The Unified Modeling Language and the SAE Avionics Architecture Description Language

Ed Colbert
Director, USC Software Engineering Certificate Program
ecolbert@usc.edu
(213) 821-1240

Based on presentation developed with Bruce Lewis, U.S. Army Aviation and Missile Command (AMCOM)

Society of Automotive Engineers (SAE) is developing standard Avionics Architecture Description Language

Basic research funded by
- U.S. Defense Advanced Research Projects Agency (DARPA)

Based on
- MetaH
  - Design by Honeywell for specification of real-time, fault-tolerant, securely partitioned, dynamically reconfigurable multi-processor system architectures
- Unified Modeling Language (UML)
  - Object Management Group's (OMG) standard language for object-oriented software development

LA ACM/IEEE 2002 -- Ed Colbert
Problems Developing Embedded Real-Time Systems

- Reliability, safety, & performance are constant concerns
- Wrong or late answer could be deadly
- Difficult to integrate
- Few means of assessing impact of decisions early
  - Often, don’t perceive that system exceeds processor resources until late
  - Adding or changing resources is expensive, if possible
  - Many projects cut back on capabilities so software fits hardware
  - Despite increased costs of integration, maintenance, & upgrading
- Typically very long lives & must be upgraded throughout
  - More capabilities required in each new system or upgrade
    - e.g. multimedia, situation awareness, mission simulation & training
  - Capacity on original processors is soon exhausted as user needs increase
    - If not exhausted when fielded
    - Hardware becomes obsolete
    - Re-hosting of software to new hardware is expensive

LA ACM/IEEE 2002 -- Ed Colbert
Current development process
- Manual, paper intensive, error prone, resistant to change
- Disjoint models
- Models not kept up

Well-designed architecture is essential
- Most architectural descriptions are
  - Informal documents
  - Usually centered on box-and-line diagrams, with explanatory prose
- Visual conventions are idiosyncratic & usually project-specific
- Results
  - Are only vaguely understood by developers
  - Cannot be analyzable for consistency or completeness
  - Are only hypothetically related to implementations
    - Properties cannot be enforced as system evolves
  - Cannot be supported by tools to help software architects with their tasks
Outline

- Problems Developing Embedded Real-Time Systems
- How Avionics Architecture Description Language Will Help
- Overview of AADL
- Draft Language Elements
- Extending UML
- Draft UML Metamodel for AADL
- AADL/UML Generic Missile Example
- SAE Standardization of AADL
- Final Notes

What is an Architecture Description Language?

- Describe high-level designs
- Treats system as collection of connected components
  - Layout of components defines structure
  - Connectors define communication
  - Component interfaces are first-class citizens
  - Attributes narrowly defines
    - Semantics for component interactions,
    - Systemic behaviors, and
    - Emergent properties
- Does NOT describe algorithms, data structures or circuits
Avionics Architecture Description Language

0 Provides notations that support domain-specific architectural style or styles
- Notations for common computation & communication paradigms
- Architecture formally specified using notation or notations
0 Models & methods to analysis
- Estimate characteristics
- Verify product characteristics
0 Provides/supports domain-specific software patterns
0 Library of configurable/generic components
- Components that satisfy architecture guidelines for "plug-in" use
- Components organized by some taxonomy

Model-Based AADL Process

Explicit Architecture Engineering Model

Architecture-based Requirements Analysis

Architecture-based Design and Implementation

Rapid Integration Predictable System Upgradeability

Architecture-based System Integration
Model-Based AADL Engineering

Outline

- Problems Developing Embedded Real-Time Systems
- How Avionics Architecture Description Language Will Help
- Overview of AADL
- Language Elements
- Draft Language Elements
- Extending UML
- Draft UML Metamodel for AADL
- AADL/UML Generic Missile Example
- SAE Standardization of AADL
- Final Notes

LA ACM/IEEE 2002 -- Ed Colbert
SAE AADL Based on MetaH

- AADL
- Standards In-Progress
- TOOLS
  - Analysts
  - System Construction
    - Analyzes schedules and complexities of software for integrated production control
  - Support tools
    - Safety, Mission-Critical Architectures
    - Real-Time, Safety, Mission-Critical
    - Large-scale, Event and Dynamic architecture capabilities

Future Production Tools

What is MetaH?

