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

Web

www.aadl.info

www.aadl.info/wiki

Customer Relations

Email: info@sei.cmu.edu

SEI Phone: +1 412-268-5800

SEI Fax: +1 412-268-6257

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

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.

This material has been approved for public release and unlimited distribution except as restricted below.

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.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0002757