



Carnegie Mellon
Software Engineering Institute

The Society of Automotive Engineers (SAE) Architecture Analysis & Design Language (AADL) Standard

An International Industry Standard for
Embedded & Real-time Systems



Bruce Lewis

Chair, SAE AS-2C Subcommittee

Army AMCOM SED

bruce.a.lewis@us.army.mil

256-876-3224



Carnegie Mellon
Software Engineering Institute

Peter Feiler

Technical lead, editor

Software Engineering Institute

phf@sei.cmu.edu

412-268-7790

SAE





SAE AADL Standard

An Enabler of Predictable Model-Based Embedded System Engineering

- Notation for specification of task and communication architectures of Real-time, Embedded, Fault-tolerant, Secure, Safety-critical, Software-intensive systems
- Fields of application: Avionics, Automotive, Aerospace, Autonomous systems, ...
- Based on 15 Years of DARPA funded technologies
- Standard approved & published Nov 2004
- www.aadl.info

SAE





SAE AS-2C AADL Subcommittee

- Bruce Lewis (US Army AMRDEC): Chair
- Peter Feiler (SEI): technical lead, author & editor
- Steve Vestal (Honeywell): co-author
- Ed Colbert (USC): UML Profile of AADL
- Joyce Tokar (Pyrrhus Software): Ada & C Annex

Other Voting Members

- Boeing, Rockwell, Honeywell, Lockheed Martin, Raytheon, Smith Industries, General Dynamics, Airbus, Axlog, European Space Agency, TNI, Dassault, EADS, High Integrity Solutions

Coordination with

- NATO Aviation, NATO Plug and Play, French Government COTRE, SAE AS-1 Weapons Plug and Play, OMG UML & SysML



AADL-Based System Engineering

System Analysis

- Schedulability
- Performance
- Reliability
- Fault Tolerance
- Dynamic Configurability

System Integration

- Runtime System Generation
- Application Composition
- System Configuration

Software
System
Engineer

Architecture
Modeling
Abstract, but
Precise

SAE AADL

Predictive
Embedded System
Engineering
Reduced
Development &
Operational Cost

Application
Software

Execution
Platform

Composable
Components

Ac
Target
Recognition

Guidance
& Control

Supply

Mec

Information
Fusion

Ambulato

& Signal
Processing

GPS	DB	HTTPS	Ada Runtime
-----	----	-------	-------------

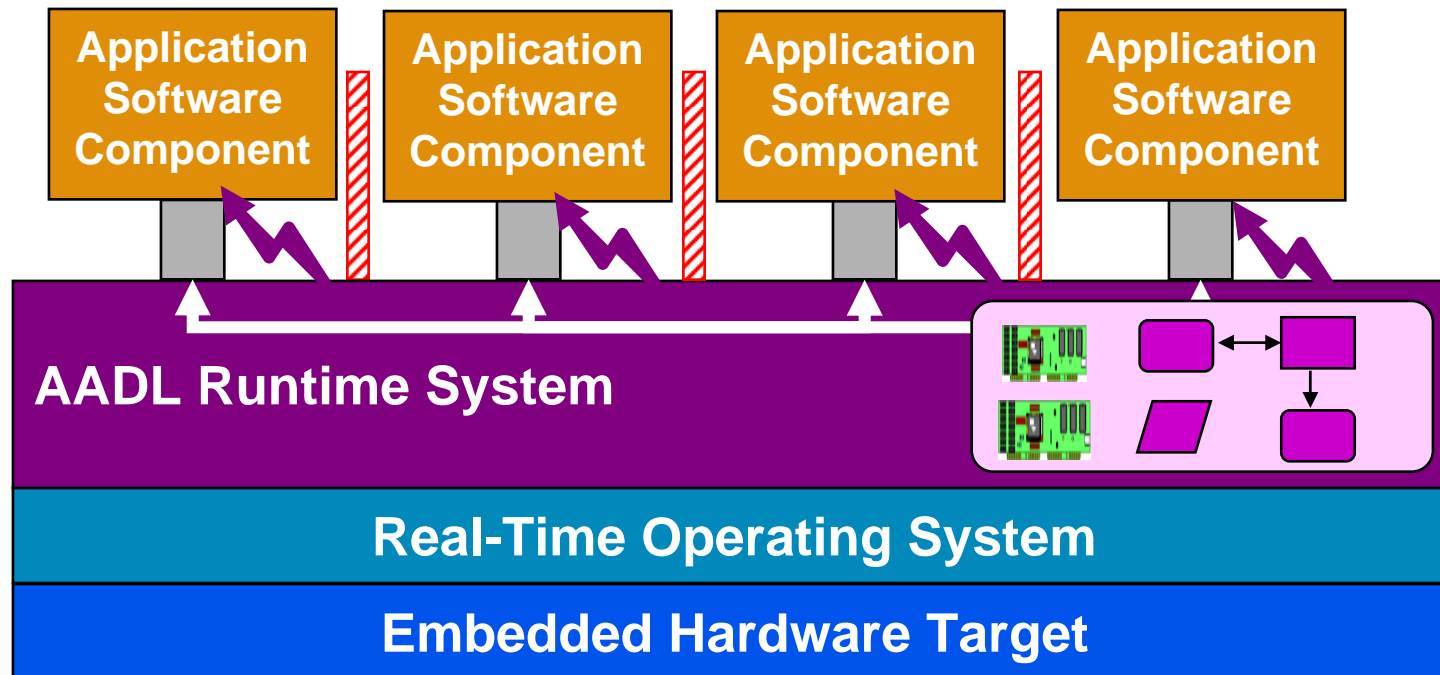
.....

Devices	Memory	Bus	Processor
---------	--------	-----	-----------





A Partitioned Portable Architecture



Strong Partitioning

- Timing Protection
- OS Call Restrictions
- Memory Protection

Interoperability/Portability

- Tailored Runtime Executive
- Standard RTOS API
- Application Components





The AADL Standard

- Requirements document SAE ARD 5296
 - Input from aerospace industry
 - Balloted and approved in 2000
- SAE AADL document SAE AS 5506
 - Core language published by SAE Nov 2004
- In review to be balloted late 2004
 - Graphical AADL notation **Alignment with industry practice**
 - UML profile of AADL for UML 1.4 and UML 2.0
 - AADL Meta model, XMI domain model, XML schema
 - Ada and C Annex
- In development **Key to tool interoperability**
 - Reliability Modeling Annex
 - Partitioning Annex (ARINC653)





AADL: The Language

Components with precise semantics

- Thread, thread group, process, system, processor, device, memory, bus, data, subprogram

Completely defined interfaces & interactions

- Data & event flow, synchronous call/return, shared access
- End-to-End flow specifications

Real-time Task Scheduling

- Supports different scheduling protocols incl. GRMA, EDF
- Defines scheduling properties and execution semantics

Modal, configurable systems

- Modes to model transition between statically known states & configurations

Component evolution & large scale development support

AADL language extensibility





AADL Language Extensions

- New properties through property sets
- Sublanguage extension
 - Annex subclauses expressed in an annex-specific sublanguage
- Project-specific language extensions
- Language extensions as approved SAE AADL standard annexes
- Examples
 - Error Model
 - ARINC 653 Partition
 - Behavior
 - Constraint sublanguage

SAE





Two-Tier Tool Strategy

- Open Source AADL Tool Environment (OSATE)
 - Developed by SEI
 - Low entry cost solution (no cost CPL)
 - Multi-platform support based on Eclipse
 - Vehicle for in-house prototyping of project specific architecture analysis
 - Vehicle for architecture research with access to industrial models & industry exposure to research results
- Commercial Tool Support
 - UML tool environment extension based on UML profile
 - Extension to existing modeling environment with AADL export/import
 - Analysis tools interfacing via XML or XML to native filter
 - Runtime system generation tools

Artisan, Rational, ...

