Software Engineering Institute | Carnegie Mellon

Research and Technology Highlight

Industry Standard Notation for Architecture-Centric Model-Based Engineering

January 20, 2010

DESCRIPTION

The SAE International Architecture Analysis & Design Language (AADL) [SAE 04/09] integrates concepts from research in software architecture into an international standard suite for modeling and analyzing the architecture of the operational software, the computer system, and the mission system of safety-critical, performance-critical, and mission-critical software-reliant systems in order to facilitate next generation industrial model-based embedded systems engineering practice. AADL's well-defined semantics that include specification of architecture dynamics in terms of modes and standardized mechanism for semantically consistent extensions provides the basis for automatically deriving analytical models to validate non-functional requirements (such as performance, safety, security and reliability) through formal analysis and simulation. In addition it supports auto-generation of application-specific runtime executives and rapid system construction from validated AADL models. The standardized AADL XMI interchange format facilitates model interchange between organizations, interfaces with existing and emerging analysis tools, and supports system validation through analysis of integrated subsystem AADL models. An SEI staff member has been the technical leader and author of the SAE AADL standard, under the sponsorship of the U.S. Army Aviation and Missile Research Development and Engineering Center (AMRDEC) Software Engineering Directorate (SED).

PROBLEM IT ADDRESSES

Safety-critical, performance-critical, and mission-critical systems have become increasingly software reliant. The cost of developing such systems has increased exponentially under the current practice of "build then test" and has become unaffordable—reaching $10B for the next generation aircraft, for example [AVSI 09]. Eighty percent of faults introduced during development are currently not caught until integration/acceptance testing and actual operation and repaired at a cost factors as high as 110.

Architectural models with well-defined semantics that include the interactions between the physical mission system, the computer system, and the embedded application software, support discovery of system-level problems through predictive model-based analysis of critical non-functional properties early in and throughout development to complement and refocus testing. Industrial proof-of-concept case studies have shown that this model-based engineering approach can dramatically reduce the cost and increase confidence in that expected system behavior will be met [AVSI 09].

CURRENT PRACTICE

Today's practice of developing software-reliant systems can be characterized as *build then test* with an increasing use of architecture modeling in the process. UML-based models are used to document the architecture and design of a system. Performance, safety, security, and reliability models are created independently with varying abstractions by different teams, resulting in analyses that are inconsistent with one another and with the actual system architecture [AVSI 09]. The resulting systems are "tested into submission" until testing budgets are exhausted. Despite best design practices and the use of fault tolerance techniques, most system-level faults are not discovered until late in the development process or even until operation. Typically, the cause of this pattern can be traced to mismatched assumptions between application system components and with the underlying runtime system [Feiler 09]. A number of recent studies have identified this problem and recommended a paradigm shift towards architecture-centric predictive analysis through increasing use of formal analytical frameworks. Some of these studies are the Leveson Study on the role of software in spacecraft accidents [Leveson 04], National Research Council Study on Certifiably Dependable Software Systems [NRC 07], the GAO Space-based Software Study [GAO 08], and the NASA Software Complexity Study [NASA 09].

APPROACH

The AADL was designed to represent the structure and dynamics of the runtime view of

- a system's software architecture
- the architecture of distributed computer system in terms of processors, memory, buses, and networks
- the physical mission system, in terms of the parts interfacing with the embedded software system as well as the deployment of the software on the computer system and its interaction with physical mission system

The AADL modeling notation is based on DARPA-funded software architecture research that occurred in the 1990s; its development has resulted in an industry standard architecture description language with well-defined semantics. This standardization of AADL concepts in a Meta model with execution semantics and through annexes that extend them in a semantically consistent manner accommodates multiple quality attribute dimensions in a single architecture-centric model repository. From this repository, analytical models are automatically generated, which remedies the problem of inconsistency between analytical models and the architecture. AADL models can be created through a standardized textual and graphical syntax as well as a UML profile for AADL; they can also be generated from architectural information in existing design databases and models. A standardized XMI interchange format facilitates interchanging AADL models created by different teams and interfacing with different toolsets. AADL models of subsystems can be integrated and analyzed early in development—enabling integration before implementation, which we refer to as *virtual integration*. System architecture models can be refined and analyzed at different levels of fidelity throughout the system development life cycle as has been demonstrated in industrial pilot projects (see ASSERT and AVSI SAVI in the Transition Section below).