- AADL with supporting toolset for specifying, analyzing, & integrating computer control systems
  - Supports system architectures that are
    - Real-time
    - Fault-tolerant
    - Securely partitioned
    - Dynamically reconfigurable
    - Multi-processor

- Design by Honeywell

Design by Honeywell

LA ACM/IEEE 2002 -- Ed Colbert
MetaH Toolset

- Analyzes
  - Schedulability
  - Reliability
  - Safety

- Generates integrated, environment-specific code for
  - Application components
  - Executive
  - "Architectural glue"

MetaH Generated Partitioned Architecture

Strong Partitioning
- Timing Protection
- OS Call Restrictions
- Memory Protection

Portability
- Application Components
- Tailored MetaH Executive
- MetaH Kernel
MetaH Evaluation & Demonstration Projects

- Missile G&C reference architecture (AMCOM SED)
- Missile Re-engineering demonstration (AMCOM SED)
- Space Vehicle Attitude Control System (AMCOM SED)
- Reconfigurable Flight Control (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 CWR (BMDO, MaxTech)
- Adaptive Computer Systems (DARPA, Honeywell)
- Avionics System Performance Management (AFRL, Honeywell)
- Ada Software Integrated Development/Verification (AFRL, Honeywell)
- JSTo vehicle control (Honeywell)
- IFMU reengineering (Honeywell)

Effort Saved on AMCOM Generic Missile Project Using MetaH

- Total Project 50%
- Port Phase Only 90%

LA ACM/IEEE 2002 -- Ed Colbert
Problems Developing Embedded Real-Time Systems
How Avionics Architecture Description Language Will Help
Overview of AADL
Draft Language Elements
Extending UML
Draft UML Metamodel for AADL
AADL/UML Generic Missile Example
SAE Standardization of AADL
Final Notes

UML provides modeling concepts & notations for typical software modeling projects
Users may need
- Additional features and/or notations
- Non-semantic information attached to models
UML core concepts can be extended or specialized by users
- 3 built-in extension mechanisms
  • Stereotype
  • Constraint
  • Tagged Value
- Combine to form a Profile
Can extend UML metamodel by explicitly adding new metaclasses & other meta-constructs
- Depends on modeling tools or use of meta-metamodel facility
Outline

- Problems Developing Embedded Real-Time Systems
- How Avionics Architecture Description Language Will Help
- Overview of AADL
- Draft Language Elements
- Extending UML
- Draft UML Metamodel for AADL
- AADL/UML Generic Missile Example
- SAE Standardization of AADL
- Final Notes

Model-Element
  - Name: Sehp

Behavior (from Properties)
Component.Classifier (from Component)
Feature (from Properties)
Property.Type (from Properties)
Connection (from Properties)
Component.Invocation (from LanguageMark)
Property (from Properties)
System.Instructor (from System)
Avionics Architecture Description Language and UML

Problems Developing Embedded Real-Time Systems
How Avionics Architecture Description Language Will Help
Overview of AADL
Draft Language Elements
Extending UML
Draft UML Metamodel for AADL
AADL/UML Generic Missile Example
SAE Standardization of AADL
Final Notes

LA ACM/IEEE 2002 -- Ed Colbert
Avionics Architecture Description Language and UML

System Architecture

1. **AMICOM.GMSLNT System Implementation**
   - **Class Diagrams**

2. **AMICOM.GMSLNT System Architecture**
   - **Collaboration Diagram**

LA ACM/IEEE 2002 -- Ed Colbert
MissileInFlight Mode Type
Class Diagram.

<table>
<thead>
<tr>
<th>&lt;AADL_mode-type&gt;&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>MissileInFlight</td>
</tr>
</tbody>
</table>

Ports
- Sensed.Body.Rates : out Vectors.Vector-3D-Type
- FlightTime : out Standard.Float
- MissileAttitude : out Vectors.Vector-3D-Type
- MissilePosition : out Vectors.Vector-3D-Type
- RLTF : out Vectors.Vector-3D-Type
- LA ACMffIEEE : out Standard.Bolean

MissileInFlight_AMCOM

Properties
- Allowed.Bindings = Nil
- Bindings = Nil
- BuildOptions = Nil
- Criticality = Nil

<<AADL-system-type>>

MissileInFlight

<<AADL-system-type>>

Environment

<<AADL-system-type>>

Missile

MissileInFlight

<<AADL-system-type>>

Environment

<<AADL-system-type>>

Missile

LA ACM/IEEE 2002 -- Ed Colbert
Avionics Architecture Description Language and UML

