Validation of Safety-Critical Embedded Systems with AADL

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Peter H Feiler
April 2009
Architecture-Centric Engineering Approach

Virtual Integration & Validation of System Architecture

Availability & Reliability
- MTBF
- FMEA
- Hazard analysis

Security
- Intrusion
- Integrity
- Confidentiality

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

Resource Consumption
- Bandwidth
- CPU time
- Power consumption

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

SAE AADL Architecture Model

Auto-generated analytical models

Safety-Critical Systems Validation Feiler, April 2009
© 2008 Carnegie Mellon University
Outline

Multi-fidelity Model-based Analysis

• Physical system analysis
• Resource budget analysis
• End-to-end latency analysis
• Scheduling analysis
• Security analysis
• Fault analysis
Parts model & mass

- Processor, memory, bus/network, physical target system
- Weight limits, net weight, gross weight
- Weight budgets & recursive rollup

Electrical power

- Power system with capacity as AADL bus type
- Power supply and power budget as bus access properties
- Connected power systems
- Power system upgrade implies change in weight
Mass & Electrical Power Analysis

Hardware component specs

```
bus EtherSwitch

features
  Power: requires bus access PowerSupply {
    SEI::PowerBudget => access 0.100 W;
  };

properties
  SEI::PowerCapacity => 1.0 W;
  SEI::BandWidthCapacity => 100.0 Mbps;
  SEI::NetWeight => 5.0 kg;
end EtherSwitch;

processor Xeon

features
  HS: requires bus access EtherSwitch {
    SEI::PowerBudget => access 75.0 mW;
  };

Power: requires bus access PowerSupply {
  SEI::PowerBudget => access 4.9 W;
};

properties
  SEI::NetWeight => 0.2 kg;
end Xeon;
```

Early HW model excluded power supply

- MissionProcessor2: Xeon.solo: Sum of weights / Gross weight 0.350 kg (no limit specified)
- MissionProcessor3: Xeon.solo: Sum of weights / Gross weight 0.350 kg (no limit specified)
- PilotDisplay: MFD: Sum of weights / Gross weight 5.000 kg (no limit specified)
- Switch: EtherSwitch: Sum of weights / Gross weight 5.000 kg (no limit specified)
- ApplicationSystem: EmbeddedApp.SubSystemParts: Sum of weights / Gross weight 5.000 kg (no limit specified
- Platform: ComputingPlatform.ThreeProcessorParts: Sum of weights / Gross weight 6.050 kg (no limit specified
- mysystem.parts: Sum of weights 11.050 kg below weight limit 20.000 kg (44.7 % Weight slack)

Increase in Switch supply overloads power supply

Power system change implies change in mass

![Power analysis chart]

- Switch Power overage: supply 100.0 mW vs. budget total 225.0 mW
- MainPowerSupply Power capacity 15.0 W vs. budget total 14.8 W
- Weight Total Marker (20 items)
- mysystem.communication: Sum of weights 36.05 kg exceeds weight limit 20.0 kg
Multi-Fidelity Resource Budgeting

Resource capacities for processors, memory, bus/networks

- Compute resources: MIPS, MB, bandwidth
- Physical resources: power

Budgets for major subsystems

- Capacity and budget totals
- Early deployment decisions & resource-specific budget totals
- Port group connections & bandwidth budgets

System decomposition & budget refinement

- Budget rollup & re-negotiation

Task & communication refinement

- Rates, WCET and budgets
Basic System Architecture

Subsystem bound to processor

High speed bus

1553 bus
Initial Partition Allocations

Parts model with subsystem allocations

Overallocation of memory
Worst Case Use Scenario

Longest path for page change processing

Warning Annunciation Manager

Display Manager

Page Content Manager

Cockpit Display

Flight Manager

Flight Director

Situation Awareness

Weapons Manager

Comm. Manager

1553 Access

Indirect connectivity due to late addition
Flight Director Command Flow

Cockpit Display

Display Manager

Page Content Manager

Flight Manager

Flight Director

Request for new page

New page content
Partition-Level Flow Latency

Subsystem latency exceeds expected latency

Lower bound latency inherent to partition architecture
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
Managed Latency Jitter through Deterministic Sampling

- Navigation Sensor Processing
- Integrated Navigation
- Guidance Processing
- Flight Plan Processing
- Aircraft Performance Calculation

**AADL**

- Input-compute-output (ICO) AADL thread semantics
- Immediate and delayed data port connections for deterministic sampling

Safety-Critical Systems Validation Feiler, April 2009
© 2008 Carnegie Mellon University
Latency has increased
Software-Based Latency & Jitter Contributors

Execution time variation: algorithm, use of cache
Processor speed
Resource contention
Preemption
Legacy & shared variable communication
Rate group optimization
Protocol specific communication delay
Partitioned architecture
Migration of functionality
Fault tolerance strategy
What If Scheduling Analysis

If a system is not schedulable

Explore these options using AADL and analysis tools

- Leverage operational modes (higher fidelity)
- Use faster processor
- Add second processor
- Rewrite code to reduce worst-case execution time
- Consider lower signal processing rate for controller

Allocations/scheduling
- Binpacker (CMU)
- Cheddar (U. Brest)
- Versa (U.Penn)
- RapidRMA (TriPacific)
- In-house tools
What If Task Allocation & Schedulability

Task allocation & schedulability using binpacking technique with allocation constraints

<table>
<thead>
<tr>
<th>Processor Capacities</th>
<th>Thread Bindings</th>
<th>Message Bindings</th>
<th>Network Capacities</th>
<th>AADL Property Bindings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>% load</td>
<td>% Available/Overload</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SystemConfigurations_mys...</td>
<td>169%</td>
<td>-69%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Connected 2 processor system

<table>
<thead>
<tr>
<th>Processor Capacities</th>
<th>Thread Bindings</th>
<th>Message Bindings</th>
<th>Network Capacities</th>
<th>AADL Property Bindings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>% Load</td>
<td>% Available/Overload</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SystemConfigurations_mys...</td>
<td>100%</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>69%</td>
<td>31%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Confidentiality** concerns that sensitive data should only be disclosed to or accessed/modified by authorized users, i.e., enforcing prevention of unauthorized disclosure of information.

**Objective:** Model security attributes for an architecture to verify that data is properly accessed and handled by users and applications.

**Confidentiality frameworks**

- Bell-LaPadula framework: military applications
- Chinese wall framework: commercial applications
- Access role/role-based access framework

Low fidelity consistency checking of security levels
AADL and Safety-Criticality

Fault management

• Architecture patterns in AADL
  – Redundancy, health monitoring, …
• Fault tolerant configurations & modes

Dependability

• Error Model Annex to AADL
• Specification of fault occurrence and fault propagation information
• Use for hazard and fault effect modeling
• Reliability & fault tree analysis

Behavior validation

• Behavior Annex to AADL
• Model checking
• Source code validation

Consistency checking of safety-criticality levels
Consistency Checking of Domain Information

Data range limits and units of measurement for input & output
Setpoint constraints on data streams as bounded value deltas
Expected miss rates and miss rate contributors
State vs. state delta & guaranteed delivery
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.