Software Engineering Institute | Carnegie Mellon

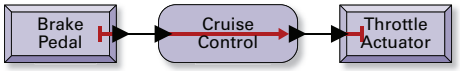
More Accurate Prediction of Real-Time Performance

Through Architecture Analysis using Model-Based Engineering Tools

Real-Time Challenges

Designers must satisfy stringent real-time (RT) performance requirements that call for increased hardware diversity and complexity of embedded software systems.

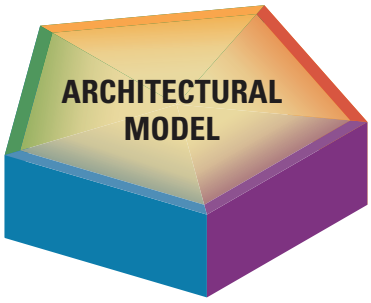
Consider how a cruise controller on an automobile maintains vehicle speed, over varying terrain, when it is engaged by the driver. When the brake is applied, the system must relinquish speed control until told to resume. The system must also steadily increase or decrease vehicle speed when directed.



Another example is the way an automated teller machine must respond quickly to a customer's instructions. Keeping processing time for a transaction to a minimum is important for customer satisfaction and as a differentiating product feature. And, knowing response time and delays helps to determine the transactions per second.

Or, how a stability control system schedules the sampling of data from sensors and the transmission of commands to the flaps of an airplane. If the data is not transmitted on time, the plane becomes unstable.

Model-based engineering (MBE) tools and techniques give designers of systems with RT performance requirements the analysis capabilities to handle greater complexity and increasingly rapid change in capability and technology.



REAL-TIME PERFORMANCE

Deadlock/Starvation
Latency
Execution Time/Deadline

The SEI uses model-based engineering tools, methods, and techniques to more accurately predict real-time performance.

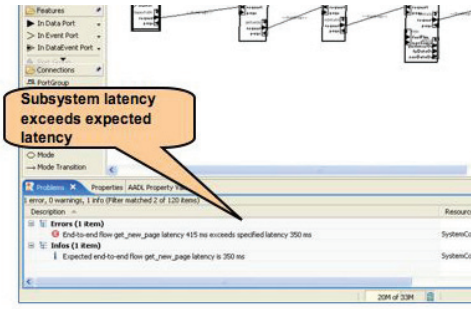
Analysis for End-to-End TIming and Latency

The SEI uses MBE tools to model and analyze an RT system's architecture in terms of software and hardware components and their interactions, system execution behavior, and quality attributes.

The SEI approach provides the information designers need to make choices that reduce worst-case latency and ensure the meeting of critical deadlines. Also, it supports the analysis needed to determine the end-to-end schedulability of the system as conceived.

Furthermore, the MBE tools and techniques that the SEI uses allow a designer to create a single system architectural model to

- prevent miscommunication
- support architecture tradeoffs
- cut risk and cost through early analysis, incremental development, and automated integration
- reduce time-to-market



MBE analysis identifies issues early and throughout the life cycle.

Modeling System Architectures Using the Architecture Analysis and Design Language (AADL)
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

Real-Time Analysis Concern	SEI MBE	Answer
Reduce worst-case flow latency and execution time	✔	Decision support for adding processors, using a faster processor, rewriting code, or lowering controller's signal processing rate
Assure schedulability	✔	Support for end-to-end timing and latency analysis
Fidelity of model to real system	✔	Capture of architecture and execution behavior in one model that represents the final system

Read more

Developing AADL Models for Control Systems: A Practitioner's Guide (CMU/SEI-2007-TR-014)

Flow Latency Analysis with the Architecture Analysis & Design Language (AADL) (CMU/SEI-2007-TN-010)

More Accurate Prediction of Real-Time Performance

Through Architecture Analysis using Model-Based Engineering Tools

The SEI MBE Toolkit

The SEI uses the ***Architecture Analysis & Design Language (AADL)*** to document system architecture and provide a platform for multiple analyses. An international industry standard, the AADL supports multiple analyses from a single architectural model, modeling and analysis throughout the life cycle, and analysis of runtime behavior (what) rather than functional behavior (how).

