A few words about AADLv2: objectives and ecosystem

Jérôme Hugues, ISAE/DMIA

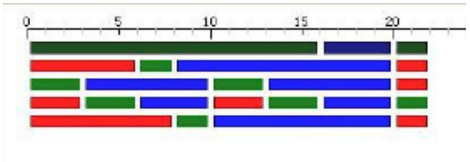
Credits for the materials go to
Bruce Lewis
US Army RDEC, SEI Affiliate

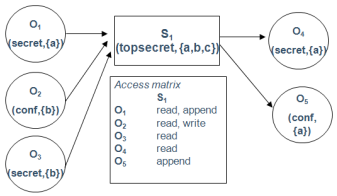
Peter H Feiler
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

AADL

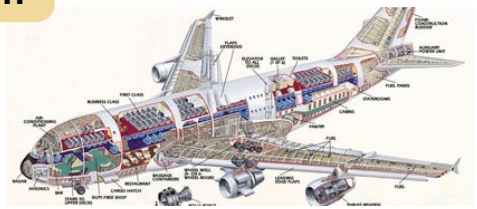
Potential Model-based Engineering Pitfalls

Inconsistency between independently developed analytical models



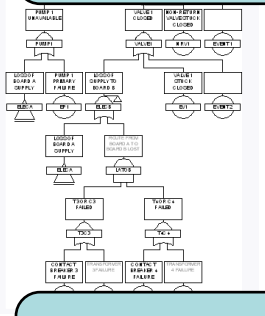


Confidence that model reflects implementation



The system

Architecture-centric model repository



System models

Generation from validated models

System implementation

System Level Fault Root Causes

AADL

Violation of data stream assumptions
- Stream miss rates, Mismatched data representation, Latency jitter & age

End-to-end latency analysis
Port connection consistency

Partitions as Isolation Regions
- Space, time, and bandwidth partitioning
- Isolation not guaranteed due to undocumented resource sharing
- fault containment, security levels, safety levels, distribution

Partitioned architecture models
Model compliance

Virtualization of time & resources
- Logical vs. physical redundancy
- Time stamping of data & asynchronous systems

Virtual processors & buses
Synchronization domains

Inconsistent System States & Interactions
- Modal systems with modal components
- Concurrency & redundancy management
- Application level interaction protocols

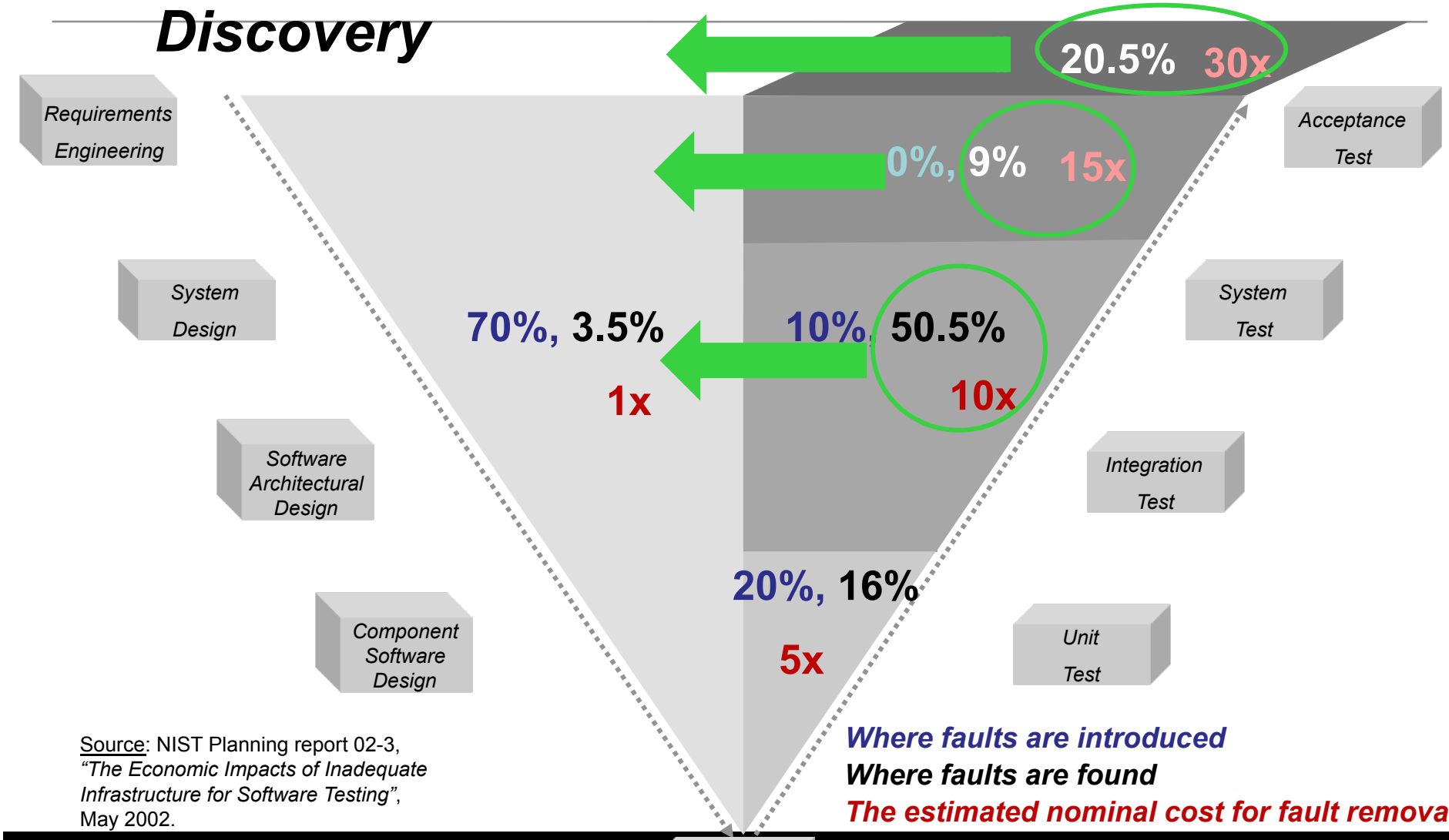
Fault propagation
Security analysis
Architectural redundancy patterns