TNI Stood





Benefits

- Model-based system engineering benefits
 - Analyzable architecture models drive development
 - Predictable runtime characteristics at different modeling fidelity
 - Model evolution & tool-based processing
 - 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 analysis properties
 - Interchange & integration of architecture models
 - Tool interoperability & extensible engineering environments
 - Aligned with UML-based engineering practices





Two Case Studies

- Full-scale analysis & integration
 - Port of missile guidance system
 - Tool-supported analysis & generation
- Pattern-based analysis of systemic issues
 - Modernized avionics system architecture
 - Change in real-time architecture concepts

SAE





MetaH Case Study at AMCOM

- Reengineered Missile Application
 - Missile on-board software and 6DOF environment simulation originally in Jovial
 - Ported to Ada83, executing on dual i80960MC, Tartan Ada, VME Boards
 - Built to Generic Missile Reference Architecture
 - Specified in MetaH, 12 to 16 concurrent processes
 - Timing analysis early in reengineering effort
 - Runtime executive generated by MetaH toolset
 - MetaH reduced total re-engineering cost 40% on first project it was used on. Missile prime estimated savings at 66%.

SAE





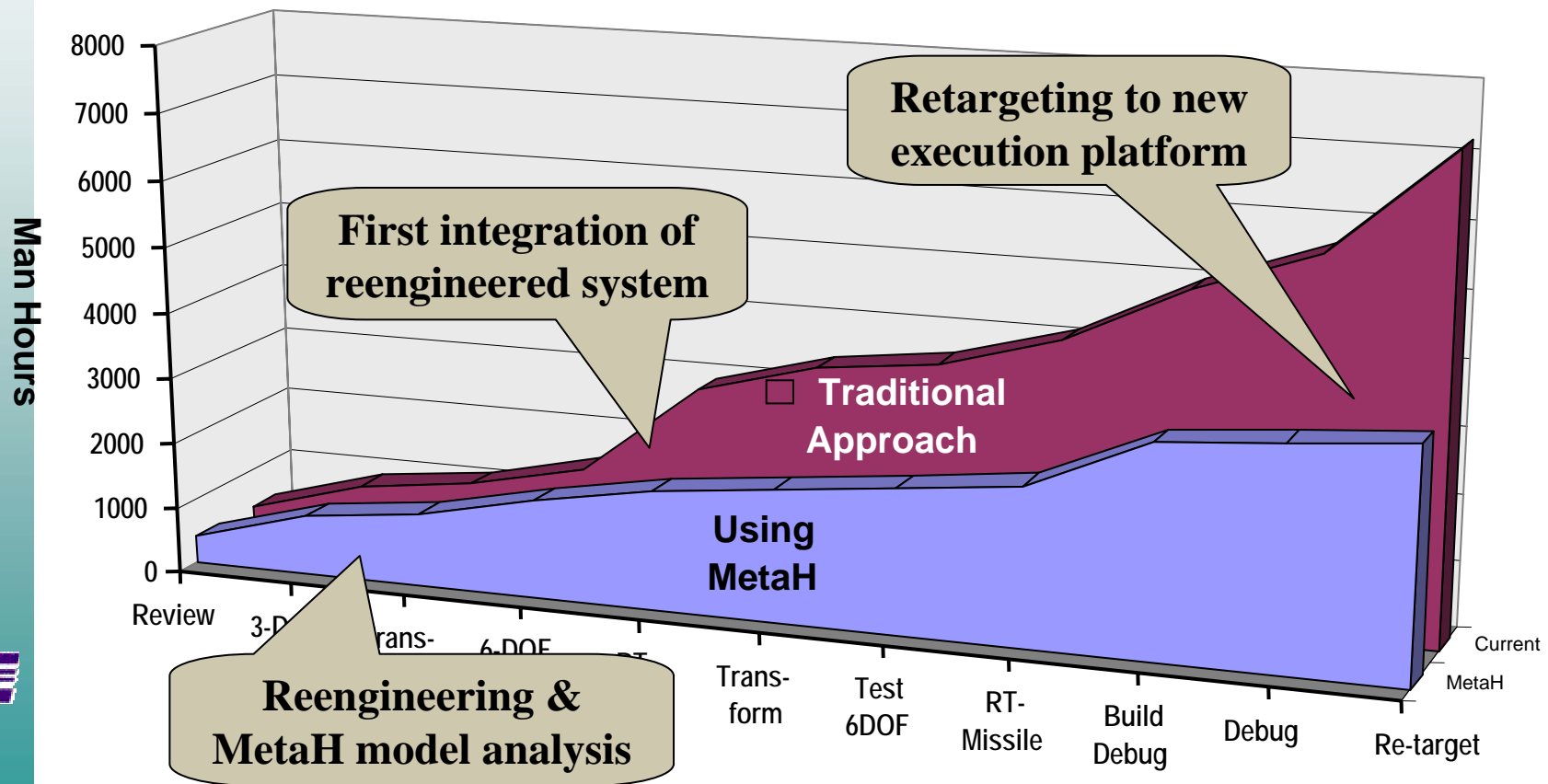
MetaH Case Study at AMCOM - 2

- Missile Application ported to a new execution environment
 - Multiple ports to single and dual processor implementations
 - New processors (Pentium and PowerPC), compilers, O/S
 - First time executable, flew correctly on each target environment
 - Execution platform description and binding specification in MetaH model
 - Port of runtime executive virtual machine to new processor & O/S
 - Ports took a few weeks rather than 10 months



AMCOM Effort Saved Using MetaH

Total project savings 50%, re-target savings 90%





AADL-Based Pattern Analysis

- SAE AADL employs
 - Components with precisely defined execution semantics
 - Explicit component interactions
 - Separation of concerns
- Pattern-based architecture analysis approach
 - Uses design patterns in analysis
 - Identifies systemic problems early
 - Enables the right choices with confidence
 - Provides analysis-based decisions

