Model Based Computer System Engineering with the SAE Architecture Analysis & Design Language (AADL)

Bruce Lewis
Chair, SAE AS-2C Architecture Design Language Subcommittee
Army AMCOM Software Engineering Directorate

bruce.a.lewis@us.army.mil
256-876-3224
Building and Modifying (Engineering) RT Computer Based Systems

• How predictable is the success of your hardware/software integration?

• How much rework is built-in cost in your programs to avoid failures to integrate?

• What will it cost (impact) to upgrade your architecture to handle new software, new hardware?

• What percent of programs fail due to integration issues?
Engineering of Embedded Systems

- Embedded systems are defined as the integrated computer hardware + software in its system context.

- System integration is the largest single technical cause of program failure. Embedded systems are very complex... We need help...
  - Major issues
    - S.E. requires predictive analysis of the integrated whole
    - Impact of change must be understood in each dimension of behavior

- Best system engineering will dominate complex systems – less rework, more automation => cheaper faster better
Embedded System Engineering Requires Hardware + Software + System Context Analysis

• Issues
  – Engineering methods are analytical, require precise definition
  – Embedded S. E. largely missing in Software Engineering practice

• Solution
  – AADL provides an international standard with precise definition
  – Integrates embedded software and system engineering (Peter)
  – Is a stepping stone to new levels of embedded system engineering and validation (examples later)
  – Demonstrated on large systems (examples later)
Many Cross Cutting Dimensions

• Cross Cutting => change one aspect, others impacted.

• Issues
  – Timing, utilization of resources, ordering/phasing, safety, security
  – Architecture failure => failure of the system or mission

• Solution
  – Integrated Architecture Modeling & Analysis Based process.
  – AADL - Single language to capture & analyze multiple dimensions
  – Built to support incremental development, automated integration
  – Complements SW engineering
SAE AADL Standard
An Enabler of Predictable Model-Based Embedded System Engineering

- AADL - A precise analyzable architecture description language for component based specification of Real-time, Embedded, Fault-tolerant, Secure, Safety-critical, Software-intensive systems (HW & SW) and for automated component based integration.
- Fields of application: Avionics, Automotive, Aerospace, Autonomous systems, …
- Based on 12 Years of DARPA funded technologies
- 5 years development for core Standard
- Core Standard (AS5506) published Nov 2004
- www.aadl.info
Where AADL Fits in the Development and System Evolution Process
Model-Based System Engineering

Architecture Modeling & Analysis

Requirements Analysis

System Integration

Rapid Integration
Predictable Operation
Upgradeability
Reduced Cost

Predictive Analysis Early In & Throughout Life Cycle
Model-Based Embedded System Engineering

Document the Architecture
Abstract, but Precise

AADL Runtime System
- Application Software Integration

System Analysis
- Schedulability
- Performance
- Reliability
- Fault Tolerance
- Dynamic Configurability

System Construction
- AADL Runtime System
- Application Software Integration

 execution Platform

Application Software

Execution Platform

- GPS
- DB
- HTTPS
- Ada Runtime

- Devices
- Memory
- Bus
- Processor
Analysis of Operational Characteristics

Useful from many current approaches

- UML
- xUML
- SysML
- Existing Architecture
- Simulink
- Domain Component generators
- Product Line
- Reference Architecture
- Arch Tradeoff Analysis M
- AADL Model of Runtime Architecture
  - Error Model Annex
  - Behavior Annex
  - FPGA Annex
  - ARINC653 Annex
- Analysis based on precise semantics
NATO ALWI-CL xUML AADL Reference Architecture to Operational Architecture Process

1. Requirements
2. Build Platform Independent xUML API Model
3. Automatically Generate Target Code
4. Automatically Generate AADL From Spreadsheet
5. Refine xUML API Model
6. F16 Aircraft Platform Specification
7. E2
8. JSF
9. PIM or PDM Problems ?
10. Analyse AADL Specification
11. Identify Problems in AADL
12. AADL Analysis Results
13. Automatically Generate Target System
14. Target System

Provided by Chris Ramstrick
KENNEDY CARTER
Modified by Bruce Lewis
AADL Standardization Process
MetaH: Proof of Concepts for AADL

1991 DARPA DSSA program begins
1992 Partitioned PFP target (Tartan MAR/i960MC)
1994 Multi-processor target (VME i960MC)
1995 Slack stealing scheduler
1998 Portable Ada 95 and POSIX middleware configurations
1998 Extensibility through MetaH-ACME Mapping
1998 Reliability modeling extension
1999 Hybrid automata verification of core middleware modules

Numerous evaluation and demonstration projects, e.g.
- Missile G&C reference architecture, demos, others (AMCOM SED)
- Hybrid automata formal verification (AFOSR, Honeywell)
- Missile defense (Boeing)
- Fighter guidance SW fault tolerance (DARPA, CMU, Lockheed-Martin)
- Incremental Upgrade of Legacy Systems (AFRL, Boeing, Honeywell)
- Comanche study (AMCOM, Comanche PO, Boeing, Honeywell)
- Tactical Mobile Robotics (DARPA, Honeywell, Georgia Tech)
- Advanced Intercept Technology CWE (BMDO, MaxTech)
- Adaptive Computer Systems (DARPA, Honeywell)
- Avionics System Performance Management (AFRL, Honeywell)
- Ada Software Integrated Development/Verification (AFRL, Honeywell)
- FMS reference architecture (Honeywell)
- JSF vehicle control (Honeywell)
- IFMU reengineering (Honeywell)
From Research to Standard