Performance impedance mismatches
- Processor, memory & network resources
- Compositional & replacement performance mismatc
- Unmanaged computer system resources

Resource budget analysis & task roll-up analysis
Resource allocation & deployment configurations

AADL

Cost & Time Reduction due to Early Fault Discovery



Source: NIST Planning report 02-3, *"The Economic Impacts of Inadequate Infrastructure for Software Testing"*, May 2002.

Where faults are introduced

Where faults are found

The estimated nominal cost for fault removal

AADL: The Language

Designed for standardized incremental, composable, quantitative analysis and generative system integration

Precise semantics for components & interactions

- Thread, process, data, subprogram, system, processor, memory, bus, device, virtual processor, virtual bus, abstract
- Typed properties, properties with units and model reference values

Continuous control & event response processing

- Data and event flow, synchronous call/return, shared access
- End-to-End flow specifications, black box flow specs

Operational modes & fault tolerant configurations

- Modes & mode transition, mode specific properties & configurations

Modeling of large-scale systems

- Component variants, packaging of AADL models, public/private

Accommodation of diverse analysis needs

- Extension mechanism (property set, sublanguage) standardized

Key Elements of SAE AADL Standard

AADL

Core AADL language standard (SEI) Impact – tools/anal/integ

- Textual & graphical, precise semantics, extensible

AADL Meta model & XMI/XML standard (SEI) – Impact – analysis

UML profile for AADL – In process (Thales) – Complementary use

- Annex of OMG MARTE, guildelines for modeling AADL concepts

Error Model Annex (Honeywell) Update.

- Fault/reliability modeling, hazard analysis. V2 started.

Behavior Annex – Draft (Airbus) balloted. Partial to complete.

- Externally observable behavior of components

Programming Guidelines, Data Modeling Annexes – Draft (ENST)

ARINC 653 Annex – Draft (ENST) balloted. Accepted.

