

# Approved Errata to AADL V2.1

Feb 2016

See also [www.github.com/saeaadl/aadlv2.1/issues](http://www.github.com/saeaadl/aadlv2.1/issues)

## Table of Contents

4.5 Subcomponents.....	4
Specifying Array Sizes.....	4
4.7 Prototypes.....	4
Prototype Bindings.....	4
4.8 Annex Subclauses and 11.1 Property Sets.....	4
Annex Subclauses in Property Sets.....	4
5.1 Data Section.....	5
Provides data access for data components.....	5
5.4 Thread Section.....	5
Predeclared Complete and Error ports.....	5
Predeclared Stop and Abort ports.....	5
Provides data access for threads.....	5
6.2 Virtual Processors.....	6
Requires Bus Access in Virtual Processors.....	6
6.4 Buses.....	6
Provides Bus Access in Buses.....	6
6.3/4/5 Component Types with Modes.....	6
No ports to trigger mode transitions.....	6
8.x Component Internal Events and Processor Feature Proxies.....	7
self.<eventname> and processor.<subprogramorport> references explicitly declared entities.....	7
Properties on internal features processor feature proxies.....	7

8 Features.....	7
8.1 Classifiers on abstract features.....	7
8.1 Feature Array Legality Rules.....	8
8.2 Feature Groups and Types.....	8
Properties in Feature Group Types.....	8
Nested Naming of Feature Group Features.....	8
8.7 Bus Component Access.....	8
Refinement of abstract feature.....	8
9 Connections.....	9
Connection from incoming to outgoing features.....	9
9.2.3 New connection patterns for Component Arrays and Feature Arrays.....	9
9.2.4 Port Communication Timing.....	9
9.4 Access Connections.....	9
Bus Access Directions.....	9
9.4 Connection type matching for feature groups.....	10
9.4 Connection type Matching Equivalence/complement.....	10
10 Flows.....	10
Hardware Flows.....	10
Flow refinement definition.....	10
10.3 End to end flow.....	10
11 Properties.....	11
11.1.2 applies to all in property definitions.....	11
inherit and applies to.....	12
11.3 Annex specific sub-path in property associations.....	12
11.3 Final keyword in property association.....	12
11.4 Compute term in property expressions.....	13
12 Modes.....	13
Abstract features as mode transition triggers.....	13
A. Predeclared Properties.....	13

Circular Dependencies between AADL_Project, Timing, and Memory properties.....	13
A.1 Deployment Properties.....	14
Scheduling_Protocol.....	14
Actual_Memory_Binding.....	14
Actual_Connection_Binding and friends.....	14
Memory bindings for virtual processors.....	14
A.2 Thread Properties.....	15
Priority.....	15
A.3 Timing Properties.....	15
Process_Swap_Execution_Time.....	15
Status: Documented. Implemented.....	15
Scaling_Factor.....	15
Period.....	16
Reference_Time.....	16
A.4 Communication Properties.....	16
Data_Rate and Data_Rate_Units.....	16
Latency.....	16
Allow on Feature instead of Port.....	17
Transmission_Type.....	17
Input_Time/Output_Time.....	17
Status: Accepted (Jan 2016). Documented. Implemented.....	17
A.5 Memory Properties.....	17
Source_* Properties.....	17
Stack_Size, Data_Size, Code_Size.....	17
Byte_Count property.....	18
Memory_Size property.....	18
A.6 Programming Properties.....	18
Activate_Entrypoint_Source_Text.....	18
A.7 Modeling Properties.....	18

Classifier_Substitution_Rule.....	18
C.3 AADL Meta Model Classes.....	19
Category inheritance for features.....	19

## 4.5 Subcomponents

### Specifying Array Sizes

**Issue:** (L8) states that the array size cannot be changed once it is specified. However, we currently allow the array size to be supplied via a property. Since the property value can change in different contexts this would violate L8.

**Proposed correction:** Allow array size to only be specified via numeral or property constant.

**Status:** Accepted 4/13, Documented, Implemented

## 4.7 Prototypes

### Prototype Bindings

**Issue:** A component type/impl extension may bind a prototype. The standard states that once bound a prototype cannot be bound again. The extension may also have a prototype refinement statement. This statement has no impact as the prototype is bound. The standard is silent as to whether this is acceptable.

