Model-Based Embedded System Engineering & Analysis of Performance-Critical Systems

Peter H. Feiler
Jan 2007
Outline

Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing
• Real-time Performance: End-to-end Latency
• Security: Confidentiality Analysis
• Data Quality: Temporal Data Consistency
• Availability & Reliability: Fault Tree Analysis
• Conclusions
Software & System Engineering

System Engineering

Embedded Software System Engineering

SysML

Physical System Model

Operational System

Physical characteristics relevant to embedded application as properties in AADL model

Physical Component

Computing Platform

Embedded Software

SAE AADL

Application Domain

© 2006 by Carnegie Mellon University
A Control Engineer Perspective

with Text_IO; package Main is
begin
  type real is digits 14;
  type flag is boolean;
  x : real := 0.0;
  ready : flag := TRUE;
end Main;
© 2006 by Carnegie Mellon University
© 2006 by Carnegie Mellon University

Validation of assumptions
Data content properties
Data stream characteristics
Range, delta, miss rate, age, variation

AADL Tools
AADL Runtime
Runtime Architecture Model
Timing analysis
Reliability analysis
Refine properties

Simulink
Component Analysis
Application Code
Tune parameters

Matlab

AADL Runtime Data

package Dispatcher is
A.p1 := B.p2;
Case 10ms:
dispatch(a);
dispatch(b);
end

package Main is
begin
type real is digits 14;
type flag is boolean;
x : real := 0.0;
ready : flag := TRUE;
end

with Text_IO;
Validar simulation
AADL Tools
Runtime Data
R1 R2 R3 R4 12 12 5 6 23 34 8 8 24 23 234
R2 R3 R4 12 12 5 6 23 34 8 8 24 23 234

R1 R2 R3 R4 12 12 5 6 23 34 8 8 24 23 234

with Text_IO;
package Main is
begin
type real is digits 14;
type flag is boolean;
x : real := 0.0;
ready : flag := TRUE;
end

package Dispatcher is
A.p1 := B.p2;
Case 10ms:
dispatch(a);
dispatch(b);
end

package Main is
begin
type real is digits 14;
type flag is boolean;
x : real := 0.0;
ready : flag := TRUE;
end
Annotated Architecture Model

- Schedulability analysis
- Latency analysis
- Safety analysis
- Reliability analysis
- Fault annotations
- Timing annotations
- Alternative Hardware Bindings

Low incremental cost for additional analyses & simulations!!!
Impact Analysis from Models

- Security
  - Intrusion
  - Integrity
  - Confidentiality
- Resource Consumption
  - Bandwidth
  - CPU time
  - Power consumption
- Data Quality
  - Data precision/accuracy
  - Temporal correctness
  - Confidence
- Availability & Reliability
  - MTBF
  - FMEA
  - Hazard analysis
- Real-time Performance
  - Execution time/Deadline
  - Deadlock/starvation
  - Latency
Outline

• Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing
• Real-time Performance: End-to-end Latency
• Security: Confidentiality Analysis
• Data Quality: Temporal Data Consistency
• Availability & Reliability: Fault Tree Analysis
• Conclusions
Resource Budgeting

• Resource management throughout life cycle
• Resource budgets for processors, memory, bus/networks
  – Compute resources: MIPS, MB, bandwidth
  – Physical resources: power consumption
• Budgets for major subsystems
  – System wide & resource specific budget totals
• System decomposition & budget refinement
• Task & communication model
  – Budgets against execution & communication rates
  – Scheduling analysis
Resource Budget Properties

• Property definition: Example CPU resource

```plaintext
MIPSCapacity: aadlreal units SEI::Processor_Speed_Units applies to (processor, system);

Processor_Speed_Units : type units
    (KIPS, MIPS => KIPS * 1000, GIPS => MIPS * 1000);
```

• Property use

```plaintext
processor missionProcessor
properties
    SEI::MIPSCapacity => 1500 MIPS;
end missionProcessor;

System AvionicsSystem
properties
    SEI::MIPSbudget => 300 MIPS;
```
Resource Budget Analysis Demo

- Parts model of major subsystems
- Does the total system budget exceed the total capacity?

- Initial subsystem/partition assignment
- Does the budget total of assigned subsystems exceed processor & memory capacity?