Modeling an Embedded System Architecture

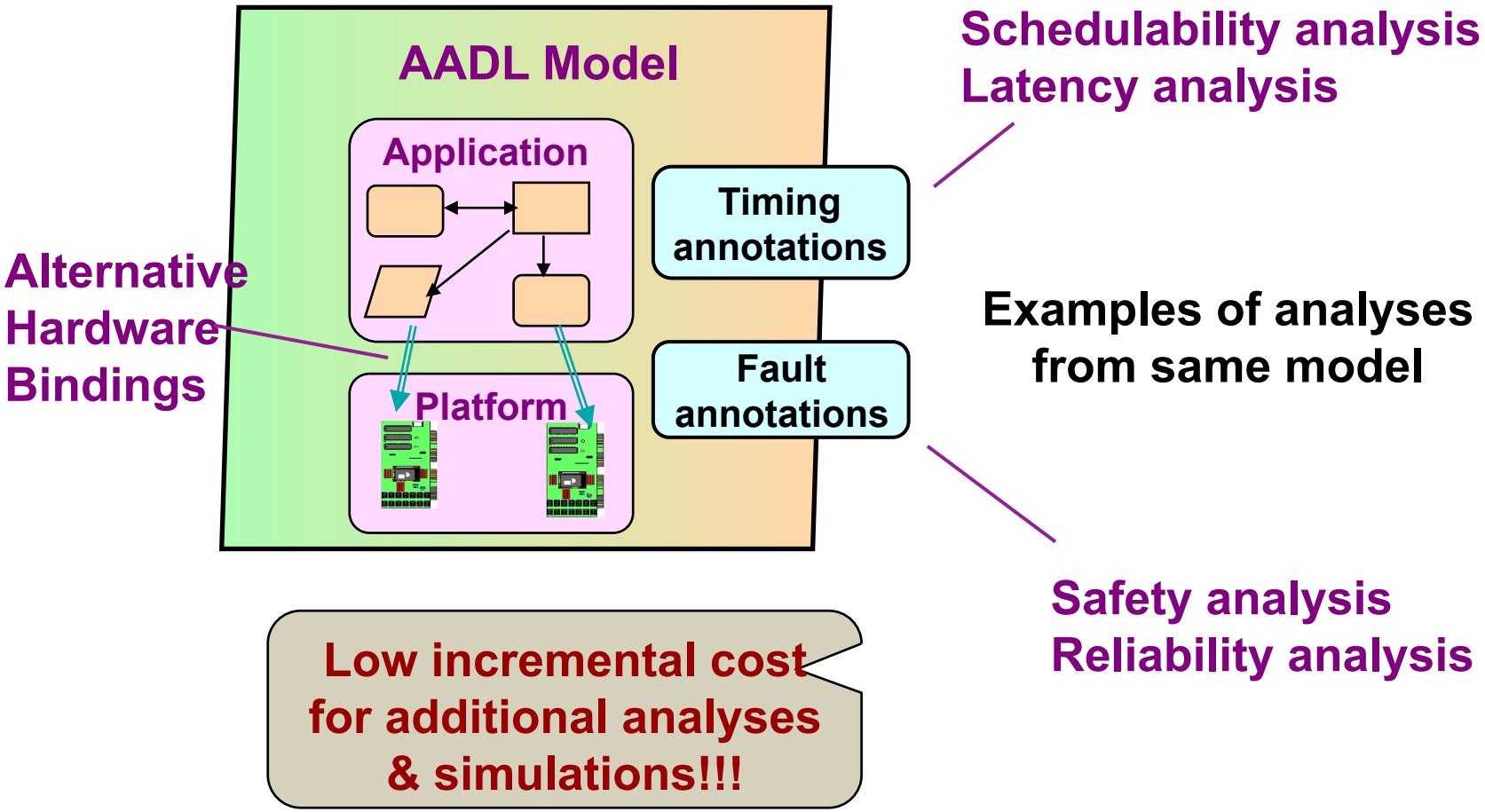
Elements of an embedded system architecture

- Application SW Architecture (task & communication) PLUS
- Computer platform architecture (processors & networks) PLUS
- Physical system/environment (interface with embedded SW/HW) PLUS
- Logical interface between software and physical system PLUS
- Physical interface between computer platform and physical system PLUS
- Deployment of software on computer platform