SAE International as a standards forum attracted strong participation by U.S. and European aerospace and space industry, whose committee members became champions for first pilot projects as soon as the AADL standard was published in 2004. Feedback from these pilot projects resulted in a revision of the standard that was published in January 2009. As an industry standard, the SAE AADL has provided a stable technology platform in which a number of international industry initiatives have invested and on which the research community has developed prototyping and transition vehicles.

BENEFITS

An architecture-centric model-based engineering practice based on an industry standard architecture modeling notation allows whole industry sectors to jointly invest in a technology-intensive approach to improving the development of software-reliant systems. AADL supports this practice for software-reliant systems with safety, performance, and mission-criticality requirements. It allows system-level problems to be detected earlier in the life cycle and elusive time-sensitive, non-deterministic behaviors to be discovered through formal analytical techniques. The resulting practice will greatly reduce the cost of development, validation, and certification by increasing confidence in the safety, reliability, and performance of the system.

RESEARCH

In the 1990s, DARPA-funded research in software architecture fostered the creation of a number of architecture description languages (ADLs). MetaH was an ADL created for embedded systems by Steve Vestal at the Honeywell Technology Center [Vestal 94]. It supported modeling of a software task and communication architecture deployed on a computer system architecture specification and interfaced to a physical mission platform. It also introduced modes to represent dynamic reconfiguration of systems. A MetaH toolset supported architecture consistency checking, scheduling analysis, reliability analysis, and auto-generation of a runtime executive. Its successful use on a missile guidance system at U.S. Army AMRDEC SED led to a number of other pilot uses, as well as an investigation into its extensibility in which the SEI mapped MetaH to the ACME architecture interchange format developed by Garlan et al. under DARPA funding [ACME 98].

These developments led to the kickoff in 1999 of the SAE AS-2C Architecture Description Language Subcommittee of the Embedded Computing Systems Committee in the Avionics Systems Division.[1] The purpose of this group was the creation of the SAE AADL (which was originally named the Avionics Architecture Description Language). Bruce Lewis (AMRDEC SED) served as subcommittee chair, with Peter Feiler (SEI) as technical lead and author of the language. AADL incorporates concepts from MetaH and ACME, and the standard document includes a hybrid automata specification of the task execution semantics including initialization, finalization, reconfiguration dynamics of mode changes, and error recovery. The AADL standard (renamed Architecture Analysis & Description Language) was approved by 23 voting member organizations and published in November 2004. The approval and publishing of a set of annex standards followed in June 2006. These annexes include an error model extension to AADL to support various forms of hazard, reliability, and fault impact analysis. In January 2009, a revision of the standard was published; it incorporates concepts gained from industrial pilot projects and the application of formal analysis frameworks by the research community.

The AADL standards committee is currently defining annex standards that extend AADL in a semantically consistent manner. In particular, a Behavior Annex, a Data Modeling Annex, a Code Generation Annex, an ARINC653 Partitioned Architecture Annex, and a revision to the Error Model Annex are current-

[1] SAE International, once known as the Society of Automotive Engineers, is actually the largest provider of avionics systems standards, through its Avionics Systems Division. SAE member Elmer Sperry created the term automotive from the Greek autos (self) and the Latin motivus (of motion) to represent any form of self-powered vehicle.

ly in progress; all are expected to be published in the next 12 months. In this context, we are cooperating with a number of researchers and industrial users in the U.S. and Europe to strengthen the specification of these annex standards and validate them through mapping into formal analysis frameworks. Over 160 publications in refereed conferences and journals by over 50 research groups provide evidence that the AADL standard suite and the SEI-provided Open Source AADL Tool Environment (OSATE) have been embraced by the research community and used to focus their research on industrial engineering problems (e.g., University of Illinois [Mohan 09], University of Aachen [Noll 09]).