**Proposed correction:** Add a legality rule that a bound prototype cannot be refined.

**Status:** Accepted 4/13. Documented.

## 4.8 Annex Subclauses and 11.1 Property Sets

### Annex Subclauses in Property Sets

**Issue:** Annex subclauses are currently allowed in component types, component implementations, and feature group types. They are not allowed in property sets. It is useful to allow annex subclauses in property sets in order to specify constraints and invariants on properties. For each annex there will be one annex subclause in a property set, i.e., annex subclauses are not declared with individual property or property type declarations.

**Proposed correction:** Add annex subclause construct into property set grammar rule.

**Status:** Accepted 4/13, documented, implemented.

## 5.1 Data Section

### Provides data access for data components

**Issue:** Currently there is no way to declare access to a data subcomponent of a data component. The exception is that in a port connection declaration we do allow portname.datasubcomponentname to model aggregate data ports.

**Proposed correction:** Allow provides data access for data components. This indicates that a data subcomponent in a data component is accessible externally.

**Status:** Accepted 9/13, documented, implemented

## 5.4 Thread Section

### Predeclared Complete and Error ports

**Issue:** The standard describes those ports as predeclared and has a rule that they are declared implicitly. As I have been revising the Error Model annex I am referencing the Error port as well. Dealing with implicit declaration by referencing the port when references can occur in the core model and in annexes gets confusing and does not fit the overall AADL philosophy of explicit declaration and consistent type checking of references.

**Proposed correction:** Treat them as reserved names instead of implicitly predeclared. Require the modeler to declare the Complete port or Error port explicitly in the feature section before referencing it in connections.

**Status:** Accepted 4/13. Documented. Implemented.

### Predeclared Stop and Abort ports

**Issue:** The hybrid automaton for threads has abort and stop requests and aborted/stopped exit paths. We support complete and error, but not abort and stop. In the behavior annex and BLESS annex it may be desirable to refer to these events.

**Proposed correction:** Add stop/abort as ports with predefined semantics.

**Status:** Accepted Apr14. Documented. Implemented.

### Provides data access for threads

**Issue:** Threads can own persistent data components, but not make them accessible outside. Processes and systems can make them accessible.

**Proposed correction:** Allow provides data access for threads.

**Status:** Accepted Apr14. Documented. Implemented.

## 6.2 Virtual Processors

### Requires Bus Access in Virtual Processors

**Issue:** AADL specification does not allow to use BUS and VIRTUAL BUS to describe interactions between ARINC-653 partitions (virtual processor), e.g. throughput of messages between partitions.

**Proposed correction:** Provide the ability to declare Requires Bus Access and Requires Virtual Bus Access in components of category VIRTUAL PROCESSOR. Also consider other component types, such as device, system, processor,

**Status:** Accepted for V2.1. (Logged in Github. Look there for details.)

## 6.4 Buses

### Provides Bus Access in Buses

**Issue:** Bus access goes to the bus directly. It does not allow us to model specific number of access points to bus.

**Proposed correction:** allow provides bus access on bus

**Status:** Approved 7/13 Documented. Implemented.

## 6.3/4/5 Component Types with Modes

### No ports to trigger mode transitions

**Issue:** Bus, virtual bus, memory have modes but mode transitions cannot be triggered by incoming ports.

**Proposed correction:** Allow ports to trigger mode transitions.

**Status:** Approved 1/16 (originally pre V2.1) Implemented in OSATE since Oct 2011. documented. implemented.

## 8.x Component Internal Events and Processor Feature Proxies

**self.<eventname> and processor.<subprogramorport> references explicitly declared entities**

**Issue:** Component internal events are referred to as *Self*.<eventname> in connections and mode transitions and are assumed to exist. They are also used in EMV2 to identify detection events/event data to the core model. Currently they are not explicitly defined.

**Proposed correction:** Explicitly declare such internal events in an “internal features” section of a component implementation. Do the same for **processor** features (ports and subprogram access). These sections are after **subcomponents** and before **calls**.

**Self** is optional since we now have explicit internal feature declarations. Same for **processor** features.

Example:

```
internal features
  OutofRangeError: event data Data1;
  TimingError: event;
processor features
  APIFcn: processor SubprogramClassifier;
```

**Status:** Accepted 2/14, documented, implemented.

### Properties on internal features processor feature proxies

**Issue:** currently we do not allow any properties on these.

**Proposed correction:** Allow properties that make sense for event/event data ports, subprograms.

**Status:** Ok. Need to identify the specific list. Should align with those for ports. Logged on Github.

## 8 Features

### 8.1 Classifiers on abstract features

**Issue:** Abstract features can only refer to prototypes, not classifiers. We had it in V2 but it got accidentally removed in V2.1 when simplifying refinement to specific feature categories.