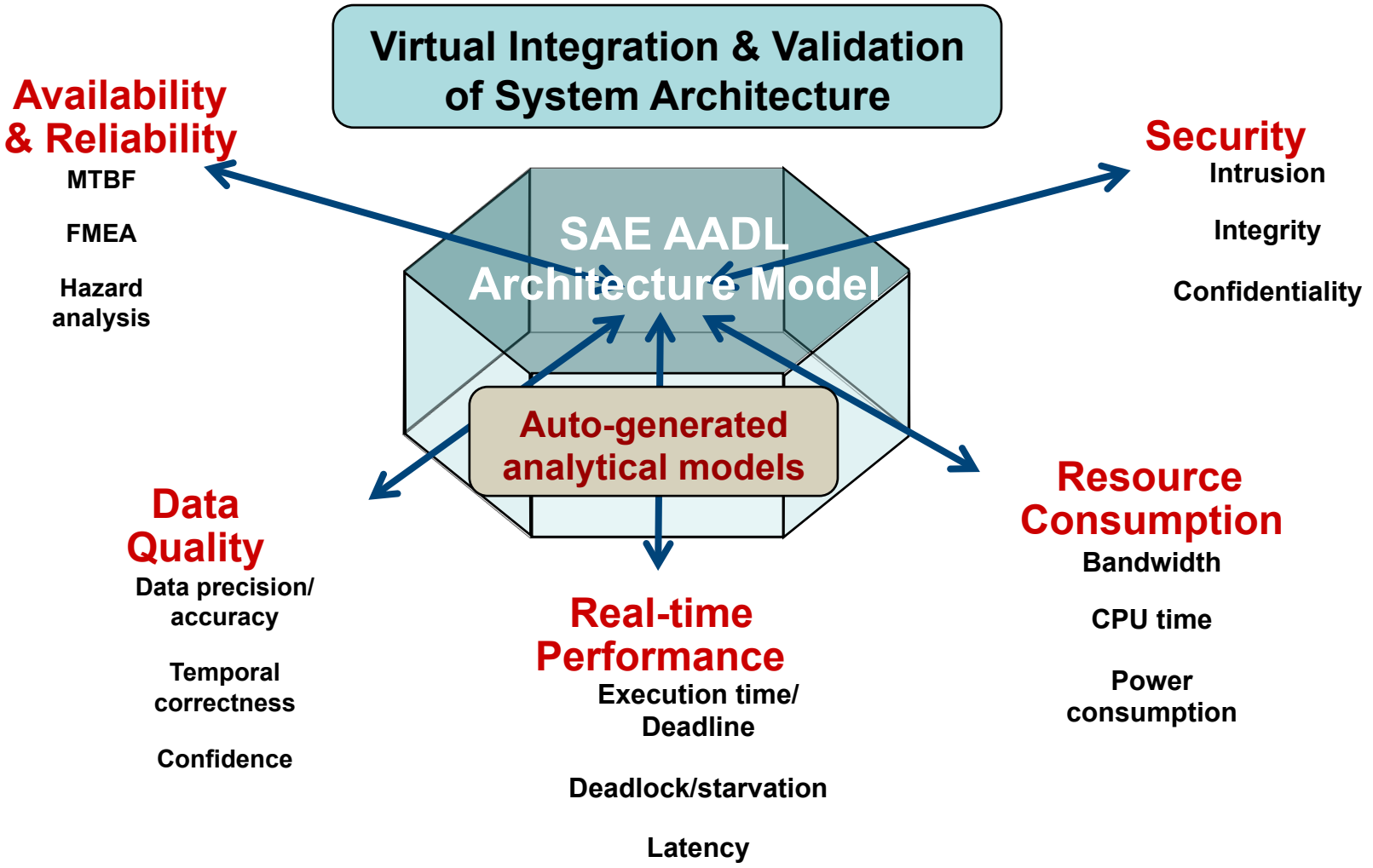
SAE AADL supports modeling, analysis, and auto-generation of embedded system architectures.

Single Source Architecture Model

AADL



Architecture-Centric Engineering Approach



Rapid Growth, Diversity of AADL Toolsets

AADL

OSATE – Open Source – Editor with analysis

- SEI developed, full language editing and semantic checking, multiple analysis plug-ins, Eclipse based, integrated text and graphical editing with TOPCASED. **New graphics editor being dev for V2**.

TOPCASED – Open Source – Model Bus Framework for integration of tools and methods

- Airbus led , 20 companies, Metamodeling Framework, AADL Graphics, AADL XML, model transformation, Behavior Annex, also will support UML, stable July 2007, includes new tools from SPICES.