SAE





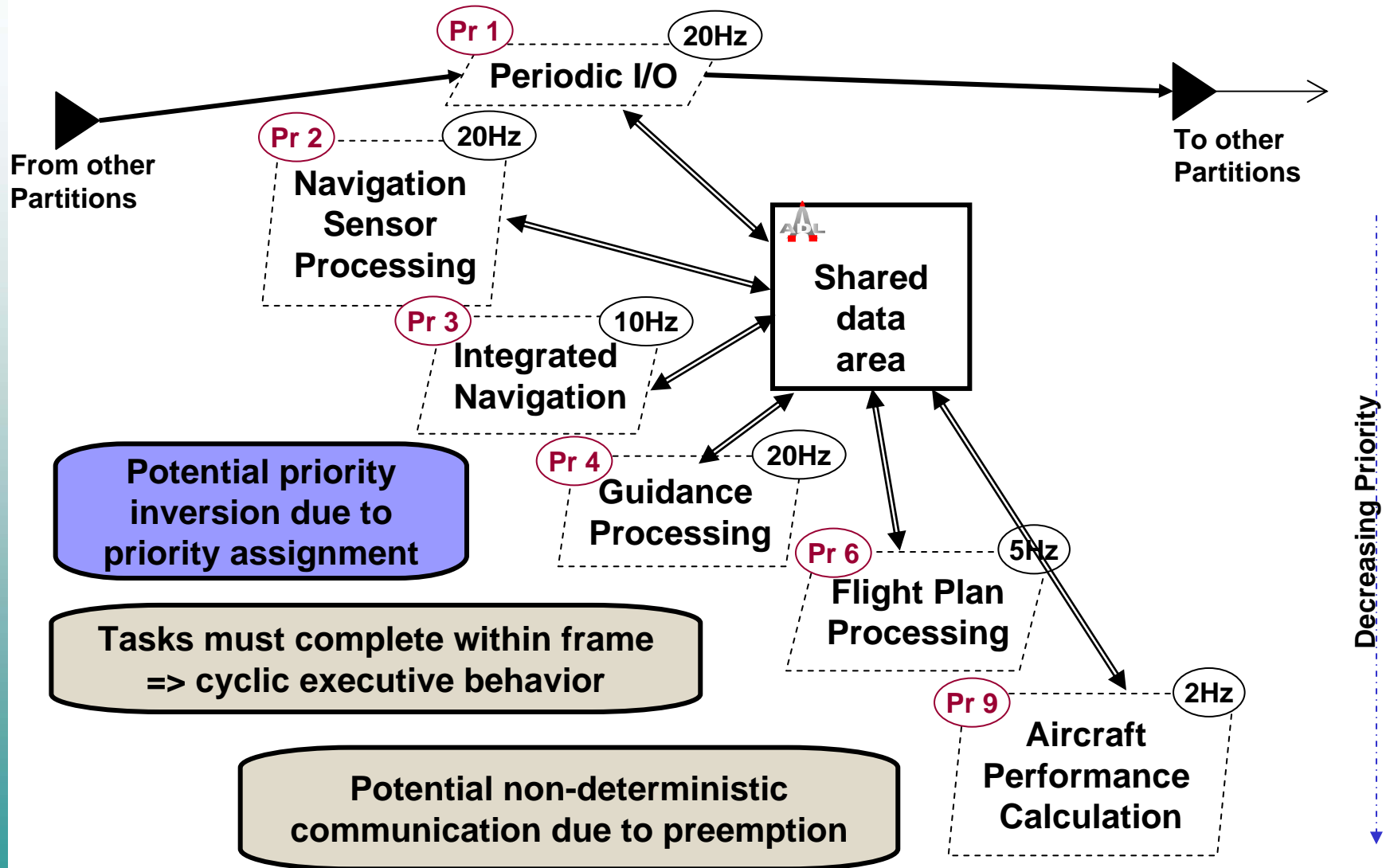
An Avionics System Case Study

- Migration from static timeline to preemptive scheduling
 - Identified issues with shared variable communication
 - Migration potential from polling tasks to event-driven tasks
- Flexibility, predictability & efficiency of port-based communication
 - Support for deterministic transfer & optimized buffers
- Effectiveness of connection & flow semantics
 - Bridge to control engineers
 - Insulate from partition scheduling decisions
 - Support end-to-end latency analysis
- Analyzable fault-tolerant redundancy patterns
 - Orthogonal architecture view without model clutter