**Proposed correction:** Allow classifier.

**Status:** Accepted 7/15. [Logged on github. Look there for details.](#)

## 8.1 Feature Array Legality Rules

**Issue:** Must only be declared in threads, devices, processors (L8). This does not allow them to be used for access features.

**Proposed correction:** Allow them to be declared in component categories with access features, i.e., data, bus, memory, system, abstract.

**Status:** Accepted 4/13. Documented. Implemented.

## 8.2 Feature Groups and Types

### Properties in Feature Group Types

**Issue:** Currently only contained property associations are allowed in feature group types. They are defined to always have a containment path. This does not allow properties to be associated with the feature group type itself.

**Proposed correction:** Add property association in addition to contained property association into the grammar rule of feature group types.

**Status:** Accepted 4/13. Documented. Implemented.

### Nested Naming of Feature Group Features

In a connection declaration, features of a feature group can currently only be named one nesting level deep, e.g. feagrp.elemA but not feagrp.elemA.elemB.

**Proposed correction:** Allow nested naming of features within feature groups to an arbitrary depth.

**Status:** Discussed and accepted. Considered for V2.1.

## 8.7 Bus Component Access

### Refinement of abstract feature

**Issue:** AADL-8.7(L4) and AADL-8.7(C2) contradicts each other.

(L4) An abstract feature can be refined into a bus access. In this case, the abstract feature must not have a direction specified.

(C2) If a bus access feature is a refinement of an abstract feature, then the direction of the abstract feature, if specified, imposes a restriction on the access right, i.e., in implies read-only, and out implies write-only.



**Proposed correction:** Remove "In this case, the abstract feature must not have a direction specified." from AADL-8.7(L4).

**Status:** Documented.

## **9 Connections**

### **Connection from incoming to outgoing features**

**Issue:** Connections from incoming to outgoing features should not be allowed.

**Proposed correction:** add legality rule

**Status:** Approved 9/15. Documented. Implemented.

### **9.2.3 New connection patterns for Component Arrays and Feature Arrays**

**Issue:** Additional useful patterns are connection for odd and for even, as well as connecting to an element two over.

**Proposed correction:** Add Odd\_To\_Odd, Even\_To-Even, Next\_Next, Cyclic\_Next\_Next, Previous\_Previous, Cyclic\_Previous\_Previous

**Status:** Documented. Implemented.

### **9.2.4 Port Communication Timing**

**Issue:** 9.2.4(36) refers to Transfer\_Type property.

**Proposed correction:** Change to Transmission\_Type.

**Status:** Documented. N/A implementation.

## **9.4 Access Connections**

**Issue:** Data access connections currently do not allow you to specify access to a data subcomponent of a data component (field of a record).

**Proposed correction:** Allow datasubcomponent . dataelement\_subcomponent

**Status:** Accepted Apr14. Documented. Implemented.

## **Bus Access Directions**

**Issue:** AADL-9.4(1) states: Access connections for subprograms and buses are bidirectional. Data and bus access connections may be bidirectional or directional.

**Proposed correction:** Data and bus access connections may be bidirectional or directional.

**Status:** Documented.

## **9.4 Connection type matching for feature groups**

**Issue:** In feature group connections we currently require outgoing connection (outer) destination to be a subset (and vice versa for incoming). For bi-directional feature groups this is not possible. Outer should always be a super set of the inner.

**Proposed correction:** Change wording for the outer to be the super set.

**Status:** Accepted Apr14. Documented. Implemented.

## **9.4 Connection type Matching Equivalence/complement.**

**Issue:** Complement is the same as equivalence. When applying equivalence to up/down connections the direction must be the same, when going across they must be opposite.

**Proposed correction:** Change wording.

**Status:** Accepted Apr14. Documented. Nothing to implement.

# **10 Flows**

## **Hardware Flows**

**Issue:** Flows are currently restricted to software components.

**Proposed correction:** Allow for flows to go through hardware components as well. This includes adding flow specifications to hardware components and allow their inclusion in end-to-end flows and flow implementations.