Two recent aspects of the SEI's research in improving architecture-centric engineering practice that build on model-based engineering with AADL are the development of a virtual upgrade validation method focusing on four root cause areas of system-level problems due to runtime architecture decisions sponsored by the Army Strategic Software Improvement Program (ASSIP), and the creation of a quantitative framework for system reliability validation and improvement sponsored by the US Army AMRDEC Aviation Engineering Directorate (AED).

TRANSITION

The transition strategy for an effective architecture modeling notation is being enacted through the AADL industry standard and a series of international industry initiatives to invest in and pilot this architecture-centric model-based engineering technology. To foster quick adoption, the SEI has provided a reference implementation of the OSATE (www.aadl.info) based on the Eclipse and the Eclipse Modeling Framework (www.eclipse.org). In Europe, an established commercial tool environment (STOOD by www.ellidiss.com) has been upgraded to include AADL support. With the emergence of the OMG MARTE profile for embedded systems (www.omgmarte.org), which has included an AADL profile in cooperation with the SAE AADL committee, commercial UML tool will support the creation of AADL models. The SAE AADL committee has also established a cooperative relationship with the OMG SysML working group (www.omgsysml.org) to align the two standards.

Industrial initiatives using AADL have been occurring since the standard was initially published. The first industrial initiative using AADL as a core technology was the ASSERT project led by the European Space Agency in cooperation with 29 partners; ASSERT ran from 2004-2007, and was funded at the level of 15M Euros. It developed and validated reference architectures for two satellite families and a tool chain for the model-based analysis and auto-generation of satellite systems from these reference architecture models (www.assert-project.net). TOPCASED followed in 2005 as a 4–5 year industry initiative of 28 partners led by Airbus to develop an industrial open source tool infrastructure for model-based engineering of embedded systems, with OSATE as part of the tool suite (www.topcased.org). In 2006, the three-year ITEA SPICES initiative of 15 research and industrial partners began to develop a model-based engineering method that incorporates modeling in CCM and AADL for analysis and auto-generation into SystemC.

In 2008, a consortium of aerospace companies—including Boeing, Lockheed Martin, and Airbus; several suppliers; as well as the FAA and the DoD—under the umbrella of the Aerospace Vehicle Systems Institute (AVSI) started a multi-phase System Architecture Virtual Integration (SAVI) initiative to establish a technology-intensive architecture-centric practice for performing system analysis early and throughout the life cycle. The SAVI approach is to use a model repository and model bus (for consistent information interchange) based on industry standards. For the proof-of-concept phase, AADL and OSATE were chosen as key technologies for a case study to analyze multiple quality attribute dimensions at several levels of fidelity on a multi-tier aircraft model and to illustrate the ability to support airframer (aerospace manu-

facturer)/supplier interactions through architecture model interchange via a model repository and model bus [AVSI 09]. This phase also included a return on investment (ROI) study of the effectiveness of this architecture-centric model-based engineering approach, led by Jőrgen Hansson of the SEI.

Based on the successful proof-of-concept, SAVI in November 2009 started its second of four phases. This second phase includes the involvement of commercial tool vendors to contribute to the tool infrastructure and increased integration of system engineering and embedded software system engineering methods.

In addition to SAVI, the SEI has applied architecture-centric model-based engineering based on AADL to a number of customer systems, including

- an early study of the CAAS migration from a federated to an integrated modular avionics (IMA) architecture
- the validation of the Mission Data System reference architecture for autonomous space vehicles by NASA/JPL
- a comparative study of six helicopter architectures
- the quantitative analysis of AADL models of the Apache Block Upgrade 3, in the context of an architecture evaluation using the SEI Architecture Tradeoff Analysis Method® (ATAM®) method

The SEI is also incorporating this architecture-centric model-based technology into a NASA–funded, model-based IV&V framework.