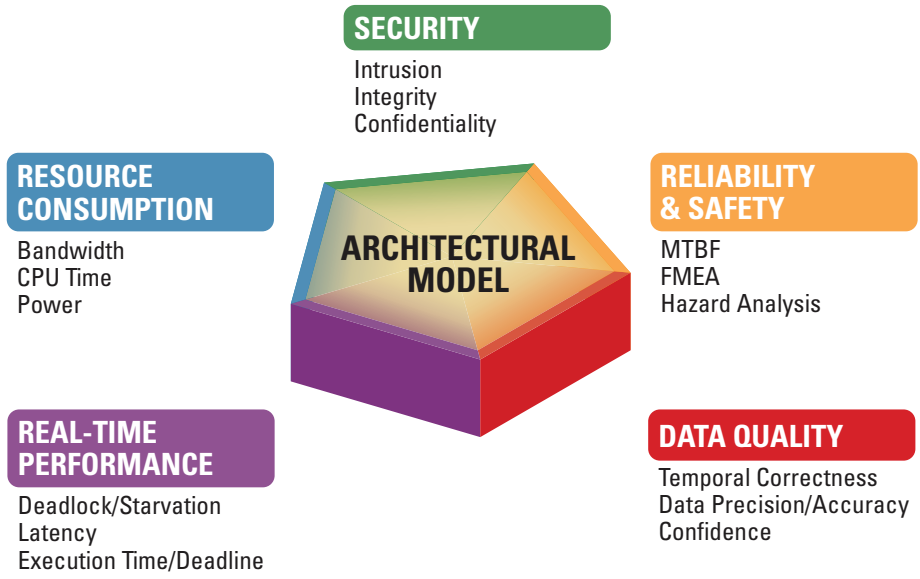
Through its ***XML/XMI interchange format,*** the AADL supports model interchange and tool chaining. And, the SEI offers the ***Open Source AADL Tool Environment (OSATE)*** flow latency analysis plug-in that validates the latency of flow implementations and compares it to the latency value for the flow specification. If the flow implementation latency value exceeds the value for the flow specification, an error is reported.

The SEI has developed OSATE as a set of plug-ins for processing of AADL models that includes:

- a syntax-sentitive text editor, with integrated error reporting
- a parser and semantic checker for textual AADL with conversion into AADL XML
- an unparser for AADL XML to textual AADL conversion
- support for multi-enterprise development through a version control system interface

AADL also can be used with

- UML state and process charts through its UML profile
- Architecture Tradeoff Analysis Method®, to drill into root causes and develop quantitative analysis assurance cases, to support claims made about the safety, security, or reliability of a system



Prevent System Integration Problems and Simplify Life-Cycle Support

Modeling of system quality attributes is often done—when it is done—with low-fidelity software models and disjointed architectural specifications by various engineers using their own specialized notations.

These models are typically not maintained or documented throughout the life cycle, making it difficult to predict the impact of change on attributes that cut across system functionality. The unanticipated effects of design approaches or changes are discovered only late in the life cycle, when they are much more expensive to resolve.

Analysis of a *system architecture model* offers a better way to predict the behavior of quality attributes. The SEI approach to model-based engineering (MBE) allows analysis

- using a single architecture model
- early and often in the development life cycle or on an existing system architecture
- at different architecture refinement levels
- along diverse architectural aspects such as behavior and throughput

Integration is a major cost and risk in complex systems. System understanding is a major cost driver during system maintenance. Proper use of MBE tools can prevent system integration problems and simplify life-cycle support.

System Architecture Modeling and Analysis

The Carnegie Mellon® Software Engineering Institute (SEI) provides technical assistance and guidance to transform the architectural design process from one based on human evaluation to one based on automated analysis.1

This analysis includes

- validating system quality attributes early in the design phase
- facilitating system integration
- conducting impact and tradeoff analysis using architecture models

For predicting and validating specific nonfunctional properties using model-based engineering, the SEI can help you to

- perform analysis that gives greater assurance that deployment will succeed
- evaluate fault tolerance of architectures
- adopt analytical resource models to validate performance behavior, power consumption, and network bandwidth usage
- model security aspects of architecture
- conduct analysis to guide localized architectural change
- validate data quality requirements such as temporal correctness, accuracy/precision, and confidence

Put MBE to work on your projects quickly!
Register for training by the Software Engineering Institute.
Go to www.sei.cmu.edu/training/p72.cfm.

1 One large defense contractor, for instance, blames human interpretation of the complexity involved with embedded systems for decreasing productivity to 6 or fewer lines of code per day.

0003-2010-03