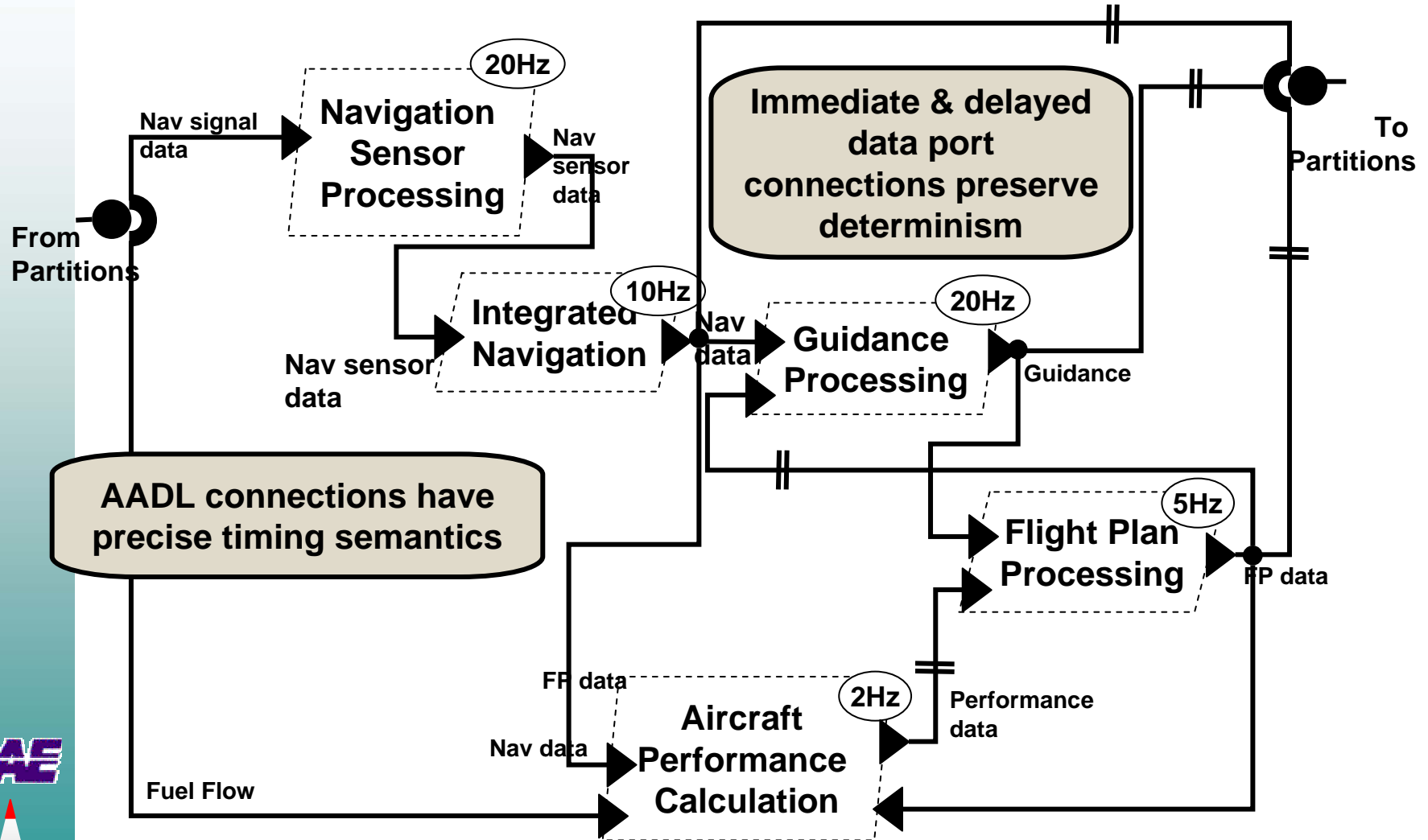




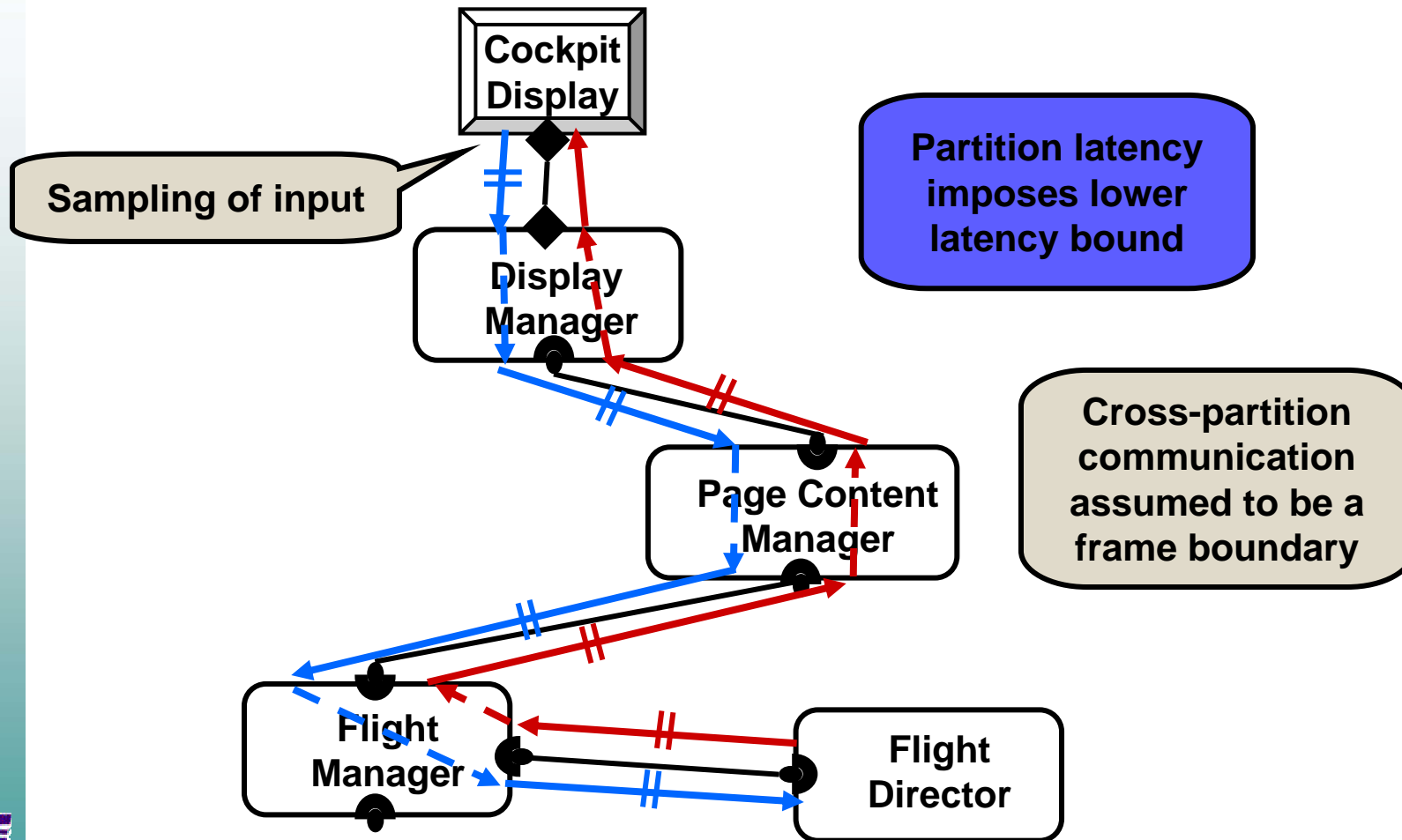
A Naïve Thread-based Design



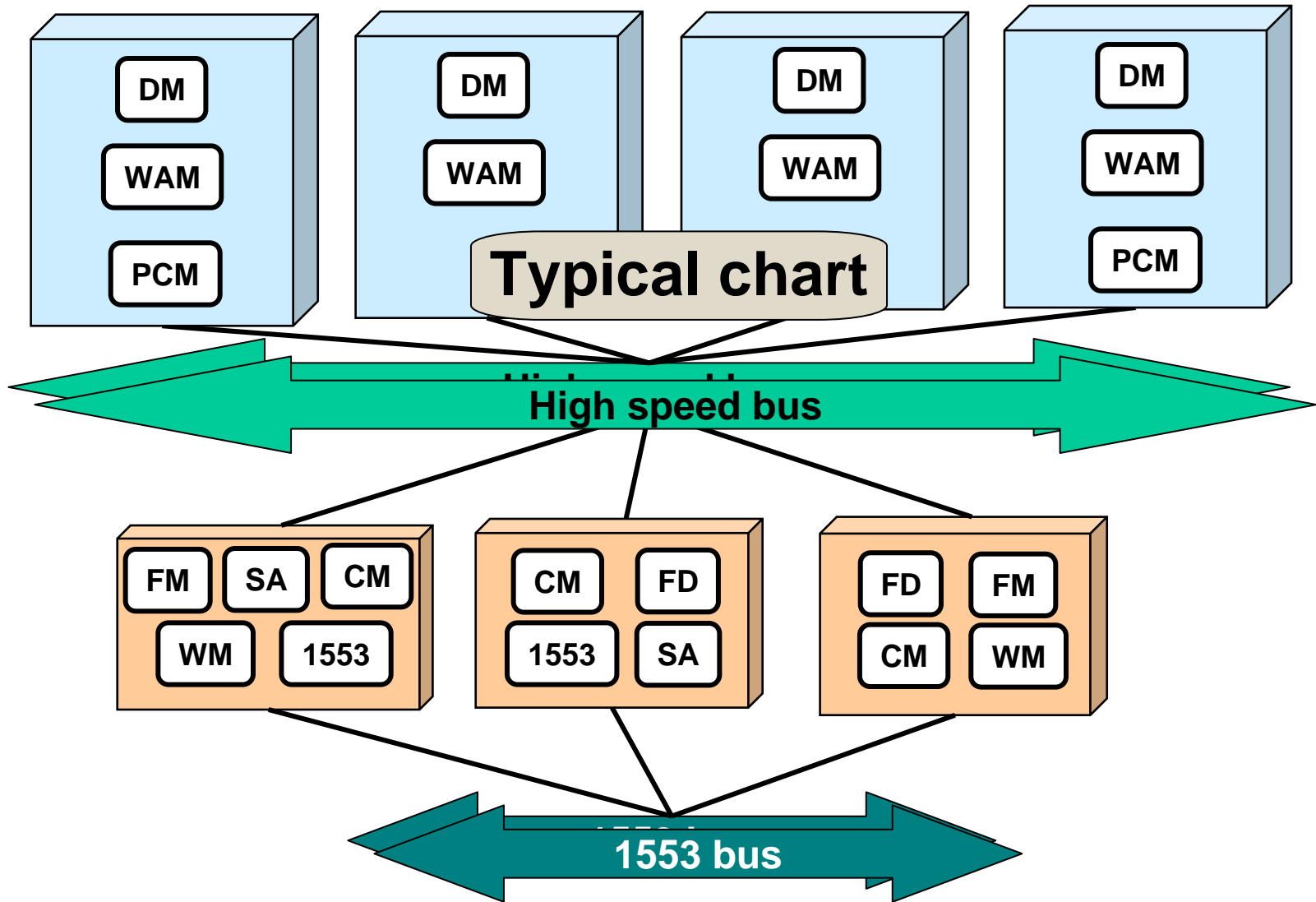
Flight Manager in AADL



Command Flow Timing

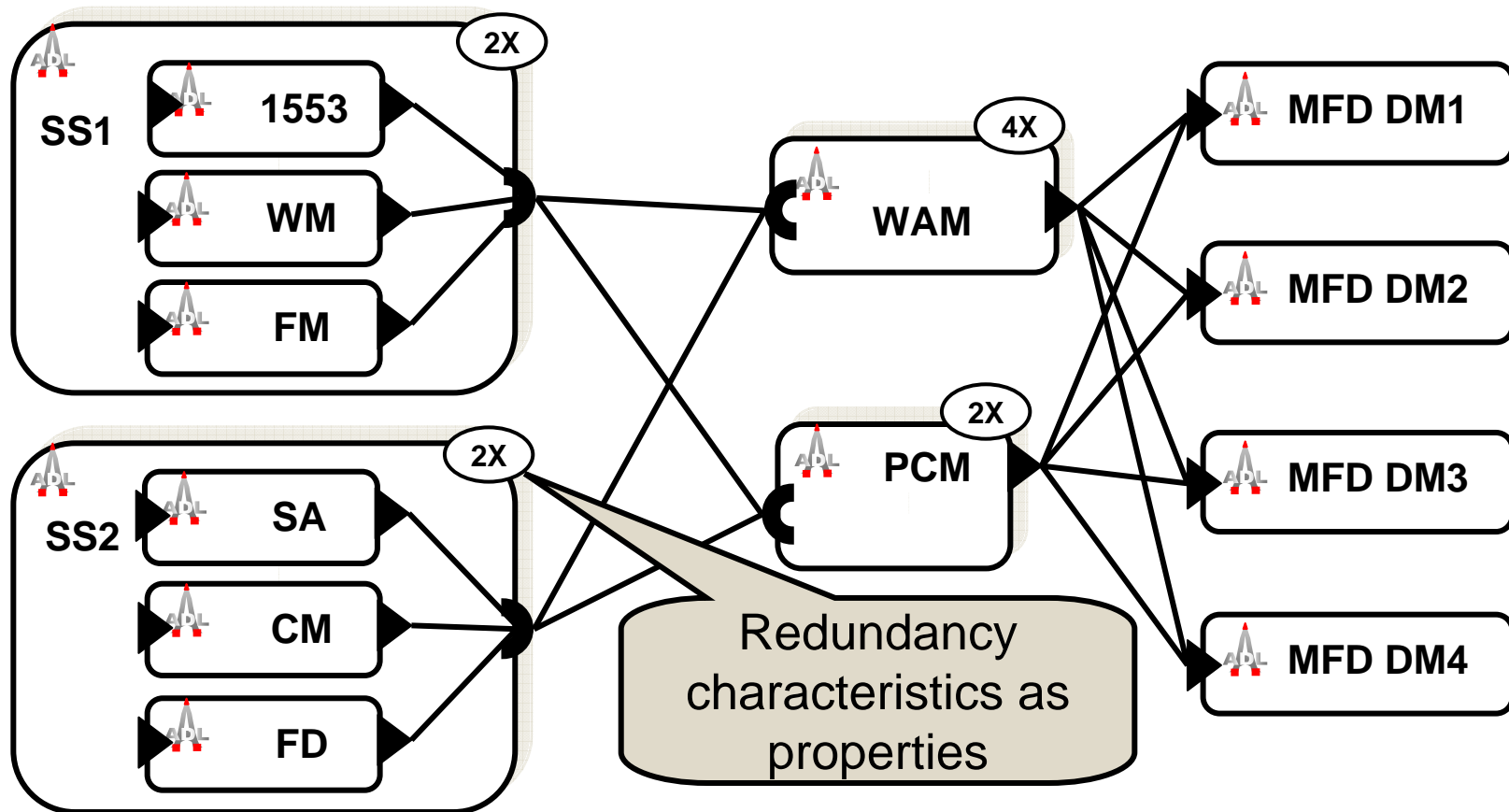


System Redundancy



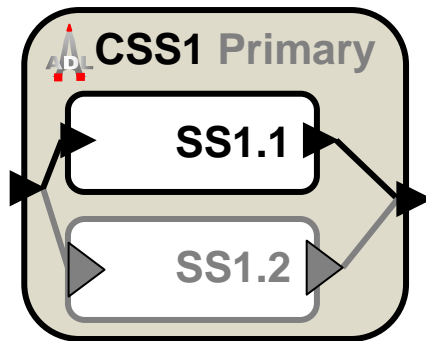
Redundancy Specification

- Redundancy abstraction
- Co-location constraints on execution platform binding

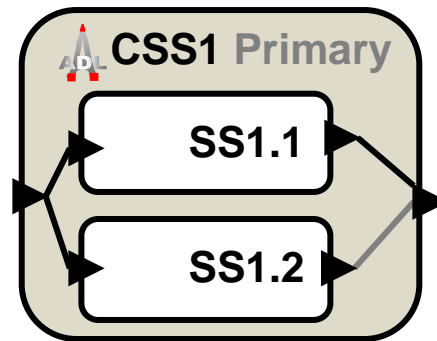


Primary/Backup Patterns

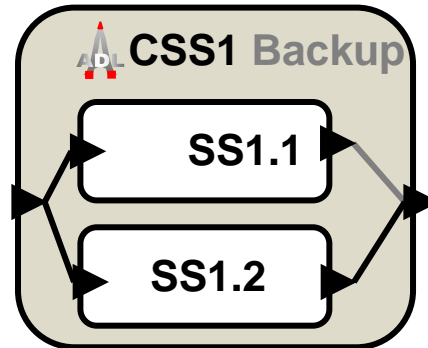
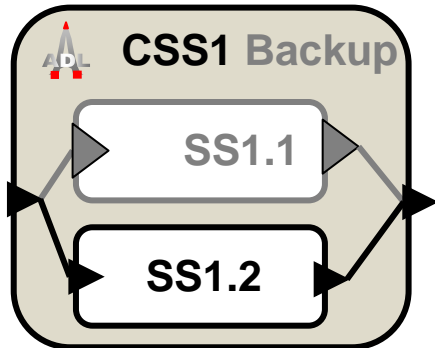
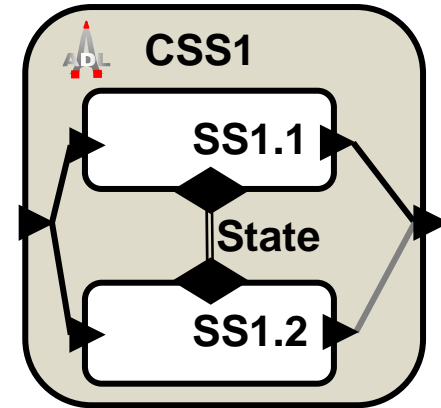
Passive Backup



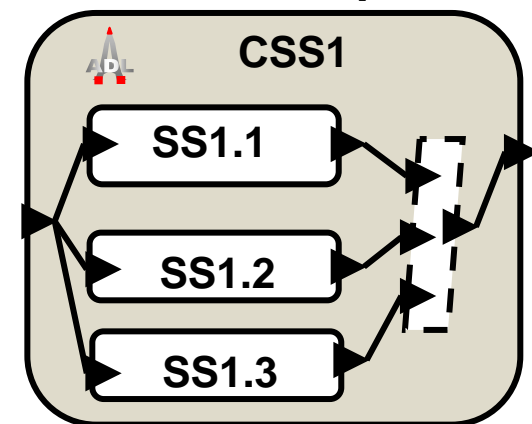
Hot Standby



Continuous State Exchange



Voted Output





AADL In Use

- Examples of system modeling & analysis
- Modeling of reference architectures
- Verification of system architectures
- SBIR & STTR projects

SAE



Automating Timing and Safety Analyses from Architecture Specifications

Steve.Vestal@Honeywell.com

9 February 2005



Honeywell

Architecture Model

Hardware architecture specification

- 8 central processors (CPs)

- 18 I/O processors (IOPs)

- a switched network architecture

 - 12 switches, 62 point-to-point cables, mix of 10Mbps and 100Mbps

 - Used for safety analysis

 - Used for globally asynchronous end-to-end timing analysis

- a time-triggered architecture

 - 8 multi-drop time-triggered busses, 10Mbps

 - Used for globally time-triggered end-to-end timing analysis

 - Used for globally asynchronous end-to-end timing analysis

Software Architecture