- Subsystems interactions
- Does the bandwidth budget total exceed network backbone capacity?
- What are network/bus-specific workloads?
Outline

• Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing
  • Real-time Performance: End-to-end Latency
• Security: Confidentiality Analysis
• Data Quality: Temporal Data Consistency
• Availability & Reliability: Fault Tree Analysis
• Conclusions
Migration of Legacy Implementation

Flight Manager

From other Partitions

Periodic I/O

20Hz

To other Partitions

Shared data area

Legacy code uses shared data area

Efficient thread communication

Latecy variation

Potential Priority Inversion

Navigation Sensor Processing

Pr 2

20Hz

Integrated Navigation

Pr 3

10Hz

Guidance Processing

Pr 4

20Hz

Flight Plan Processing

Pr 6

5Hz

Aircraft Performance Calculation

Pr 9

2Hz

From other Partitions

Decreasing Priority
Intended Data Flow

- From other Partitions
  - 20Hz
  - 10Hz
  - 20Hz
  - 5Hz
  - 2Hz

- To other Partitions
  - 20Hz
  - 5Hz
  - 2Hz

- Navigation Sensor Processing
  - Pr 2
  - Pr 3
  - Pr 4

- Integrated Navigation
  - 10Hz

- Shared data area

- Guidance Processing
  - Pr 6

- Flight Plan Processing
  - Pr 9

- Aircraft Performance Calculation

- Intended flow documented in design document table
  - Priority assignment achieves desired data flow

Decreasing Priority
Flow-based Flight Manager Model

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

Data Flow:
- From Partitions to Partitions:
  - Fuel Flow
  - Nav data
  - Nav sensor data

Data Types:
- 20Hz
- 10Hz
- 5Hz
- 2Hz
- Phase delay of Periodic I/O

© 2006 by Carnegie Mellon University
Scheduling Analysis

• Scheduling protocol determines analysis
  – Processor budget for static time line (cyclic executive)
  – Rate-monotonic Analysis (RMA) for preemptively scheduled fixed-priority tasks
  – 100% utilization for Earliest Deadline First (EDF)

• What if analysis of
  – Schedulability under different dispatch & scheduling schemes
  – Miss-estimated worst case execution times (WCET)

Commercial real-time system analysis tools provide such support
Scheduling Analysis Demo

• If a single processor system is not schedulable
• Explore these options using AADL and analysis tools
  – Leverage operational modes
  – Processor speed dependent execution time
  – Rebind to different execution platform
  – Reduce worst-case execution time
  – Identify schedulable rate from sensitivity analysis results

• Might consider
  – Repartition system
  – Use faster processor
  – Add second processor
  – Rewrite code to make it faster
  – Consider lower signal processing rate for controller
Outline

• Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing

Real-time Performance: End-to-end Latency
• Security: Confidentiality Analysis
• Data Quality: Temporal Data Consistency
• Availability & Reliability: Fault Tree Analysis
• Conclusions
device brake_pedal
features
  brake_status: out data port bool_type;
flows
  Flow1: flow source brake_status;
end brake_pedal;

system cruise_control
features
  brake_status: in data port;
  throttle_setting: out data port;
flows
  brake_flow_1: flow path brake_status -> throttle_setting;
end cruise_control;

device throttle_actuator
Features
  throttle_setting: in data port float_type;
flows
  Flow1: flow sink throttle_setting;
end throttle_actuator;
High-level Flow Analysis

- Determine minimum response time, maximum command rate
- Display Manager
- Cockpit Display
- Warning Annunciation Manager
- Page Content Manager
- Flight Manager
- Flight Director
- Situation Awareness
- Weapons Manager
- Comm. Manager
- Indirect connectivity due to late addition
- 1553 Access
- Subsystems mapped to partitions
Response Time Analysis Demo

• High-level end-to-end analysis
  – Account for latency impact of partition hops
  – Account for device latency
  – Account for subsystem processing

• Detailed end-to-end latency analysis
  – AADL model from design data base
  – Task model with 165 end-to-end flows
Outline

• Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing
• Real-time Performance: End-to-end Latency

Security: Confidentiality Analysis
• Data Quality: Temporal Data Consistency
• Availability & Reliability: Fault Tree Analysis
• Conclusions
Distributed Data Processing

Freshness? Confidence? Security?

Freshness: 10s
Confidence: 70%
Security: secret
Security: Objective

- **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.

- **Means** to achieve confidentiality include enforcing access control, perform encryption, partitioning of system.

- **Confidentiality frameworks**
  - Bell-LaPadula framework: military applications
  - Chinese wall framework: commercial applications
  - Access role/role-based access framework
Bell-LaPadula: Subjects and Objects

- In Bell–La Padula, subjects operate on objects.
  - Subjects need permission, expressed as security level, to use objects.

- The security level of a subject or object is a pair:
  - Classification
    - Drawn from a partial order of classifications (e.g., unclassified < confidential < secret < top secret).
  - Set of categories
    - Drawn from a set of labels (e.g., NATO, Nuclear, Crypto).