STOOD – Commercial – Development support, Editor, Analysis

- CASE toolset supporting UML, HOOD and AADL. Includes transformations between notations, document support, requirements support. Works with OSATE, TOPCASED, OCARINA. Includes AADL simulator, Cheddar scheduling analysis. **New work will support MARTE to AADL and reverse.**

OCARINA – Open Source – Middleware generation and system integration

- ENST AADL graphics and middleware generation and integration to AADL model of tightly coupled or network distributed processors. Creates formal model of executive integrated in AADL. Generates to network protocols. **New ARINC 653 generator to AADL 653 Annex plus constraint lang for analysis.**

Fremont – Open Source, Formal analysis based tools, consulting and OSATE support

- AADL to ACRS (process algebra), formal analysis of concurrent resources, AADL to Charon, generation and integration of hybrid control systems, AADL Architecture Simulator

CHEDDAR – Open Source – Scheduling analysis

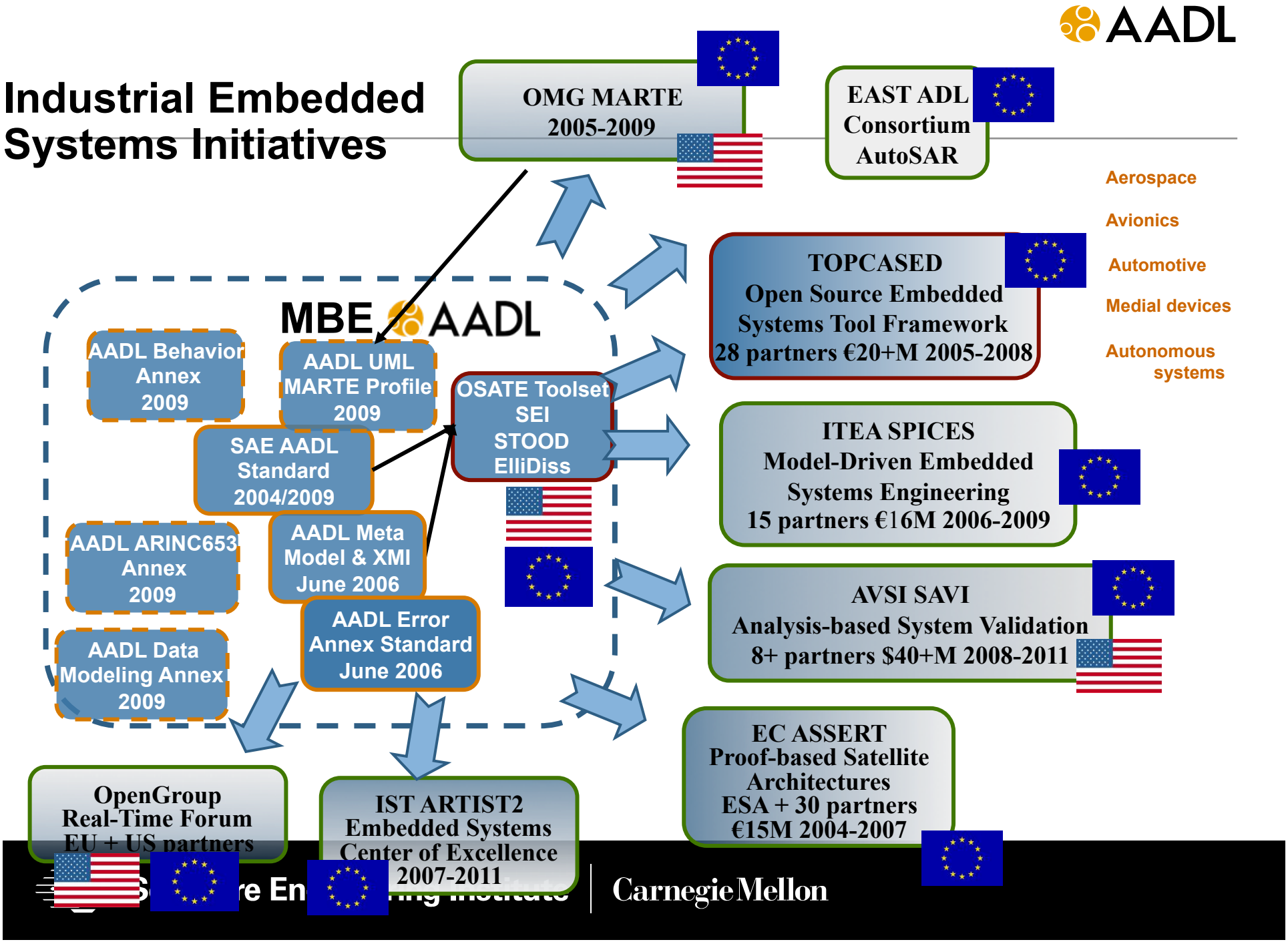
EDICT – Commercial – Fault Tolerant Systems and Security Analysis