Function/application specifications were generated from a preliminary spreadsheet listing signal data.

Total of 1322 signals to/from 40 functions (includes redundant sensors/signals).

Honeywell

Age Scheduling Results

Connections were merged (multiplexed) if

- They had the same route between the same processors
- The connected processes had the same periods
- 2644 merged to 610

Processes at same rates on same IOM (but not CP) were merged

- 1472 merged to 203

We modeled every CP, IOM, and bus as a resource

Workload (AMPL model):

- 1425 variables (one for each process/processor and message/bus pair)
- 1872 constraints (one for each resource and one for each signal)

Model generated from AADL spec in about 45 seconds

Feasible solution found by CONOPT in about 45 seconds

Honeywell

Analyzable and Reconfigurable AADL Specifications for IMA System Integration

David Statezni
Advanced Technology Center
Rockwell Collins, Inc.

Proof of Concept Example

Generic Display System with Rockwell Collin's Switched Ethernet LAN

- ⇒ Only LAN-related entities modeled
- ⇒ Model generated from Input/Output & Thread data stored in Database

Model Size

- ⇒ 5 Common Processing Modules
- ⇒ 13 Virtual Machines
- ⇒ 90 Threads
- ⇒ 165 End-to-end Data Flows

A detailed image of an F-35 fighter jet in flight, viewed from a low angle. The aircraft is white with grey accents and has the number "98-30" and a cross symbol on its side. It is flying towards the right. The background is a vibrant, abstract digital landscape with swirling patterns in shades of blue, green, and orange, overlaid with a grid of glowing lines and various alphanumeric characters like "E6", "38", "S 2", "GUX", "27A", "12X76", and "365".