**Status:** Accepted Apr14 for AADLv3. Logged in Github under V2.1 as V3 candidate.

## **Flow refinement definition**

**Issue:** Flow\_implementation\_refinement is not defined but used.

**Proposed correction:** Remove flow implementation refinement.

**Status:** 9/14. Documented. N/A implementation.

## **10.3 End to end flow**

**Issue:** There is a need to describe data flows between components before any details regarding their implementation is available. Right now an end to end flow have to reference a connection, but the connection could not be available yet. For example, we would like to define an end to end flow and to attach latency requirement to it, and then to use it as an input for the tool that will generate an architecture model satisfying to all requirements.

So, we need to be able to have such a model:

```
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;
```

**Proposed correction:** do not require to have the complete end to end flow and let people redeclare the flow. In this case syntax of `end_to_end_flow_spec` can be updated as follows.

Existing:

```
end_to_end_flow_spec ::=
  defining_end_to_end_flow_identifier : end to end flow
start_subcomponent_flow_or_etef_identifier
  { -> connection_identifier
  -> flow_path_subcomponent_flow_or_etef_identifier } +
  [ { ( property_association } + } ]
  [ in_modes_and_transitions ] ;
```

Proposed:

```
end_to_end_flow_spec ::=
  defining_end_to_end_flow_identifier : end to end flow
start_subcomponent_flow_or_etef_identifier
  [ { -> connection_identifier ->
  flow_path_subcomponent_flow_or_etef_identifier } *
  -> connection_identifier ]
  -> end_subcomponent_flow_or_etef_identifier
  [ { ( property_association } + } ]
  [ in_modes_and_transitions ] ;
```

**Status:** Accepted Apr14 (Jan 2016 decided for V2.2). Logged on Github.

## 11 Properties

### 11.1.2 applies to all in property definitions

**Issue:** In AADL V1 the applies to clause allowed for **applies to (all)** to indicate that the property can be used on any model element. In AADLV2 this was left out. Users would have to specify **applies to** (Named Element) to accomplish the same.

**Proposed correction:** Allow the keyword all in the applies to of a property definition.

**Status:** Accepted 4/13. Documented. Implemented.

## **inherit and applies to**

**Issue:** Currently when a property is declared as inherit, all super classes that the property could be inherited from must be declared explicitly in the applies to.

This creates a headache for modelers in two ways: the class hierarchy of the Meta model must be well known ; We cannot distinguish whether a property can also be associated with the enclosing classifier without being used for inheritance to the lower level components.

**Proposed correction:** Do not require declaration of ancestor classes from Meta model in applies to. Limits items in applies to to indicate whether the the specified model elements can have the property.

**Status:** Accepted during Apr14 meeting. (Jan 2016 decided to be V3) Logged on Github.

## **11.3 Annex specific sub-path in property associations**

Currently the standard allows an annex-specific subpath in a contained property association path, e.g.,

```
Occurrence => 0..1 poisson applies to system1.subsys2 annex env2 {**  
powerfailureevent **} ;
```

This allows annex-specific property associations to be included in the core model. This creates a dependency of the core model on an annex. Annex-specific property associations should be supported by the respective annex and it can do so as the property sublanguage is available to annexes and AADL V2.

**Proposed correction:** Limit annex specific contained property associations to the respective annex subclause properties section.

Simplified syntax for such paths is proposed as follows:

*sub1.sub2@(annexname)<annexmodelementpath>#propertyname* for paths starting in the core model and

*annexmodelement1.annexmodelement2* for paths purely within the annex.

**Status:** Accepted 4/13.

## **11.3 Final keyword in property association**

Users can ensure that once a value has been assigned, it cannot be changed. This is currently done with the keyword **constant** as part of a property association.

In most cases this is desirable for any use of the property. Therefore, we could indicate in the property definition that a value could be assigned only once. This could be done with a keyword **final**.

**Proposed simplification:** Allow **final** to allow only a single assignment.

**Status:** Accepted during Apr14 meeting. For V3. Logged on Github.

## 11.4 Compute term in property expressions

The **compute** term allows a user-defined function to determine the property value. As stated in the standard indicates that the result value of the user specified function must provide a value of the type expected by the property.

The standard does not explicitly state that the model object with which the property is associated is made available to the function as the first parameter.

