NamespaceModel.java

SECTION NAME	ID	POS	NEG	RULE TEXT	COMMENTARY	RULE FULL TEXT	
AADL Specification	p41n1	0	0	Identifiers in specification namespace must be unique.	Checked when processing package and property set nodesBUT: predeclared pack	(N1) An AADL specification has one global namespace. The package and property set identifiers reside i	0
AADL Specification	p41n2	0	0		0	(N2) These package and property set identifiers qualify the names of individual elements contained in the	0
AADL Specification	p41n3	0	0		0	(N3) Package declarations represent labeled namespaces for component type, component implementatic	0
AADL Specification	p41n4	0	0		0	(N4) Property set declarations represent labeled namespaces for property type and property definition de	0
AADL Specification	p41n5	0	0		Provided by parser	(N5) Packages and property sets may be separately stored. Those packages and property sets are consi	0
AADL Specification	p41n6	0	0		Provided by parser	(N6) Defining identifiers in AADL must not be one of the reserved words of the language (see Section 15.	0
AADL Specification	p41n7	0	0		Provided by realization of AADLIdentifer class	(N7) The AADL identifiers and reserved words can be in upper or lower case (or a mixture of the two) (se	0
AADL Specification	p41n8	0	0		0	(N8) The AADL does not require that an identifier be declared before it is referenced.	0
							0
Packages	p42n1	0	0	A defining package name must be unique in the global namespace.	Checked by counting private and public package declarations	(N1) A defining package name consists of a sequence of one or more package identifiers separated by a	0
Packages	p42n2	0	0		Provided by parser	(N2) The public and private section of a package may be declared in separate package declarations; thes	0
Packages	p42n3	0	0		0	(N3) Associated with every package is a package namespace that contains the names for all the element	0
Packages	p42n4	0	0		0	(N4) The package namespace is divided into a public section and a private section. Items declared in the	0
Packages	p42n5	0	0		Can be checked after all possible references are known	(N5) The reference to an item declared in another package must be an item name qualified with a packag	0
Packages	p42n6	0	0		Can be checked after all possible references are known	(N6) The reference to a property other than predeclared properties must be an property name qualified w	0
Packages	p42n7	0	0	The package name in a import_declaration must exist in the global namespace.	Checked when processing package imports	(N7) The package name in a import_declaration must exist in the global name space.	0
Packages	p42n8	0	0		Checked when processing package imports	(N8) The property set identifier in a import_declaration must exist in the global name space.	0
Packages	p42n9	0	0		Can be checked after all possible references are known	(N9) Items declared in the private section of the package can only be referenced from within the private s	0
Packages	p42n10	0	0		0	(N10) If the qualifying package identifier of a qualified reference is missing, the referenced component clas	0
Packages	p42n11global	0	0	The package name referenced in an alias_declaration must exist in the global namespace.	Checked when processing package aliases	(N11) The package name referenced in an alias_declaration must exist in the global namespace and mus	0
Packages	p42n11import	0	0	The package name referenced in an alias_declaration must be listed in the import_declaration.	Checked when processing package aliases	(N11) The package name referenced in an alias_declaration must exist in the global namespace and mus	0
Packages	p42n12	0	0	Classifier referenced is not found in public package section.	Checked when processing classifier aliases	(N12) The classifier referenced by the alias_declaration must exist in the name space of the public section	0
Packages	p42n12other	0	0	Wrong reference structure, no package is referenced	Checked when processing classifier aliases	(N12) The classifier referenced by the alias_declaration must exist in the name space of the public section	0
Packages	p42n13	0	0		Provided by parser	(N13) The classifier referenced by the alias declaration must refer to a component type or a feature group	0
Packages	p42n14package	0	0	Conflict of alias name of the package with imported package.	Checked when processing package and classifier aliases	(N14) The defining identifier of an alias_declaration must be unique in the namespace of the package con	0
Packages	p42n14defining	0	0	Conflict of alias name of the package with package name where it is defined.	Checked when processing package and classifier aliases	(N14) The defining identifier of an alias_declaration must be unique in the namespace of the package con	0
Packages	p42n14other	0	0	The defining identifier of an alias_declaration is not unique in package namespace.	Checked when processing package and classifier aliases	(N14) The defining identifier of an alias_declaration must be unique in the namespace of the package con	0
Packages	p42n15	0	0	The defining identifier of an alias_declaration is not unique in package namespace. (identifier not specified)	Checked when processing classifier aliases	(N15) The alias_declaration makes the publicly visible identifier of classifiers declared in another package	0
Packages	p42n16	0	0	The defining identifier of an alias_declaration is not unique in package namespace. ("all" alias declaration)	Checked when processing all aliases by intersection of two namespaces	(N16) If the alias_declaration renames all publicly visible identifiers of component types and feature group	0
Packages	p42n17	0	0		Can be checked after all poible references are known	(N17) The identifiers introduced by the alias_declaration are only accessible within the package. When de	0
Packages	p42n18	0	0		0	(N18) The alias declared for a component type can be used instead of a qualified component type in a refe	0
Packages	p42l1	0	0		Provided by parser	(L1) The defining package name following the reserved word end must be identical to the defining packag	0