Acronyms

Acronym	Description	Acronym	Description
AADL	Architecture Analysis & Description Language	IV&V	Independent Validation & Verification
ADL	Architecture Description Language	JPL	Jet Propulsion Laboratory
AED	Aviation Engineering Directorate	MARTE	Modeling and Analysis of Real-time and Embedded systems
AMRDEC	Aviation and Missile Research Development and Engineering Center	NASA	National Aeronautics and Space Administration
ASSERT	Automated proof-based System and Software Engineering for Real-Time applications	OMG	Object Management Group
ASSIP	Army Strategic Software Improvement Program	OSATE	Open Source AADL Tool Environment
ATAM	Architecture Tradeoff Analysis Method	SAVI	System Architecture Virtual Integration
AVSI	Aerospace Vehicle Systems Institute	SED	Software Engineering Directorate
CAAS	Common Aviation Architecture Systems	SPICES	Support for Predictable Integration of mission Critical Embedded Systems
CCM	CORBA Component Model	TOPCASED	Toolkit in OPen source for Critical Applications and SystEm Development
DARPA	Defense Research Advanced Projects Agency	UML	Unified Modeling Language
GAO	Government Accountability Office	XMI	XML Metadata Interchange
IMA	Integrated Modular Avionics		

® Architecture Tradeoff Analysis Method and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

REFERENCES

[AVSI 09] Feiler P. H., Hansson J., de Niz D., & Wrage L. *System Architecture Virtual Integration: An Industrial Case Study* (CMU/SEI-2009-TR-017). Software Engineering Institute, Carnegie Mellon University, 2009. http://www.sei.cmu.edu/library/abstracts/reports/09tr017.cfm

[Feiler 09] Feiler, Peter H. "Challenges in Validating Safety-Critical Embedded Systems." *Proceedings of SAE International AeroTech Congress*. Seattle, WA (USA), November 2009. https://www.sae.org/technical/papers/2009-01-3284

[Garlan 97] Garlan, David, Monroe, Robert T., & Wile, David. "Acme: An Architecture Description Interchange Language," 169-183. *Proceedings of the 1997 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON'97).* Toronto, Ontario, Canada, November 1997. IBM Press, 1997.

[GAO 08] General Accounting Office Report to Congressional Committees. *DOD's Goals for Resolving Space-Based Infrared System Software Problems Are Ambitious* (GAO-08-1073). September 2008.

[Leveson 04] Leveson, Nancy. "*The Role of Software in Spacecraft Accidents." AIAA Journal of Spacecraft and Rockets 41*, 4, July 2004.

[NASA 09] Dvorak, Daniel L. *NASA Study on Flight Software Complexity* (NASA/CR-2005-213912). NASA Office of Chief Engineer Technical Excellence Program, March 2009.

[Noll 09] Marco Bozzano, Alessandro Cimatti, Marco Roveri, Joost-Pieter Katoen, Viet Yen Nguyen and Thomas Noll. *Codesign of Dependable Systems: A Component-Based Approach*, Seventh ACM-IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE'2009).

[NRC 07] Jackson, Daniel, Ed. *Software for Dependable Systems: Sufficient Evidence?* Committee on Certifiably Dependable Software Systems, National Research Council. National Academic Press, 2007. ISBN: 0-309-10857-8

[SAE 04/09] SAE International. *Architecture Analysis and Design Language v2.0 (AS5506A)*. January 2009.

[Sha 09] Mohan, Sibin, Nam, Min-Young, Pellizoni, Rodolfo, Sha, Lui, Bradford, Richard, & Fliginger, Shana. "Rapid Early-Phase Virtual Integration," 33-44. *Proceedings of 30th IEEE Real-time Systems Symposium*. Washington, DC (USA), December 2009. IEEE Computer Society, 2009.

[Vestal 93] Vestal, S. & Binns. P. "Scheduling and communication in MetaH," 194-200. *Proceedings of the 14th Annual Real-Time Systems Symposium*. December, 1993. IEEE Computer, 1993.