- 
  (Class1, Set1) dominates (Class2, Set2) if and only if
    - Class1 >= Class2
    - Set1 ⊇ Set2
Example

Minimum security level required is (secret, {a,b})

Uncontrolled sanitization as (conf, {a}) is dominated by max(class_i, cat_i), i=1..3

Access matrix

\[
\begin{array}{c|c}
\text{S}_1 & \text{O}_1 \quad \text{O}_2 \quad \text{O}_3 \quad \text{O}_4 \quad \text{O}_5 \\
\hline
\text{S}_1 & \text{read, append} & \text{read, write} & \text{read} & \text{read} & \text{append}
\end{array}
\]

Error: O_4 is read-only
Warning: O_1 are O_2 shared more freely than necessary.
Analyzed System: Errors

CompleteSystem_Impl

<table>
<thead>
<tr>
<th>Src1: Producer1</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(unclassified, {A})</td>
<td>c1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>in1</td>
</tr>
<tr>
<td>(Secret, {A, B})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>in2</td>
</tr>
<tr>
<td>(Secret, {A, B})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
</tr>
<tr>
<td>c3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dest: Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
</tr>
<tr>
<td>(confidential, {B, C})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(confidential, {B, C})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(confidential, {A, B})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(confidential, {B, C})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp: Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(confidential, {A, B})</td>
</tr>
</tbody>
</table>

Problems

<table>
<thead>
<tr>
<th>Properties</th>
<th>AADL Property Values</th>
<th>Error Log</th>
<th>Progress</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 errors, 0 warnings, 0 infos</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Connection c3's source security level (confidential, {B, A}) does not equal the level of its destination (confidential, {C, B})
- The security level of output port result, (confidential, {B, A}), does not equal the level, (secret, {B, A}), of its containing component
- The security level of subcomponent dest, (confidential, {C, B}), is not dominated by the security level, (secret, {B, A}), of its containing component
Outline

• Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing
• Real-time Performance: End-to-end Latency
• Security: Confidentiality Analysis

Data Quality: Temporal Data Consistency
• Availability & Reliability: Next presentation
• Conclusions
Performance Improvement Gone Bad

A real customer experience

- Ground station to accommodate sensor load growth
  - Reduce load in network
  - Two subsystems communicate state change instead of state
- The impact
  - Other subsystems increase network load sporadically
  - Receiving subsystem goes down
- The cause
  - Transmission protocol without guaranteed delivery
  - Overload result in dropping of transmitted state deltas
  - Missing deltas result in inconsistent receiver state
Avoiding Future Mistakes

• Relevant characteristics as properties
  – State vs. state-change communication through ports
  – Bus protocols with or without guaranteed delivery

• Annotating the model
  – Application engineer characterizes data stream
  – Embedded system engineer characterizes hardware & protocols

• The analysis tool
  – Check that connections carrying state changes are bound to buses with guaranteed delivery
Capturing Domain Characteristics

• Safe upgrading of controllers
  – Data range limits and units of measurement for input & output
  – Documenting setpoint constraints as bounded value deltas
  – Consistency checking along connections

• Security levels & information flow
  – Components with security levels
  – Security levels & containment hierarchy
  – Security levels and connections
  – Security levels & execution platform components

• Safety criticality & control of components
  – Component have safety criticality levels
  – Impact on high criticality components
Outline

• Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing
• Real-time Performance: End-to-end Latency
• Security: Confidentiality Analysis
• Data Quality: Temporal Data Consistency

Availability & Reliability: Next presentation
• Conclusions
Outline

• Model-based Embedded System Engineering
• Resource Consumption: Resource Budgeting
• Real-time Performance: Concurrency & Timing
• Real-time Performance: End-to-end Latency
• Security: Confidentiality Analysis
• Data Quality: Temporal Data Consistency
• Availability & Reliability: Fault Tree Analysis

Conclusions
Predictable Embedded System Engineering

- Tracking of requirements, planned and actual resource usage
- Analysis at multiple levels of fidelity
- Multiple analysis perspectives from annotated architecture model
- Validated timing & data stream semantics for control algorithm and software implementation
- Codified engineering rules of thumb
Model-Based Engineering Benefits

- Analyzable models drive development
  - Prediction of runtime characteristics at different fidelity
  - Bridge between control & software engineer
  - Prediction early and throughout lifecycle
  - Reduced integration & maintenance effort

- Benefits of AADL as SAE standard
  - Common modeling notation across organizations
  - Single architecture model augmented with properties
  - Interchange & integration of architecture models
  - Tool interoperability & integrated engineering environments
SEI AADL Application & Education

• Open source AADL Tool environment (OSATE)
• Embedded Engineering With AADL Tutorial
• Public two-day course offering by SEI
  – *Model-based Engineering with SAE AADL*
• Pilot projects with customers
• Case studies of Model-based Engineering
• AADL User Guide & Embedded Systems Engineering Handbook
For more information:

Peter H. Feiler
Email: phf@sei.cmu.edu
Phone: 412-268-7790
SEI: www.sei.cmu.edu
AADL: www.aadl.info