ASAAC Modelling with AADL

André Windisch

SAE AS-2 Meeting on AADL

Edinburgh, July 2004

NATO Fighter Reference Architecture

Summary

- **ASAAC configuration and reconfiguration behaviour modelled in terms of AADL events and moding**
- **ASAAC application modelling based on AADL processes, threads, data ports, and connections**
 - Formalisation of translation scheme
 - Provision of templates for ASAAC modelling
- **Platform modelling based on hierarchical refinement (as suggested by Peter Feiler)**
 - Formalise refinement approach for incorporation into tools
- **Application and communication refinement according to OSI reference model**
 - Covers data flow – control flow transformation
 - Applicable for 2 adjacent protocol layers only
- **Synchronisation with ARINC modelling required**

SAE AS-1 Weapons Plug'n'Play Reference Architecture

GENERAL DYNAMICS
Advanced Information Systems

AADL and the Plug and Play Weapon

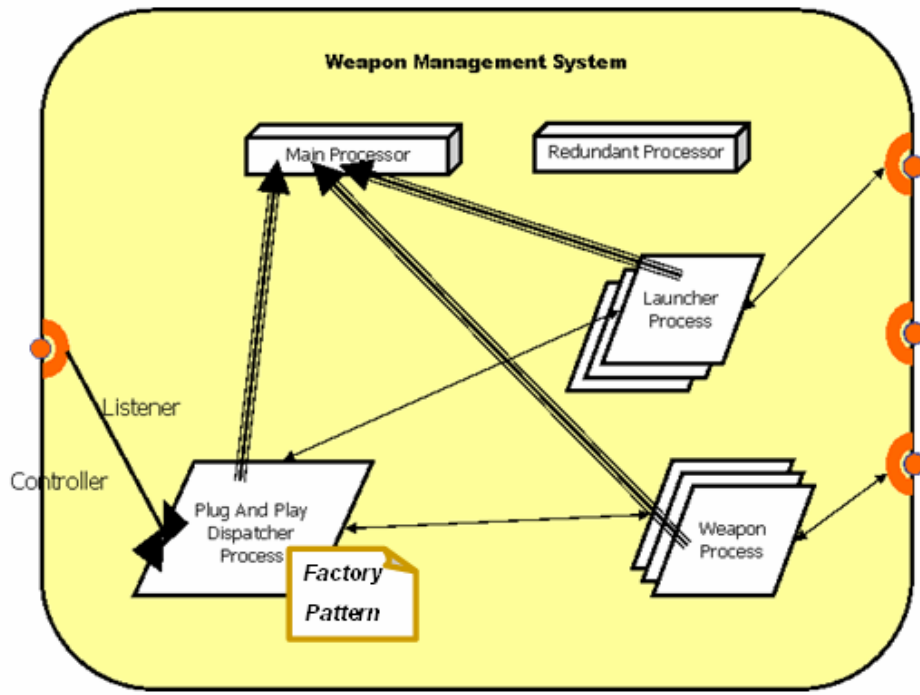
Early Experience Using the Architecture Analysis & Design Language

TC04

Yves LaCerte

3 November 2004

WMS System Implementation Example



```

system implementation WMS.WMS
subcomponents
...
connections
...
modes
...
end WMS.WMS ;
    
```

subcomponents

- processors
- processes
- bindings

connections

- port groups

modes

- MainMode: initial mode;**
- BackupMode: mode;**

Product Line



AIRBUS France

TNI-Valiosys

E.N.S.T. Bretagne

I.R.I.T.