**Proposed correction:** Add a statement to that effect.

**Status:** Accepted 4/13. Documented.

## 12 Modes

### Abstract features as mode transition triggers

**Issue:** currently limited to ports. Should allow abstract feature as well.

**Proposed change:** Change to abstract feature.

**Status** Approved 2/14. Documented. Logged on Github.

## A. Predeclared Properties

### Circular Dependencies between AADL\_Project, Timing, and Memory properties

**Issue:** Currently Time, Time\_Range are in Timing\_Properties and they refer to a constant defined in AADL\_Project which in turn refers to these property types. Same with Size and Size\_Range in Memory\_Properties and AADL\_Project.

**Proposed Correction:** Move Time, Time\_Range, Size, Size\_Range to AADL\_Project.

**Status:** Accepted 9/15. Documented. Implemented.

## A.1 Deployment Properties

### Scheduling\_Protocol

**Issue:** The property currently applies to processor and virtual processor. It is defined as **inherit**. It currently does not apply to system, thus, could not be inherited from the enclosing system of a processor.

**Proposed Correction:** Add system to the applies to.

**Status:** Accepted 4/13. Documented. Implemented.

### Actual\_Memory\_Binding

**Issue:** The property can be inherited but is not allowed in the enclosing system.

**Proposed correction:** Allow memory binding to system components

**Status:** Approved 4/14. Documented. Implemented.

### Actual\_Connection\_Binding and friends

The specification does not allow to refer systems in standard Actual\_Connection\_Binding and Actual\_Processor\_Binding properties. It prevents some kind of in-place usage of systems as models of complex hardware components.

**Proposed correction:** Add system to a list of classifiers Actual\_Connection\_Binding, Allowed\_Connection\_Binding\_Class, Allowed\_Connection\_Binding, Actual\_Processor\_Binding allowed reference to.

**Status:** Approved 4/14. Documented. Implemented.

### Memory bindings for virtual processors

**Issue:** When ARINC 653 partitions are modeled as virtual processors, it makes sense to allow users to control the binding of software components to the memory resources of that partition (as configured by the system integrator). Actual\_Memory\_Binding can only bind to a memory, but virtual processors cannot have memory subcomponents. Also note that Allowed\_Memory\_Binding cannot be done to virtual processors, although it can be done to processors.

**Proposed correction:** Allow Memory\_Bindings to virtual processors, which then gets satisfied by a memory binding of the virtual processor to a binding to memory or other component representing storage.

V3 solution: virtual memory.

**Status:** Approved as immediate fix 7/14. Documented. Implemented.

## **A.2 Thread Properties**

### **Priority**

**Issue:** There may be a priority in the concurrency control protocol. Therefore, we should allow priority on data access.

**Proposed Correction:** Allow priority on data access or data.

**Status:** New (1/2016). Approved.

## **A.3 Timing Properties**

### **Process\_Swap\_Execution\_Time**

**Issue:** process swap could also apply to virtual processor

**Proposed correction:** add it

**Status:** Documented. Implemented.

### **Scaling\_Factor**

It specifies the ratio to be used to adjust execution time of threads when deployed on different processors. The ratio is expressed with respect to a reference processor, but the Reference\_Processor property does not apply to processors.

Scaling factor is used in conjunction with Reference\_Processor, which identifies the processor to which the specified execution time applies. This property is associated with threads to identify the processor for which the execution time was specified.

The scaling factor property is redundant with a specification of the processor capacity (SEI::MIPSCapacity) or speed (SEI::cycle\_time). This creates inconsistency in the model specification if all three are specified.

We can derive the scaling factor from MIPSCapacity on processors. The processors involved are identified by the Reference\_Processor property for a thread and the actually bound processor.

We can also support a processor speed independent way of specifying the resource demand of a thread, namely Instructions Per Dispatch. This would eliminate the need for identifying a reference processor.

**Proposed Simplification:** Eliminate scaling factor. It can be derived from the processor (MIPS) capacity. Add MIPS\_Capacity as a predeclared property under the name Processor\_Capacity. This make the name consistent with the other capacity properties. Allow processor\_Capacity on processor, system, virtual processor.

**Status:** Accepted 4/13. Documented. Implemented.

## Period

Period is current not allowed on buses or virtual buses. It use at times useful to specify that a bus operates periodically. This introduces a latency similar to a sampling latency by a periodic thread.

