

AADL v3 Roadmap

AADL Meeting Jan 2017

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Peter Feiler



Overall Strategy

AADL V2.2

- publication of AADL V2.2 standard document: any moment now
- Release of OSATE 2.2.1 Update 3 (Jan 2017)
- Release of OSATE 2.2.2 will implement all of AADL V2.2
 - Major missing capability: reach down into feature groups
- New AADL V2.2 errata: <https://github.com/saeaadl/aadlv2.2>
- OSATE issue reports: <https://github.com/osate>

AADL V3

- Working slides & documents
- Prototype implementation
- AADL V3 Issues: <https://github.com/saeaadl/aadlv3>
- Discussion/working document area:
<https://github.com/saeaadl/aadlv3/wiki> and committee area at www.sae.org



AADL V3 Strategy

AADL V3 New Concepts

- Interface, configuration, binding

Rework/cleanup of existing concepts

- Type system
- Flow improvements
- Unused concepts (e.g., public/private package)
- Other candidates
- Feedback from users: how and in what form?

Draft standard document

- New document structure
- New document format
 - Aiming for Markdown
 - SAE editors will still work with Word => PDF with hyperlinks



V3 Prototyping

Separate from OSATE v2.2 release stream

- New file extension aadv3

Not yet started

Prototyping schedule and priorities

- End user need
- Validation of new concepts

Meta model changes

- Meta model for instance has not changed much from V1 to V2
 - Is not expected to change much for V3
- Meta model for declarative model
 - Current Meta model size/complexity
 - Too much enforcement by Meta model/grammar syntax
 - Validation and Quickfix correction



OSATE Infrastructure Cleanup

- V1 legacy code (e.g., AObject, Location, aadl/aaxl files)
- Error reporting/diagnostics
- Command/action
- Consolidation of public API libraries



Roadmap – Active

Compositional Interfaces (Julien, Peter*, Alexey, Jerome, Bren)

- Interface composition, Feature group improvements, Interface properties
- New: Interfaces as views, composition of properties, interface as new category?, annex composition, implementation composition as views
- Action:

Configuration and Choice points (Peter*, Brian)

- Freezing of design space and parameterized configurations
- Implementation selection for subcomponents, properties, bindings, relation to extends/prototypes
- Revised:
 - Consolidation of structure and feature classifier configuration handling
 - Inclusion of array dimensions as configuration items
- Action:



Roadmap – Active

General binding concept

- Binding type, binding point, binding instances (single target, alternative targets), Binding constraints
- Revised: Resources and resource types, Non-resource binding types, Multiplicity handling, binding of connections to platform flows, binding of features to platform layer
- Open issues: Concept of **binding** or allocation, Binding & Arrays

Array indexed connection declarations (Peter, Brendan)

- Exposure of index dimensions/sizes in interface via feature arrays
- Connection declarations with embedded index specification
- Action:



Roadmap - Candidates

Nested processors, Virtual memory, platforms (Peter, Alexey, Denis, Jerome)

Virtual platform connectivity

- Connections between virtual bus, virtual processor, virtual memory
- Virtual process/memory via virtual bus?
- Mixture of virtual and physical?
- Virtual platform flows

“Hardware” flows

- Flows between platform components
- Flow specs on hardware components
- Target of connection, virtual bus bindings

Virtual devices (Bren)

- What is the problem we are addressing:
 - Device as VHDL and SW device drivers
 - Device as part of the system architecture & part of functional architecture



Roadmap - Candidates

Unification of type systems and expression languages (Peter, Lutz*, Brian)

- Data types, property types, constraint language variable types
- Lists & sets for properties: Set with unique element semantics?
- Union of types: collapse entry point properties (3-to-1)
- *Removal of classifier/reference in expression part (typed expressions)*
- Handling of units: part of value, association via property

Property sublanguage

- Properties presented as separate sublanguage from core AADL
- Integration of proposed Units system (ISO, SysML)
- Nested naming of property sets and possibility of inheritance

Property value: Single assignment, statically scoped default with override

Specify required properties with classifier rather than applies to of definition

- Stereotype concept



Roadmap - Candidates

Usefulness of public/private package sections (Bren, Jerome)

AADL_Project propertyset & project structure (Jerome, Pierre)

Multiple (Mode) state machines aka state variables (Peter, Alex, Bren*, Jerome, Brian) [kickoff in Feb]

- Modes, BA states, EM states, hybrid annex, interacting state machines



More Candidates

Interrupt handler (Jerome)

Data aggregation via protocol

Data mapping via new binding/mapping concept

Clean up directionality of access features (Peter) [Errata]

- Need for Access_Right?

Categories on connections: make them optional or leave out? [errata]

Abstract feature as generic feature only

- Not refinable into other feature category

Abstract as generic component only

- Not refinable into other component category



New Revision Candidates

Flow trees and graphs

Flow categories: sampling, message based

- Sampling flows including up/down sampling semantics and appropriate queuing

Separation of Extends and refinement

Generic ports and other feature categories

- Ports without sampling, message (sampling/queuing) distinction
 - Note: down sampling may imply multi-element buffer
- Dealing with distinction between discrete and continuous flows
- Observable features

Alignment of spec sheet and blue print with component type and implementation

- Currently implementations represent both a variant and a realization
 - Component implementation present in both public and private package section
- Variant characteristics to choose variant
- Realization to expand instantiation



Table of Content: Organization of Std Document

Currently

- Generic package and component concepts
 - Includes annex mechanism, prototypes, aliases, abstract components as category
- Specific component categories
- Generic concept of features and specific feature categories
- Generic concept of connections and specific connection categories
- Concept of flows
- Property sublanguage
 - Property association, property and type definition
- Concept of modes
 - Impact on architecture hierarchy & topology
- Concept of system instance



Table of Content: Organization of Std Document

Proposal

- Generic architecture concepts
 - Components, features, connections/flows
 - Modes, configurations, instances
- Model organization concepts
 - Packages, aliases
 - Annexes, data sets
- Property sublanguage
- Specific component categories
- Pre-declared property sets
- Analysis specific profiles

