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integration becomes high risk, and system evolution (life-cycle support) becomes expensive and results in rapidly outdated components.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
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 *MBE Essentials: An Introduction to Model-Based Engineering and AADL*, engineers will understand how to use AADL to

- architect and engineer real-time and embedded software systems with predictable 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

**Organizations Using the AADL**

<table>
<thead>
<tr>
<th>Organization</th>
<th>Area(s) of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airbus</td>
<td>System families</td>
</tr>
<tr>
<td>Axlog</td>
<td>System families</td>
</tr>
<tr>
<td>Honeywell</td>
<td>Performance, Safety/reliability, system families, tool environment investment</td>
</tr>
<tr>
<td>EADS</td>
<td>Reference architecture modeling</td>
</tr>
<tr>
<td>European Space Agency</td>
<td>System families</td>
</tr>
<tr>
<td>General Dynamics</td>
<td>Reference architecture modeling</td>
</tr>
<tr>
<td>Rockwell-Collins</td>
<td>Performance</td>
</tr>
<tr>
<td>LAAS</td>
<td>Safety/reliability</td>
</tr>
<tr>
<td>Embry-Riddle Univ.</td>
<td>Safety/reliability</td>
</tr>
<tr>
<td>MIT</td>
<td>Classroom use</td>
</tr>
<tr>
<td>TNI</td>
<td>Tool environment investment</td>
</tr>
</tbody>
</table>

**For More Information**

Visit our Web sites
http://www.sei.cmu.edu/dependability/tools/aadl/
http://www.aadl.info

To learn more, please contact
Robert Rosenstein
P: 412-268-8468
br@sei.cmu.edu
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-2612

For Course Registration
You can register for the course *Modeling System Architectures Using the Architecture Analysis and Design Language (AADL)*.
P: 412-268-7622
courseregistration@sei.cmu.edu
http://www.sei.cmu.edu/training/p72.cfm

Related SEI Publications