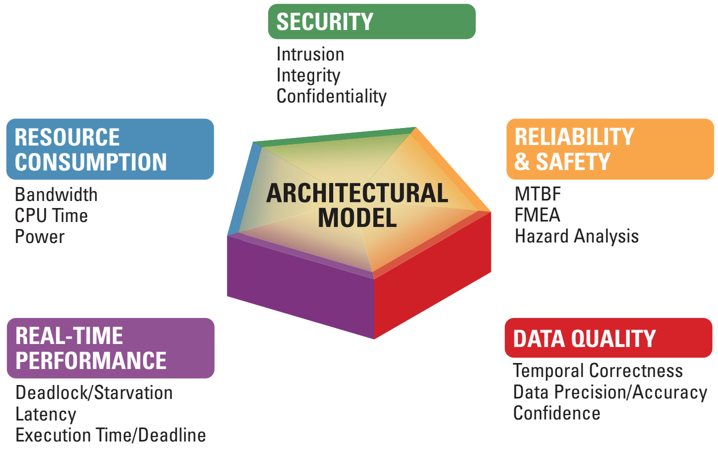
Software Engineering Institute | Carnegie Mellon

# Predictable, Model-Based Engineering for Embedded Systems

## What causes the problems noticed at system integration?

## What methods will give you greater confidence that your systems will perform as expected?

## Where can you reduce cost throughout the product life cycle?


SECURITY
Intrusion
Integrity
Confidentiality

RESOURCE CONSUMPTION
Bandwidth
CPU Time
Power

ARCHITECTURAL MODEL

RELIABILITY & SAFETY
MTBF
FMEA
Hazard Analysis

REAL-TIME PERFORMANCE
Deadlock/Starvation
Latency
Execution Time/Deadline

DATA QUALITY
Temporal Correctness
Data Precision/Accuracy
Confidence

*A single architectural model supports multiple analyses*

You've seen the results: high-risk integration, unacceptably expensive upgrades, and maintenance issues that lead to a shorter-than-anticipated life span for embedded and real-time systems.

Do you think you need a better way to assure the reliability and performance of your system, particularly when software is a key contributor?

Model-based engineering for embedded systems (MBE-ES) offers a way by focusing on the analysis of system architecture—providing the ability to detect problems with availability, security, and timeliness early on, before they conspire to raise costs, reduce effectiveness and predictability, and shorten lifespan.

### The Need for a Better Way to Predict Performance

Today, products that depend on embedded software systems come in any size—from automobiles, airplanes, and tanks to medical devices, household appliances, and MP3 players.

Most of these products are expected to do more and be more secure, and many are made to be portable or used in network environments. Software engineers are called on to deliver increasingly complex software systems that provide more functionality while consuming less power and costing less to develop and operate.

Unfortunately, system engineers do not have insight into critical system characteristics—such as

- performance
  (e.g., throughput or quality of service)
- safety
- reliability
- time criticality
- security
- fault tolerance

Using traditional means, system integration becomes high risk, and system evolution (life-cycle support) becomes expensive and results in rapidly outdated components.

### The Way Forward—Model-Based Engineering

Improved embedded systems engineering practice is architecture-based and model-driven. Well-defined software system architecture provides a framework to which system components are designed and integrated. System models that precisely capture this architecture provide the basis for predictable system engineering.

Model-based engineering for embedded systems

- reduces risk through early and repeated analysis of the system architecture
- permits the engineer to see system-wide impacts of architectural choices
- increases confidence by validating model assumptions in the operational system and permitting the system models to evolve in multiple fidelity
- reduces cost through fewer system integration problems and simplified life cycle support

### What Types of Systems Benefit from a Model-Based Approach?

- Embedded systems
- Real-time systems

### Where Can Model-Based Engineering Be Used?

- Avionics
- Automotive
- Aerospace
- Robotics
- Medical Devices
- Any software-intensive system with dependability, availability, performance, security, or safety-criticality requirements

