AADL v3 Roadmap
AADL meeting Oct 2016

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Peter Feiler

© 2016 Carnegie Mellon University

Overall Strategy

AADL V2.2

- Ballot & publication of AADL V2.2 standard document
- Release of OSATE 2.2.1 Update 2 (Oct 2016)
- New AADL V2.2 errata: https://github.com/saeaadl/aadlv2.2
  - May become smaller V3 issues
- 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 Major issues

- Require multiple meetings to discuss
- Need to ensure consistency with rest of core AADL

Smaller issues/errata

- V2.2 errata to go into current OSATE release stream
- Some may be addressed as V3 change (recorded as V3 issues)

From white paper/slides to draft standard

- Incremental prototyping of V3 implementation
- New document structure
- New document format
  - Working with SAE to improve
  - Aiming for Markdown but need to get SAE editors on board

V3 Prototyping

Separate from OSATE v2.2 release stream

- New file extension aadlv3

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
  - Impact of Meta model changes on graphical editor to be discussed with Phil

OSATE Infrastructure Cleanup

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

- Textual instance model representation
  - Compact & readable
  - Similar to declarative model
  - See separate slides for more

Prototype implementation available in OSATE 2.2.1

Roadmap – Active

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

- Interface composition, Feature group improvements, Interface properties
- July: draft standard text
- Action:

Choicepoints and configuration (Peter, Brian)

- Implementation selection for subcomponents, properties, in modes mappings, bindings, prototypes: reach down into model
- Syntactic distinction of architecture design and configuration
- 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, target sequences), Binding constraints
- New: Resources, Multiplicity handling, port binding, binding connections and flows
- Schedule:

Array indexed connection declarations (Peter, Brendan)

- Instance configuration: pattern or index mapping set on top connection
- Exposure of index dimensions/sizes via feature arrays
- Feature mapping connection, index subset mapping
- Action: revisit after array configuration, binding
- Schedule:

© 2016 Carnegie Mellon University

Roadmap - Candidates

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

"Hardware" flows

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

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

Virtual devices (Bren)

- Device as VHDL and SW device drivers
- Device as part of the system architecture & device as part of functional architecture

Roadmap - Candidates

Unification of type systems and expression languages (Alexey, Brian, Serban)

- Data types, property types, constraint language variable types
- Lists & sets for properties: Set with unique element semantics?
- Union of values: 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 default, inheritance, override, and **final**

Specify required properties with classifier rather than applies to of definition

- Compare with prototype properties

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, Julien*, Brian) [kickoff in Feb]

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

More Candidates

Table of Content (Jerome, Peter)

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]

Refinement of categories and features: can we eliminate this?

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

© 2016 Carnegie Mellon University