Software Engineering Institute | Carnegie Mellon

More Accurate Prediction of System Reliability

Through Architecture Analysis using Model-Based Engineering Tools

Reliability Challenges

System designers know that software faults may cause failure in many forms, from loss of life and equipment to loss of specific functionality, which hinders mission effectiveness.

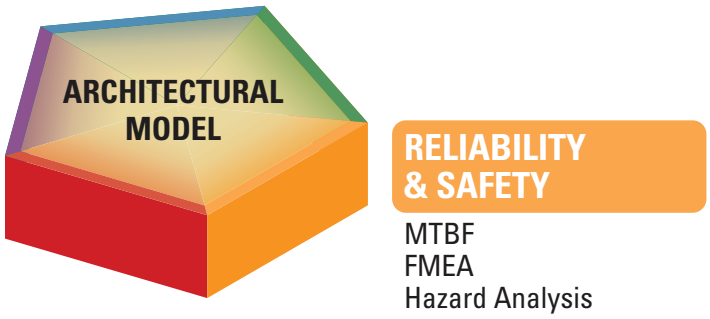
So, they want to ascertain and demonstrate a level of confidence in the system, expressed perhaps in an availability measure such as a mean time between failures (MTBF) number.

But can they predict how a system will perform when migrated to complex, new environments—such as an integrated modular avionics (IMA) architecture? In an IMA architecture, dedicated processors are replaced by virtual machines in a partitioned environment that can run applications concurrently.

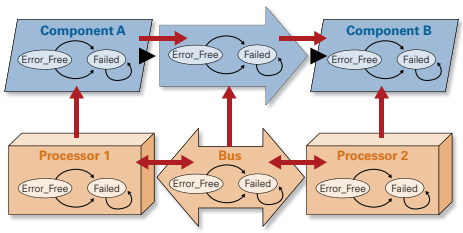
Or, can they gain insight into how choices made to improve reliability affect other critical qualities of the architecture, such as latency, CPU and bus utilization, and scheduling? Can they systematically evaluate each change prior to system integration and test?

Or, more commonly, do they know what will happen when a change to data format causes an overflow error, the kind of change that was not fully assessed before the first Ariane 5 rocket lifted off and blew up 40 seconds later?

Using model-based engineering (MBE) tools, methods, and techniques the SEI has developed an approach for dependency analysis that applies fault and repair models to system architecture. This approach promotes the early detection of potential problems for reliability (and other quality attributes), reduces system integration time and cost, and simplifies maintenance.



The SEI uses model-based engineering tools, methods, and techniques to more accurately predict system performance for availability and reliability.



Error models can be associated with application components, computing platform components, and device components, as well as the connections between them.

Error Modeling for More Accurate Reliability Prediction

The SEI approach using MBE permits error modeling on an architecture specification early and often throughout the development life cycle. These error models capture the results of failure modes and effects analysis (FMEA) and hazard analysis (HA).

This approach also supports the specification and analysis of fault trees, Markov models, and partition isolation. Error models can be associated with components, subsystems, systems, and even connections between components to describe, for example, fault behavior of data transfer.

Also, the SEI approach makes it possible to check for consistency, completeness, and traceability between the error models of interacting components as well as those of components and their subcomponents.

Reliability Analysis Concern	SEI MBE	Answer
Fault tolerance	✔	Error modeling captures FMEA results
Availability	✔	Error modeling supports fault tree analysis and MTBF
Hazard identification/risk analysis	✔	Designer can evaluate effects of partitioning and assess how other quality attributes might be affected (such as in end-to-end latency)

Read more

Dependability Modeling with the Architecture Analysis & Design Language (AADL) (CMU/SEI-2007-TN-043)

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

More Accurate Prediction of System Reliability

Through Architecture Analysis using Model-Based Engineering Tools

The SEI MBE Toolkit

The SEI uses the ***Architecture Analysis & Design Language (AADL)*** to document a system architecture and provide a platform for multiple analyses.

The AADL, an international industry standard, supports multiple analyses from a single architectural model, enables modeling and analysis throughout the life cycle, and provides analysis of runtime behavior (what) rather than functional behavior (how).

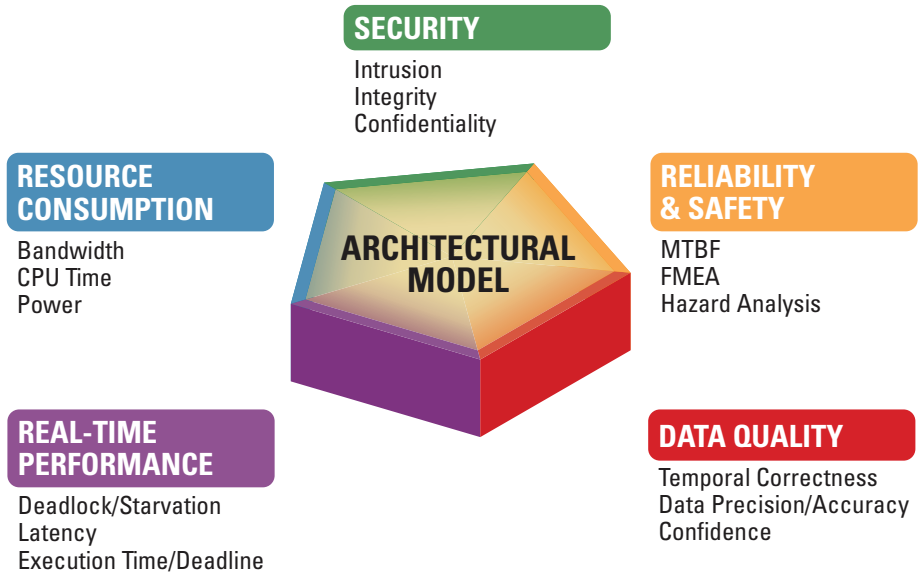
Through its ***XML/XMI interchange format,*** the AADL supports model interchange and tool chaining. The AADL error model annex supports mixed-fidelity modeling that makes it easier to modify architecture specifications and automatically regenerate reliability models at different levels of fidelity. The annex also enables improved traceability between architecture specifications and the generated models and analysis results.

And, the SEI offers the freely available ***Open Source AADL Tool Environment (OSATE)*** reliability plug-in that validates the latency of flow implementations. The SEI has developed OSATE as a set of plug-ins for processing 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

The AADL also can be used with

- UML state and process charts, through its UML profile
- the SEI 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.

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.

0003-2010-03