LaunchVehicleCmd
Accelerations

Missile
MissileAltitude
MissilePosition
FightTime

RLTF
Fin_Actuator_Cmds

CCAADL-system-type>>
l
missile:
Missile

AADL-system-implementation>>:
Missile-Environment.AMCOM

RLTF
Fin_Actuator_Cmds

AADL-system-implementation>>:
Missile-Environment.AMCOM

LaunchVehicleCmd

AADL-system-type>>
environment:
Environment

AADL-process-type>>
RefControl:
Fin_Actuator_Control

AADL-process-type>>
Data_Aquisition:
Data_Aquisition

AADL-monitor-type>>
Mode_Monitor:
Fin_Actuator_Cmds

AADL-monitor-type>>
Mode_Monitor:
Sensed_Body_Rates

AADL-monitor-type>>
Mode_Monitor:
Sensed_Body_Rates
Problems Developing Embedded Real-Time Systems
How Avionics Architecture Description Language Will Help
Overview of AADL
Draft Language Elements
Extending UML
Draft UML Metamodel for AADL
AADL/UML Generic Missile Example
SAE Standardization of AADL
Final Notes
Avionics Architecture Description Language (AADL) and UML

9 January 2002

Requirements document (ARD5296) approved by SAE
Spring 2001
- Based on
  • Existing MetaH language, toolset
  • Demo/evaluation projects

AADL Definition
- v1.0 standard by EOY 2003 (goal)
  • Most critical requirement
  • Balloting expected – September 2003
- v2.0 standard by EOY 2005-6 (goal)
  • Less critical requirements & those requiring research
  • Current draft (v0.4) about 40-50% complete
  • About 125 pages

Requesting funding for prototyping of new AADL features

Key Players

- Bruce Lewis (U.S. Army AMCOM): SAE Chair, technology user
- Ed Colbert (USC): AADL & UML Mapping
- Peter Feiler (SEI): Secretary, Co-author, Editor, technology user
- Steve Vestal (Honeywell): Meta-H Originator, Co-author

Members
- Boeing, Rockwell, Honeywell, Lockheed Martin, Raytheon, Smith Industries, Dassault Aviation, Airbus, Nalco
- NIST, NAVAir, OSJTF, British MOD, Army, European Space Agency

Liaisons
- COTRE, NATO, GOA, POSIX, OPEN (informal), OMG (informal)

Relationships with other parties
- Australian Avionics Lab: performance analysis

LA ACM/IEEE 2002 -- Ed Colbert
Avionics Architecture Description Language and UML

9 January 2002

Outline

- Problems Developing Embedded Real-Time Systems
- How Avionics Architecture Description Language Will Help
- Overview of AADL
- Draft Language Elements
- Extending UML
- Draft UML Metamodel for AADL
- AADL/UML Generic Missile Example
- SAE Standardization of AADL
- Final Notes

AADL Summary

- AADL is Architecture Description Language & tools for embedded systems domain
  - Especially for avionics systems
  - Based on MetaH
- AADL provides a means to:
  - Specify software & hardware architecture
  - Incrementally develop from prototype to specification
  - Analyze architecture formally
  - Implement final system
    - Integrating components with hardware & automatically generated system executive & glue code
  - Evolve system rapidly
    - Within development
    - Across lifecycle

LA ACM/IEEE 2002 -- Ed Colbert 19
Things AADL Is Not Intended To Do

- Specify detailed design of objects, classes, algorithms, or data structures
  - Basic UML can provide
- GUI Design
  - Can specify GUI component(s) & connections to other components
  - But not details of screens, data types, etc.
- Web-based systems
- Client-Server Systems
  - Research topic
- Run-time software updates
  - Supports dynamic reconfiguration of designed components
  - But not addition or changes to generated configuration
  - Research topic

Things You Can Do With The AADL

- Specify architecture & components for product line architecture
  - Create reusable components
- Create adaptable workstation simulation that can be retargeted to tactical embedded system without loss of fidelity
  - Processing environment risk reduction (Software First).
- Retargeting & re-engineering embedded systems
- Analysis system performance of embedded RT systems
  - Schedule
  - Safety
  - Security
- Generation architecture with separate component generation for rapidly evolvable systems
- Build open architecture avionics systems with partitioned flight control
  - Reducing Validation & Verification cost