L.A.A.S.

O.N.E.R.A. - C.E.R.T.

COTRE as an AADL profile

- Funded by the French research department (total 1.9M€, 230 m.m), from 2002 to 2004
- Goal : Real Time architecture verification (mainly from the behavioral point of view)
- Exploration project aiming to develop a demonstration tool
- Partners : AIRBUS, TNI, IRIT, LAAS, ONERA-DTIM, ENSTB

AADL
Edinburgh
meeting
July 12-15,
2004

copyright ©COTRE
all rights reserved



Example Annex Extension

```

THREAD t
FEATURES
  sem1 : DATA ACCESS semaphore;
  sem2 : DATA ACCESS semaphore;
END t;

```

```

THREAD IMPLEMENTATION t.t1
PROPERTIES
  Period => 13.96ms;
  cotre::Priority => 1;
  cotre::Phase => 0.0ms;
  Dispatch_Protocol => Periodic;

```

COTRE thread
properties

```

ANNEX cotre.behavior {**
STATES
  s0, s1, s2, s3, s4, s5, s6, s7, s8 : STATE;
  s0 : INITIAL STATE;
TRANSITIONS
  s0 -[ ]-> s1 { PERIODIC_WAIT };
  s1 -[ ]-> s2 { COMPUTATION(1.9ms, 1.9ms) };
  s2 -[ sem1.wait ! (-1.0ms) ]-> s3;
  s3 -[ ]-> s4 { COMPUTATION(0.1ms, 0.1ms) };
  s4 -[ sem2.wait ! (-1.0ms) ]-> s5;
  s5 -[ ]-> s6 { COMPUTATION(2.5ms, 2.5ms) };
  s6 -[ sem2.release ! ]-> s7;
  s7 -[ ]-> s8 { COMPUTATION(1.5ms, 1.5ms) };
  s8 -[ sem1.release ! ]-> s0;
**);
END t.t1;

```

COTRE behavioral annex



ASSERT

*Automated proof-based **S**ystem and **S**oftware
Engineering for **R**eal-**T**ime systems*

Eric Conquet

ESA/ESTEC

TEC-EME, Software Engineering and Standardization

Noordwijk, The Netherlands

- **Related strategic objective: Embedded Systems**
- **Type of instrument: Integrated Project**
- **Number of partners: 29**
- **Project cost: 15 M€**
- **Amount of EC funding: 8.3 M€**
 - *Roughly 50% of the project cost (the rest is funded by the partners)*
- **Total duration of the project: 3 Years.**
- **Starting date: 1st September 2004.**



A Research Transition Platform

- SBIR contract requires use of AADL
 - Eglin AFB, 21st Century Systems
 - Weapons Plug'n'Play compatibility analysis
- STTR contract uses AADL
 - U. Penn, Fremont Associates
 - Map hybrid control system language (Charon) into AADL