Research ADLs

• MetaH
  – Real-time, modal, system family
  – Analysis & generation
  – RMA based scheduling

• Rapide, Wright, ..
  – Behavioral validation

• ADL Interchange
  – ACME

Industrial Strength

• UML 2.0, UML-RT
• HOOD/STOOD
• SDL

DARPA Funded Research since 1990

TNI, Airbus & ESA

Extensible
Real-time
Dependable

www.aadl.info
Industry Drives AADL Standard

• Bruce Lewis (US Army AMRDEC): Chair
• Peter Feiler (SEI): technical lead, std author & editor
• Steve Vestal (Honeywell): std co-author, Error Annex
• Ed Colbert (USC): AADL UML Profile Annex
• Joyce Tokar (Pyrrhus Software): Ada & C Annex
• Mamoun Filali, P. Dissaux, P. Gaufillet (Airbus): Behavior Annex

Other Voting Members -//- Contributors

• Smith Industries, Rockwell, Honeywell, Lockheed Martin, General Dynamics, Airbus, Axlog, European Space Agency, Ellidiss, Dassault, EADS, High Integrity Systems, Ford, Toyota, Eaton, UPenn, Draper Labs, ENST -//- Boeing, Raytheon

Coordination with

• Open Systems Joint Task Force (OSJTF), NATO Aviation Systems, French COTRE, EU ASSERT, TOPCASED, SPICES, SAE & NATO & AF Weapons Plug and Play, OMG UML, MARTE
Key Elements of SAE AADL Standard

- Core AADL language standard
  - Textual, semantics
- Graphical AADL notation annex
  - Enables graphical AADL programming
- AADL Meta model & XMI/XML standard
  - Model interchange & tool interoperability
- Programming Language API Annex
  - Mapping to Ada, C/C++
- Error Modeling Annex
  - Reliability and fault modeling
- UML profile for AADL (Nov 2006)
  - Transition & Integration for UML practitioner community
- Behavior Annex (April 2007)
  - Detailed component behavior modeling

Published Nov 2004
Published July 2006
Upcoming ballots
AADL Language Overview
AADL: Standard Components and Interactions

Components with precise semantics
- SW - Data, subprogram, thread, thread group, process, system,
- HW - Processor, device, memory, bus, system
- System of systems

Completely defined interfaces & interactions
- Data & event flow, synchronous call/return, shared access
- End-to-End flow specifications

Real-time Task Scheduling
- Supports different scheduling protocols incl. GRMA, EDF
- Defines scheduling properties and execution semantics

Modal, configurable systems
- Modes to model transition between statically known states & configurations

Component evolution & large scale development support
Public and Private (proprietary) packaging
AADL language extensibility
system implementation GPS.secure

subcomponents
  decoder: system PGP_decoder.basic;
  encoder: system PGP_encoder.basic;
  receiver: system GPS_receiver.basic;

connections
  c1: data port speed_data -> decoder.in;
  c2: data port decoder.out -> receiver.in;
  c3: data port receiver.out -> encoder.in;
  c4: data port encoder.out -> s_control_data;

flows
  speed_control: flow path speed_data -> c1 -> decoder.fs1
                -> c2 -> receiver.fs1 -> c3 -> decoder.fs1
                -> c4 -> s_control_data;

modes none;

properties arch::redundancy_scheme => Primary_Backup;
end GPS;
Graphical Flight Manager in AADL
AADL Tools – Strategy and What’s Available
Two-Tier Tool Strategy

• Open Source AADL Tool Environment (OSATE)
  – Low entry cost (free), full language baseline
  – Multi-platform, tool integration, interface based on Eclipse
  – Vehicle for rapid development of specific architecture analysis, research extensions, XML search routines provided
  – Includes multiple analysis plug-ins, TOPCASED integration

• Commercial Tool Support
  – HOOD/STOOD: Extension to existing avionics modeling environment with AADL export/import (Ellidiss)
  – Analysis tools interfacing via XML (Airbus, Rockwell, Honeywell, Fredmont Associates, ASSERT)
  – UML tool environment extension based on AADL UML profile (check Artisan, Rational, ILogix, Kennedy Carter)
Standard XML and Eclipse Based Tool Integration

AADL Eclipse Front-end

Textual AADL

Semantic Checking

Graphical AADL

Declarative AADL Model

AADL Instance Model

Graphical Layout Model

Scheduling Analysis

Reliability Analysis

Safety Analysis

AADL Runtime Generator

Commercial Tool

Research prototype Eclipse Plug-in

Project-Specific In-House Tools

www.aadl.info
Rapid Growth, Diversity of AADL Toolsets

- **OSATE – Open Source**
  - SEI developed, full language editing and semantic checking, multiple analysis plug-ins, Eclipse based, integrated text and graphical editing with TOPCASED

