AADL v2.1 errata -> 3.0 issues Jan 2016

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Peter Feiler

V2.1 Errata

Only those items that really need to be addressed before V2.2/V3.0

Status: Approved, closed, open errata lists available on the public Wiki and in AS2C working Documents at www.sae.org

Updated Standard document with change tracking available as working document at www.sae.org

Abstract feature with classifier

Issue: abstract feature currently allows for feature prototype, but not classifier

```
system s
  prototypes
    fp: feature d;
  features
    f1: feature fp; --ok
    f2: feature d; -- not ok
end s;
data d
end d;
```

Approved. Not implemented yet.

Proposed correction: Allow classifier.

- Do we want to say **port** without the specific port type? Abstract feature includes bus access.
- We had it in V2.0 but removed it.

End to End Flow Specification

Addresses: How to specify requirements on end to end flows without already having a design.

Ability to define end to end flow without the connection

Specify flow with existing information and refine later with connection information (by refinement statement or by inline editing)

```
system implementation sys.impl
subcomponents
   a : process A;
   b : process B;
flows
   fl : end to end flow a.out_flow -> b.in_flow;
end sys.impl;
```

Approved. Not documented, implemented yet.

Properties on Internal Features

Issue: in property definition the applies to need to be updated for those properties that are allowed.

- Should internal features be a subclass of features, thus, all properties of features apply to internal features?
  - Making internal features a subclass would allow properties only for internal features, but we have automatic inheritance of all feature properties.
- Internal features are an independent entity in the meta model, thus, we have to go through and add its meta model name to those predeclared properties that are allowed.

- Need white paper to exercise the example of expanding a design beyond internal feature as an abstraction.

Connection rule too restrictive

Issue: "If the port connection declaration represents a connection between ports down the containment hierarchy, then the source and destination must both be incoming ports. If the source connection end is a data access feature, then it must be a requires access feature; if it is a destination connection end it must be a provides access feature“

```
thread test_thread
features
    indp : in data port test_data;
end test_thread;

subprogram test_sp
features
    rda : requires data access test_data;
end test_sp;

thread implementation test_thread.impl
calls
    cs1: {
            a: subprogram test_sp;
        };
connections
    c2: port indp -> a.rda; -- Should not be allowed
end test_thread.impl;
```

V3.0 Consideration

Confusion about access direction or too restrictive

Transfer of Aggregate Data

Issue

- Transfer of data from multiple sources as single message

AADL V1 specification

- Feature group with property indicating intended transfer as single message
- Flexibility of switching between aggregate and individual message transfer

AADL V2 specification

- Outer component with port that has a data component with subcomponents
- Users explicitly **V3.0 Consideration** component of the outer port
- Manual change of model to change transmission protocol

Desirable

- Treat the issue as a protocol issue!!
- Bind collection of connections to an appropriate virtual bus (protocol)

Pass through Connections

Issue

- Model pass through from incoming pin to outgoing pin direct wiring. In AADL terms map an incoming port of a system to an outgoing port, where the system has subsystems.
- Interpretation 1: one semantic connection whose target is the component connected to the outgoing port.
- Interpretation 2: two semantic connections, incoming and outgoing with a ID (in->out) function by the system
  - Last time our response was that flow specification indicates input to output mapping

Mode Transition Triggers (new/old)

Issue

- Memory and bus can have modes, but no ports. Ports are used to model external mode transition triggers.

Proposal

- Allow ports on memory and buses.
- How to connect to those ports:
  - Hardware to hardware: processor or device to manage modes
  - Application layer to hardware: same issue as connections from application to processor

- Discussion:
  - Stay with event ports for triggering mode transitions. Event based dispatch effectively represents an implicit mode state machine. OK.
  - Buses and memory should have modes themselves and they are triggered by event ports. They could also inherit modes from an enclosing component, which would lead to everyone wrapping memory and buses with modes.

Priority on Data Access (new)

Issue

- Concurrency control may make use of priorities to manage exclusive access. Priority is currently not allowed on data access features.

Proposal

- Allow priority on data access.
- Ok for V2.1

Bus access vs. Bus with Bus Access Needs

From Vestal: Inside Network I want to connect the cable to the requires access of the Controller bus for which I have requires access.

Current view: Visibility of access point: Network does not requires access to Controller but to cable. Someone outside then supplies the cable to an instance of Network.

Revised view: Bus becomes accessible inside Network, thus Network should be able to connect the cable.

```
end Cable;

bus implementation Cable.Imp
end Cable.Imp;

bus Controller
features
    cabl: requires bus access Cable.Imp;
end Controller;

bus implementation Controller.Imp
end Controller.Imp;

system Network
features
    cont: requires bus access Controller.Imp;
end Network;

system implementation Network.Imp
subcomponents
    cab: bus Cable.Imp;
connections
    -- Why an error that cont.cabl can't be resolved?
    c: bus access cont.cabl <-> cab;
end Network.Imp;
```