**Proposed Correction:** Add bus and virtual bus in the applies to clause.

**Status:** Approved 8/2014. Documented. Implemented.

## Reference\_Time

Reference\_Time is described in the standard, but its definition is not included in the Timing\_Properties property set.

**Proposed Correction:** Add the definition. As described the value is a reference to a component instance that represents reference time.

**Status:** Approved Jan/2016. Documented. Implemented.

## A.4 Communication Properties

### Data\_Rate and Data\_Rate\_Units

**Issue:** It specifies the amount of data to be communicated per time unit. It used to exist in the SEI property set as Data\_Volume.

It utilizes Data\_Rate\_Units, currently called Data\_Volume\_Units. This units type will be moved into a Units property set, once established.

**Proposal:** Data\_Rate as property in Communication\_Properties and Data\_Rate\_Units in AaL\_Project. Data\_Rate can be applied to features, connections, virtual buses, buses, processors, systems, devices (all entities connections can be bound to).

**Status:** Accepted 4/13. Documented. Implemented.

## Latency



**Issue:** Latency is not allowed on system, memory, virtual bus, virtual processor, abstract feature.

**Proposal:** Add them.

**Status:** Accepted. Documented. Implemented.

### **Allow on Feature instead of Port**

**Issue:** Currently Fan\_Out\_Policy, Transmission\_Type, Input\_Rate, Input\_Time, Output\_Rate, Output\_Time can be applied to ports but not abstract features.

**Proposal:** Add them.

**Status:** Accepted. Documented. Implemented.

### **Transmission\_Type**

**Issue:** Currently Transmission\_Type can be applied to port connections, but not other connections, such as feature connection, feature group connection.

**Proposal:** Change applies to from port connection to connection

**Status:** Accepted. Documented. Implemented.

### **Input\_Time/Output\_Time**

**Issue:** In some cases the time can be anytime during the execution.

**Proposal:** Allow “dynamic” as an indicator of when input or output occurs.

**Status:** Accepted (Jan 2016). Documented. Implemented.

## **A.5 Memory Properties**

### **Source\_\* Properties**

**Issue:** Source\_Code\_Size: name suggests lines of code. Just call it Code\_Size . Same for Source\_Data\_Size, Source\_Heap\_Size, Source\_Stack\_Size.

**Proposal:** call the properties X\_Size instead of Source\_X\_Size.

**Status:** Accepted Apr14. Documented, implemented. Old property names deprecated.

### **Stack\_Size, Data\_Size, Code\_Size**

**Issue:** Currently Stack\_Size, Data\_Size, Code\_Size cannot be applied to virtual processor to specify a partition stack size.

**Proposal:** add it in.

**Status:** Accepted. Documented. Implemented.

### **Byte\_Count property**

**Issue:** Byte\_Count is deprecated, users should use Memory\_Size property instead

**Proposal:** declare the property as deprecated until AADL v3

**Status:** Accepted Apr14. Documented. Implemented,

### **Memory\_Size property**

**Issue:** Memory\_Size property is limited to memory only

**Proposal:** Allow property on processor, virtual processor, and system. In that case it represents the available memory inside these.

**Status:** Open. Documented. Implemented.

## **A.6 Programming Properties**

### **Activate\_Entrypoint\_Source\_Text**

**Issue:** the property can be used to model the entrypoint for a partition or module.

**Proposal:** Activate\_Entrypoint\_Source\_Text applies to processor and virtual processor.

**Status:** accepted Apr14. Documented. Implemented.

## **A.7 Modeling Properties**

### **Classifier\_Substitution\_Rule**

**Issue:** It specifies the classifier substitution rules for refinement declarations of features and subcomponents. Can be applied to subcomponent and classifier, but not to feature.

**Proposed correction:** Add feature to applies to clause of this property.

**Status:** Accepted 4/13. documented. Implemented.

## C.3 AADL Meta Model Classes

### Category inheritance for features

**Issue:** Bus Access, Data Access, Subprogram Access, Subprogram Group Access inherit the component category. Data port and Event Data Port do not inherit the Data component category, but they should.

**Proposed correction:** List data port and event data port as subclasses of data.

**Status:** Accepted 4/13. Not documented. Not implemented as subclass in EMF Meta model. The reason is that EMF has problems with it. Users will have to explicitly add 'bus access' as Meta class into the applies to clause of a property definition.