SAE



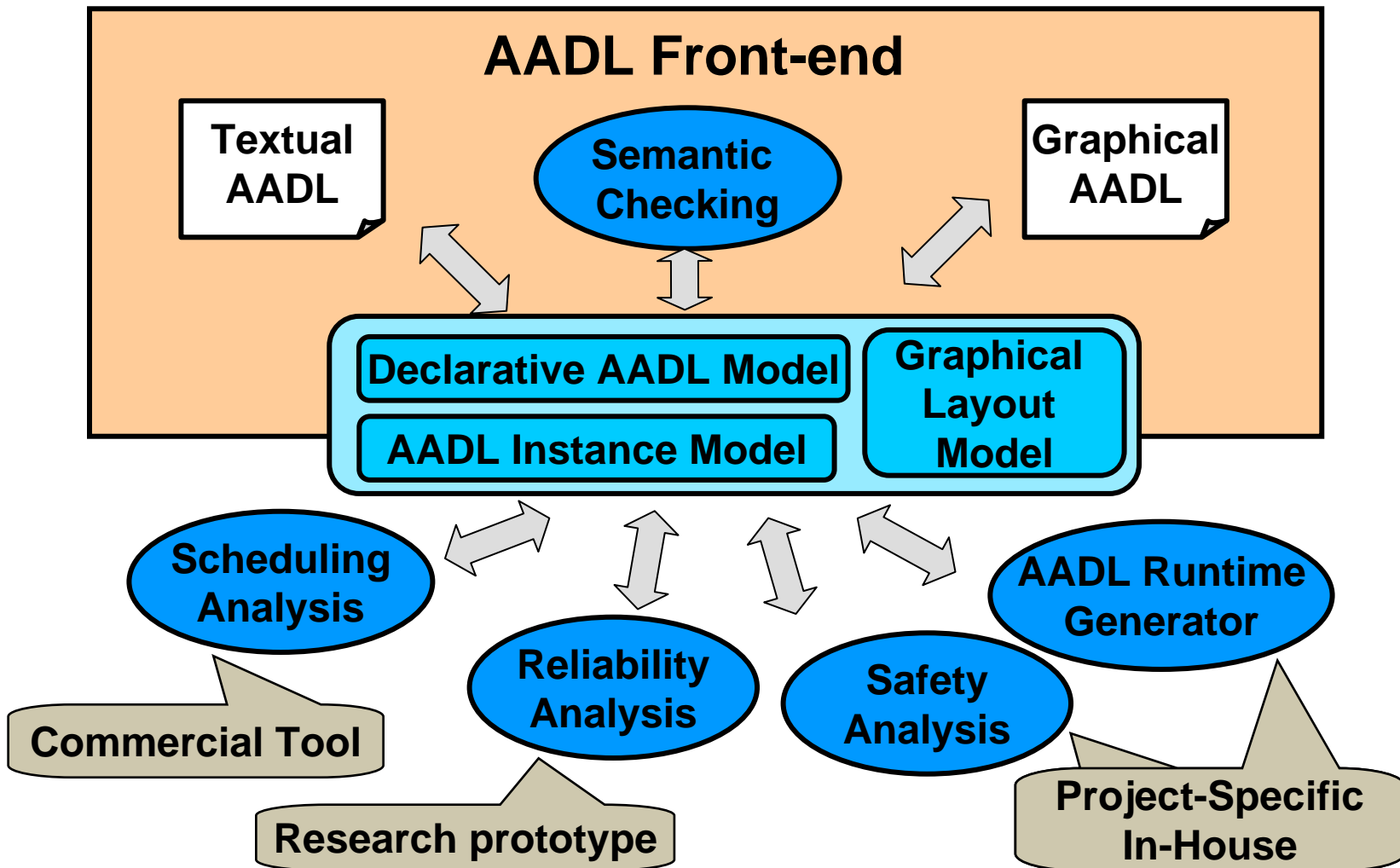


Two-Tier Tool Strategy

- Open Source AADL Tool Environment (OSATE)
 - Developed by SEI
 - Low entry cost solution (no cost CPL)
 - Multi-platform support based on Eclipse
 - Vehicle for in-house prototyping of project specific architecture analysis
 - Vehicle for architecture research with access to industrial models & industry exposure to research results
- Commercial Tool Support
 - UML tool environment extension based on UML profile
 - Extension to existing modeling environment with AADL export/import
 - Analysis tools interfacing via XML or XML to native filter
 - Runtime system generation tools



XML-Based Tool Integration Strategy





OSATE Capabilities



- OSATE Release 0.4.0 based on Eclipse Release 3
- Online AADL help
- Text to XML & XML to text
- Syntax-sensitive text editor
- Parsing & semantic checking of full AADL
- AADL property viewer
- Syntax-Sensitive AADL Object Model
- Model versioning & team support
- Model instantiation
- Model consistency checking
- AADL to MetaH translator
- Plug-in development

Over 250 downloads
internationally

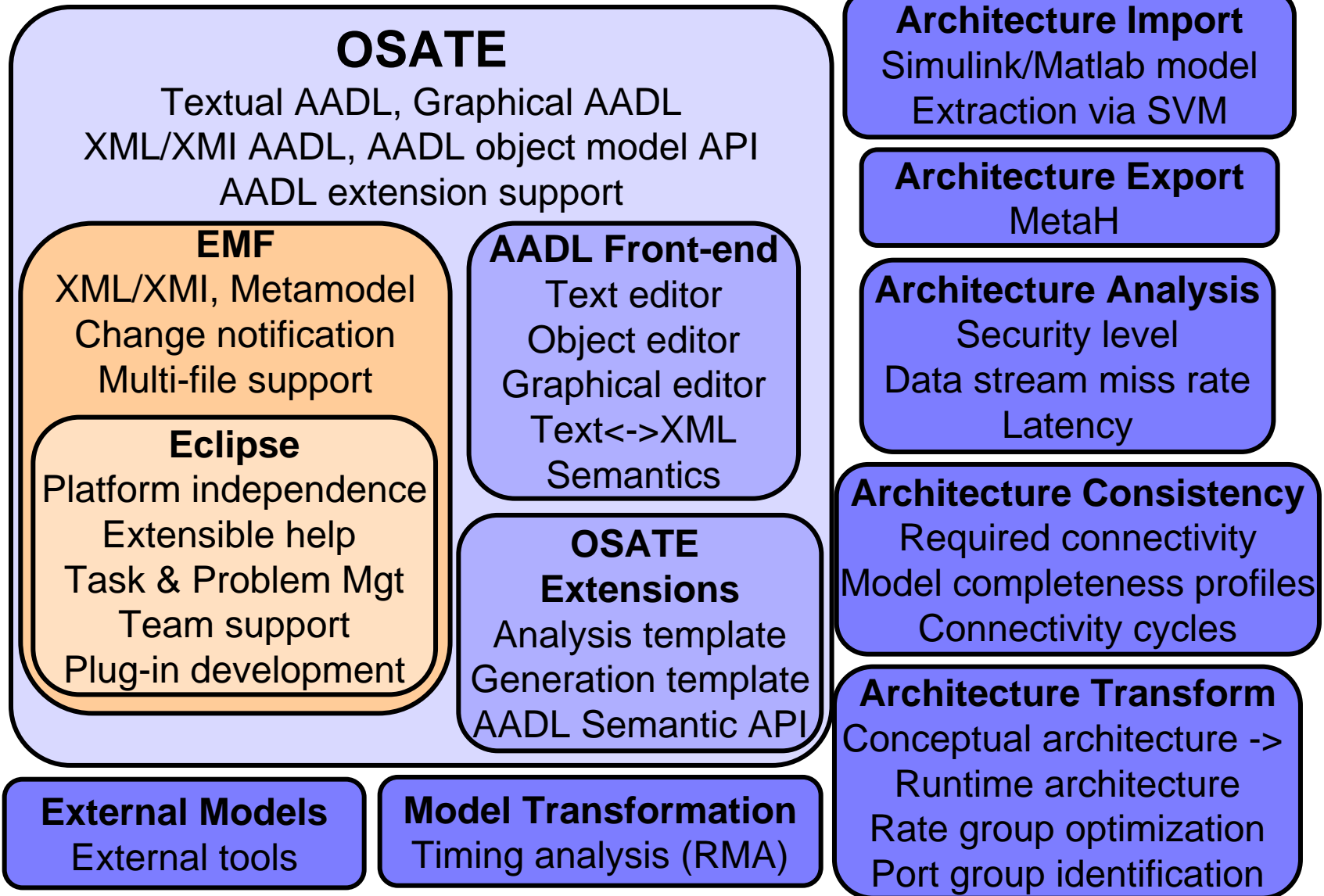
Processed 21000 line
AADL model in 20 sec

Next release Jan 2005
Graphical editor
Multi-file support





OSATE Plug-in Extensions





OSATE Plug-in Development

- Four part presentation series
 - Dec 2004 & Jan 2005
 - VTC, Webcast, telecon, video taped
 - Participants included
 - Airbus Industries, ENST, Axlog, TNI France
 - European Space Agency Netherlands
 - EADS Germany
 - US Army AMRDEC
 - Lockheed Martin, Rockwell Collins, Honeywell
 - USC, University of Pennsylvania
 - 21st Century Systems, Pyrrusoft
 - Bosch
- OSATE Plug-in Development Guide

SAE





Europe

www.tni-world.com

Stood 5

and AADL

Example of Commercial Tool Support

Pierre Dissaux, AADL meeting, Edinburgh, 12-15 July 2004

pierre.dissaux@tni-world.com



A Technology Transition Enabler

- Industry standard architecture modeling notation & model interchange format facilitates
 - Interchange of architecture models between contractors & subcontractors
 - Integration of architecture models for system of systems analysis
 - Common architecture model for non-functional system property analysis from different perspectives
 - Interoperability of modeling, analysis, and generation tools
 - Platform for research & prototyping of new architecture analysis techniques





Benefits

- Model-based system engineering benefits

Predictable runtime characteristics addressed early and throughout life cycle greatly reduces integration and maintenance effort

- Benefits of AADL as SAE standard

AADL as standard provides confidence in language stability, broad adoption, and strong tool support

SAE