# Predictable, Model-Based Engineering for Embedded Systems

The model-based engineering method for embedded systems that the Carnegie Mellon® Software Engineering Institute (SEI) uses is the Architecture Analysis & Design Language (AADL). This Society of Automotive Engineers (SAE) standard permits engineers to

- represent embedded systems as component-based system architecture
- model component interactions as flows, service calls, and shared access
- model task execution and communication with precise timing semantics
- model the binding of applications to execution platforms
- represent operational modes and fault tolerant configurations
- support component evolution and large-scale development
- accommodate reliability and safety analyses

## Benefits of the AADL

- Prediction and validation of runtime characteristics
- Validated system architectures and implementations
- Improved development process through a single annotated architecture model
- AADL is part of a model-based engineering enterprise solution
- AADL models simplify subcontractor management
- Interoperability and integration of commercial and in-house tools through an XML/XMI interchange format and a UML compatibility profile

## Course Provides MBE-ES Skills

After attending the SEI course *Modeling System Architectures Using the Architecture Analysis and Design Languages (AADL)*, engineers will understand how to use AADL to

- architect and engineer real-time and embedded software systems with predicable results
- analyze new and existing systems or new system architectures for potential systemic problems
- predictably integrate systems using validated architecture models
- perform model-based engineering early and throughout the system life cycle
- integrate an AADL-based tool environment

## Some Organizations Using the AADL

| Organization | Area(s) of Use |
|---|---|
| Airbus | System families |
| AVSI | Virtual integration approach |
| Axlog | System families |
| Honeywell | Performance, Safety/reliability, system families, tool environment investment |
| EADS | Reference architecture modeling |
| European Space Agency | System families |
| General Dynamics | Reference architecture modeling |
| Rockwell-Collins | Performance |
| LAAS | Safety/reliability |
| MIT | Classroom use |
| TNI | Tool environment investment |

## Related Websites

www.sei.cmu.edu/dependability/tools/aadl/
www.aadl.info

## For Course Registration

www.sei.cmu.edu/training/p72.cfm

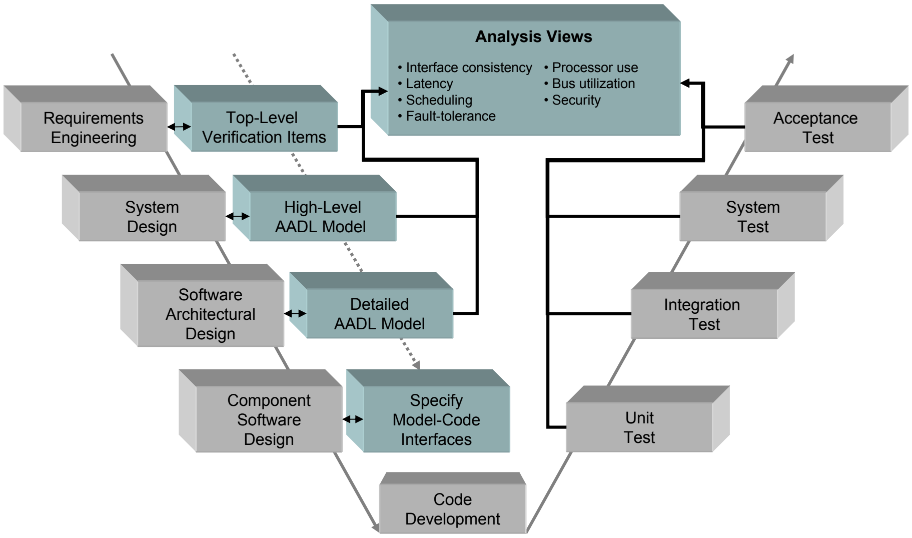
This course may also be offered by arrangement at customer sites. Email course-info@sei.cmu.edu or call +1 412-268-7622 for details.

## For More Information

Customer Relations
Phone: +1 412-268-5800
FAX: +1 412-268-6257
customer-relations@sei.cmu.edu

Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15313-2612

www.sei.cmu.edu



*AADL model development can be used in parallel with the software system's development. Then, the analysis views generated through AADL modeling can be compared to testing results during system implementation.*

01/08/2009

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.