System Verification and Validation with Model-Based Engineering

The Architecture Analysis & Design Language

As our world becomes increasingly software-reliant, reports of safety and security issues in complex systems are also increasing. Traditional verification and validation approaches are inadequate for the scale, complexity, and ubiquity of today's software-intensive systems. Some safety faults and security breaches in these systems can be life-threatening. In 2015,

- Acura issued a recall to repair automated safety systems that incorrectly braked for a non-existent obstacle when traveling next to a guard rail
- after researchers used a software vulnerability to take control of a Jeep Cherokee over the internet, the manufacturer recalled 1.4 million cars to update the software

These incidents demonstrate the need to make software safer and more reliable.

Many software defects and vulnerabilities result from incomplete and ambiguous requirements. In addition, developers make assumptions—about timing, communication protocols, concurrency, data formats, ranges, and other variables—when developing components. When the system integrator assembles the system from the components, conflicting assumptions can lead to unanticipated problems in the component interactions, which result in system failures.

AADL Enables Early Analysis

Mismatches between software components and between software and hardware often result in system problems that are caught late in the development lifecycle, during integration or operation. Late discovery of faults and vulnerabilities is expensive for developers and potentially dangerous for users of mission- and safety-critical technologies. To address this problem, SAE International released Aerospace Standard AS5506, the Architecture Analysis & Design Language (AADL). Using AADL to model complex systems can help engineers identify issues that could cause system failures when components interact.

The AADL standard, which was authored by the SEI's Peter Feiler, defines a modeling notation based on a textual and graphical representation that development organizations can use to conduct lightweight, rigorous, yet comparatively inexpensive analyses of critical real-time factors such as performance, safety, security, and reliability. AADL models capture both software and hardware components as well as their interactions. AADL can be used early in the development cycle, enabling early analysis to determine whether a system will meet its requirements, before coding even begins.

Capabilities of AADL

By using models to capture the system architecture early and analyze its compliance with its requirements, stakeholders can expect to save development time while increasing the quality of produced system.

Software architects and system engineers can use AADL to

- specify real-time, embedded, and high-dependability systems with their software and hardware concerns and specific requirements
- perform multi-tier modeling and analysis of a system
- propagate changes to the model across multiple analysis dimensions
- maintain multiple model representations in a model repository
- auto-generate analytical models
- support interfacing to specialized analysis tools
- participate in distributed team development via a model repository
- perform analyses such as reliability and safety criticality through extensions

Upcoming Features for AADL

The SEI is developing a workbench that demonstrates measurable reduction in the cost of verifying system implementations against requirements. It can be used to assure systems incrementally throughout the development lifecycle by

- assuring hazard and derived requirement coverage during architecture design iterations
- reducing verification-related rework by detecting defects earlier in development
- providing a measure of confidence throughout the lifecycle by tracking requirement quality and verification results and auto-generating assurance-case artifacts
- extending the language to capture security aspects of the system and enable discovery of vulnerabilities

Benefits of AADL

Model-based engineering offers a better way to design, develop, analyze, and maintain system architecture. Using AADL-supported modeling and analysis, system architects and developers can

- reduce risk through early and repeated analysis of the system architecture
- reduce cost through fewer system integration problems and simplified lifecycle support
- assess system-wide impacts of architectural choices
- verify that the system will be built according to requirements and specifications
- validate systems and ensure that stakeholders' requirements can be achieved
- increase confidence because the assumptions made in modeling can be validated in the operational system

Additional Resources

Wiki and Source Code

With our user community, the SEI maintains a wiki with all of our research on AADL that is accessible to the public: **wiki.sei.cmu.edu/aadl**

For more information about AADL and OSATE, our AADL tool set, visit the following web sites:

- AADL information page: **aadl.info**
- OSATE GitHub: **github.com/osate**
- OSATE download: **osate.org**

Publications, Webinars, Podcasts, and More

Visit SEI's collection of AADL resources at **resources.sei.cmu.edu/library/asset-view.cfm?assetID=453645**

Course Registration

Learn to model system architectures using AADL: **sei.cmu.edu/training/p72.cfm**

About

For four decades, the Software Engineering Institute (SEI) has been helping government and industry organizations to acquire, develop, operate, and sustain software systems that are innovative, affordable, enduring, and trustworthy.

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412.268.5800 | 888.201.4479
Web: www.sei.cmu.edu | www.cert.org
Email: info@sei.cmu.edu