What's New in SAE AADL AS-5506B (AADL V2.1)

March 18, 2012

A number of errata have been reported in AS-5506A (AADL V2). They have been approved by the committee for correction and have been balloted as AS5506B (AADL V2.1). This document summarizes the changes between AADL V2 and AADL V2.1. The Errata sheet dated Dec 16, 2011 describes all non-editorial changes in detail (see

Significant syntax changes:

- Subprogram groups can now be nested (nested libraries). In addition, subprogram groups can now contain data subcomponents (representing static data in libraries).
- Data subcomponents defined within a thread no longer can be made accessible outside the thread via *provides data access*.
- Device implementations now can contain data subcomponents to represent the fact that they may maintain state. The state can then be referred to in Behavior Annex clauses.
- Connections now require a defining identifier. They were the only named model element, where the name was optional.
- A property association with mode-specific values can only be applied to one component. Note that if the value is not mode specific, the property association can list more than one component in its *applies to* clause.
- Properties can now be defined to have nested lists of values by defining the property as *list of (list of)**. An example of such nested lists is the Connection_Pattern property.
- In AADL V2 parentheses played multiple roles in property expressions in AADL V2, resulting in an ambiguous grammar.
  - The parentheses around the set of mode-specific property values have been eliminated.
  - Boolean values are now limited to true/false, eliminating the parentheses in Boolean expressions. Full Boolean expressions will be supported through a constraint annex.
  - Square brackets are now used around the set of record field values.
  - Parentheses are used around lists. A list with a single value does now require parentheses.
- The keyword *all* has been removed as an option in the *in modes* clause. Its meaning could already be expressed otherwise.

Resolution of inconsistencies and ambiguities:

- Prototype refinement in a component implementation cannot refine a prototype defined in a component type.
- In several places the V2 grammar prevented a prototype to be referenced or bound to an actual where is should be allowed.
- A subprogram call can now refer to subprogram access features in feature groups and to *provides subprogram access* in abstract types.

- Subprogram implementations can now contain subprogram subcomponents.
- The *Dispatch_Able* property has been introduced to allow you to specify whether a thread should not be dispatched in a mode that is local to the thread. For example, perform functions in one mode and may be idle in another mode; in the latter case it should not even be dispatched.
- *Runtime_Protection* and *Runtime_Protection_Support* now also apply to virtual processor.
- Virtual bus now allows connections other than *bus access* connections.
- Contained property associations can now always refer to features inside feature groups.
- Abstract features now have defining identifiers (the syntax in V2 did not allow identifiers ☺
- An abstract feature declaration is can optionally refer to a feature prototype, but not to a component classifier or prototype. The classifier will be supplied as part of the feature prototype actual.
- The same connection name cannot be used for two different mode specific connection declarations.
- Abstract feature connections now require the keyword *feature*; they previously did not require a keyword, while all other connection declarations did.
- A *provides* or *requires access* feature in a feature group now requires an identifier. The V2 syntax accidentally allowed it to be optional.
- Multiple flow implementations can be declared for the same flow specification, either for different modes, or to represent redundant flows in a replicated system.
- Assignment of a record value to a property overwrites any previously assigned record value. An assignment cannot overwrite individual record field values and inherit the rest.
- The property association for a mode transition is now optional, as it is for modes, subcomponents, etc.