- **TOPCASED – Open Source**
  - Airbus led, 20 companies, Metamodelling Framework, AADL Graphics, AADL XML, model transformation, Behavior Annex, also will support UML, stable July 2007

- **STOOD - Commercial**
  - CASE toolset supporting UML, HOOD and AADL. Includes transformations between notations, document support. Works with OSATE, TOPCASED, and Cheddar

- **OCARINA – Open Source**
  - ENST AADL graphics and middleware generation and integration to AADL model of network distributed processors. Creates formal model of executive integrated in AADL. Generates to network protocols – CORBA, RT, FT

- **Fremont – Open Source, Consulting and Toolset support**
  - AADL to ACRS (process algebra), formal analysis of concurrent resources.
  - AADL to Charon, generation and integration of hybrid control systems.
  - AADL Architecture Simulator – integrates event driven and schedule driven

- **Generic Modeling Environment (GME) – Consortium**
  - Vanderbilt Univ, DARPA sponsored Metamodelling Framework, AADL capture and role based system security analysis, model transformation, integration.

- **CHEDDAR – Open Source**
  - Univ of Brest, advanced scheduling analysis toolset

- **Consortium and Company Owned – typically integrated analysis, generative**
AADL In Use

Architecture Specification and Automated Timing and Safety Analysis for a Large Avionics System

Steve Vestal
Larry Stockler
Dennis Foo Kune
Pam Binns
Nitin Lamba

COTRE as an AADL Tool

- Funded by the French research department (CNRS) from 2002 to 2004
- Goal: Real Time architecture verification (main behavioral point of view)
- Exploration project aiming to develop a demo prototype
- Partners: AIRBUS, TNI, IRIT, LAAS, ONERA, CERT

Analyzable and Reconfigurable AADL Specifications for IMA System Integration

David Statezni
Advanced Technology Center
Rockwell Collins, Inc.

www.aadl.info
AMCOM Missile Demo & Retarget

Total project savings 50%, re-target savings 90%

- First integration of reengineered system
- Retargeting to new execution platform
- Using MetaH (subset of AADL)
- Reengineering & model analysis

www.aadl.info
Honeywell MetaH/AADL Large System Evaluations

Air transport aircraft IMA (simplified production workload)
  Globally time-triggered
  6 processors, 1 multi-drop bus
  105 threads, 51 message sources

Military helicopter (first release, partial)
  Globally time-triggered
  14 dual processors, 14 bus bridges, 2 multi-drop buses
  306 threads, 979 [source, destination] connections

Air transport aircraft IMA (preliminary, partial)
  Globally asynchronous processors, precedence-constrained switched network
  26 processors, 12 switches
  1402 threads, 2644 [source, destination] connections

Regional aircraft IMA (production workload)
  Globally time-triggered
  49 processors, 2 multi-drop busses
  244 processes (TBD threads), 3179 [source, destination] connections
Rockwell Collins Large Proof of Concept

See full presentation on AADL web site

- Generic Display System with Rockwell Collin’s Switched Ethernet LAN
  - Only LAN-related entities modeled
  - Model generated from Input/Output & Thread data stored in Database

- Model Size
  - 5 Common Processing Modules (Processors)
  - 13 Virtual Machines (Partitions)
  - 90 Threads
  - 165 End-to-end Data Flows

- 22,000 lines of AADL generated
- OSATE can handle 35 copies with reasonable performance on laptop, 700,000 lines
• Related strategic objective: Embedded Systems
• Type of instrument: Integrated Project
• Number of partners: 29
• Project cost: 15 M€
• Amount of EC funding: 8.3 M€
  – Roughly 50% of the project cost (the rest is funded by the partners)
• Total duration of the project: 3 Years.
• Starting date: 1st September 2004.
Partnerships
SPICES Work Packages (3 of 5)
17 EU Companies, 3 yrs, Starts Sept 2006
Wins Of SAE AADL

Static & dynamic architecture in single model
  – Improved software process
Validation based on precise semantics
  – Validated system architectures
Common architecture notation
  – Sub-contractor management
Standarized interchange format
  – Tool integration & interoperability
Alignment with UML2.0
  – UML profile, OMG MARTE

www.aadl.info
Ways to Leverage the AADL Now

• Start pilot and IR&D programs. Invest in AADL based research and tool development.

• Existing systems - capture architecture in AADL to support system evolution decisions. Add detail to support additional analysis over time.

• Contractors - analyze product databases for auto generation of AADL models. SEI developed toolset supports generation of AADL.

• Use technology transition support (SEI, Consultants).
Summary

• Model-based computer system engineering benefits
  
  Predictable runtime characteristics addressed early and throughout life cycle greatly reduces rework, integration and maintenance effort/risk

• Benefits of AADL as SAE standard
  
  AADL as standard provides confidence in language stability, broad adoption, joint advancement, common precise definition, strong tool support