WWTechnology – Error handling, Safety and Information Assurance modeling using AADL

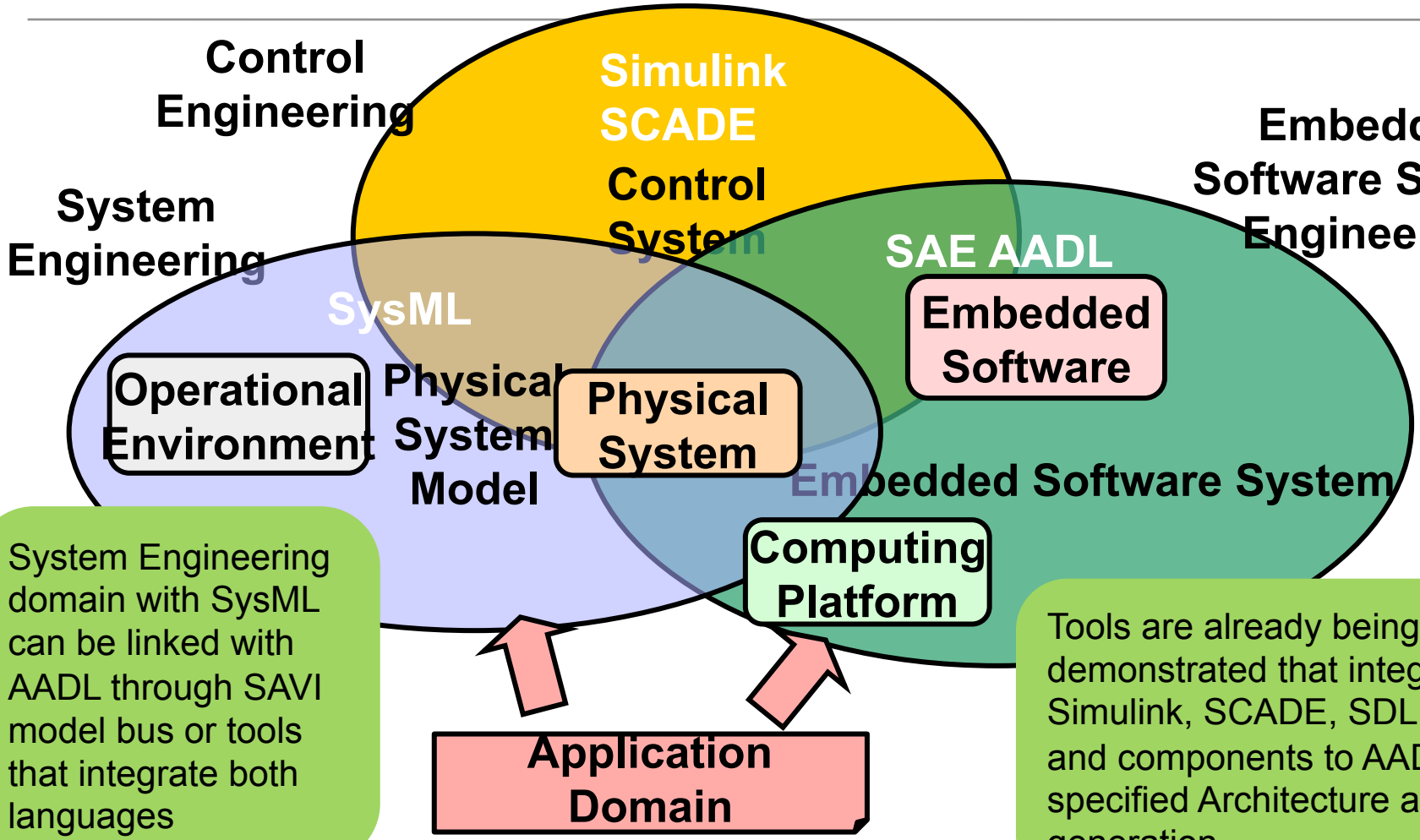
EMMESKAY – Commercial – Environment for control sys and architecture dev, AADL, Simulink, etc.

Consortium and Company Owned – SPICES, AVSI, ASSERT plus internal integrations

Industrial Embedded Systems Initiatives



Cooperative System, Control & Software Engineering



System Engineering domain with SysML can be linked with AADL through SAVI model bus or tools that integrate both languages

Tools are already being demonstrated that integrate Simulink, SCADE, SDL design and components to AADL specified Architecture and generation.