Packages	p42l2	0	0		Checked with p42n1	(L2) For each package there may be at most one public section declaration and one private section decla	0
Packages	p42l3	0	0	Public part of component implementation can contain only properties and modes if it is declared in both public and p	Checked when processing component implementations	(L3) A component implementation may be declared in both the public and private part of a package. In tha	0
Packages	p42l4	0	0	The component category does not match the category of referenced component type.	Checked when processing classifier aliases	(L4) The component category in an alias declaration must match the category of the referenced compone	0
							0
Component Types	p43n1	0	0	The defining identifier for a component type is not unique in the namespace of the package.	Checked when creating local namespaces of the packagesBUT: currently no checl	(N1) The defining identifier for a component type must be unique in the namespace of the package within	0
Component Types	p43n2	0	0	Identifier is not unique in component type local namespace.	Checked when creating local namespaces of component types	(N2) Each component type has a local namespace for defining identifiers of prototypes, features, modes,	0
Component Types	p43n3	0	0	Ancestor in a component type extension must exist.	Checked when processing component type declarationsBUT: does not work with al	(N3) The component type identifier of the ancestor in a component type extension, i.e., that appears after	0
Component Types	p43n4	0	0	Identifier is not unique in component type namespace because of it's ancestors.	Checked by recursive creating ancestors namespace when processing component	(N4) When a component type extends another component type, a component type namespace includes a	0
Component Types	p43n5	0	0		0	(N5) A component type that extends another component type does not include the identifiers of the imple	0
Component Types	p43n6	0	0		Same as p43n2?	(N6) The defining identifier of a feature, flow specification, mode, mode transition, or prototype must be u	0
Component Types	p43n7	0	0		0	(N7) The refinement identifier of a feature, flow specification, or prototype refinement refers to the closest	0
Component Types	p43n8	0	0		0	(N8) The prototypes referenced by prototype binding declarations must exist in the local namespace of th	0
Component Types	p43n9	0	0		0	(N9) Mode transitions declared in the component type may not refer to event or event data ports of subco	0
Component Types	p43l1	0	0		Provided by parser	(L1) The defining identifier following the reserved word end must be identical to the defining identifier that	0
Component Types	p43l2	0	0		Provided by parser(kinda, error is - "No viable alternative")	(L2) The prototypes, features, flows, modes, and properties subclauses are optional. If a subclause is pre	0
Component Types	p43l3	0	0	The category of the component type being extended must match the category of the extending component type.	Checked when processing component type declarations	(L3) The category of the component type being extended must match the category of the extending comp	0
Component Types	p43l4	0	0		0	(L4) The classifier being extended in a component type extension may include prototype bindings. There	0
Component Types	p43l5	0	0		Provided by parser(kinda, error is - extraneous input 'requires' expecting {'ANNEX'	(L5) A component type must not contain both a requires_modes_subclause and a modes_subclause.	0
Component Types	p43l6	0	0	Component type and it's ancestor should have both modes or requires modes subclauses.	Checked when processing component type declarations	(L6) If the extended component type and an ancestor component type in the extends hierachy contain mo	0
Component Types	???	0	0		Provided by parser(kinda, error is - no viable alternative at input '.')	The defining identifier for a component type cannot contain '.'	0
							0
Component Implen	p44n1	0	0	Component type of component implementation is not declared.	1 - Provided by parser(kinda, error is - mismatched input 'end' expecting '.') 2 - che	(N1) A component implementation name consists of a component type identifier and a component implem	0
Component Implen	p44n2	0	0	The defining identifier for a component implementation is not unique in the local namespace of the component type.	Checked when creating local namespaces of the packagesBUT: currently no checl	(N2) The defining identifier of the component implementation must be unique within the local namespace	0
Component Implen	p44n3	0	0	Identifier is not unique in component implementation local namespace.	Checked when creating local namespaces of component implementations (without	(N3) Every component implementation defines a local namespace for all defining identifiers of prototypes	0
Component Implen	p44n4	0	0	Component implementation contains identifier which intersects with component type local namespace.	Checked by intersection of local namespaces of type and impl.Problem: error mark	(N4) This local namespace inherits the namespace of the associated component type, i.e., defining identi	0
Component Implen	p44n5	0	0		0	(N5) Refinement identifiers of features must exist in the namespace of the associated component type or	0
Component Implen	p44n6	0	0		0	(N6) In a component implementation extension, the component type identifier of the component implemer	0
Component Implen	p44n7	0	0		0	(N7) When a component implementation extends another component implementation, the local namespa	0

NamespaceModel.java

Component Implem	p44n8	0	0		0		(N8) Within the scope of the component implementation, subcomponent declarations, connections, subpr	0
Component Implem	p44n9	0	0		0		(N9) The prototype referenced by the prototype binding declaration must exist in the local namespace of t	0
Component Implem	p44l1	0	0		0		(L1) The pair of identifiers separated by a dot (вЂњ.вЂќ) following the reserved word end must be identic	0
Component Implem	p44l2	0	0		0		(L2) The prototypes, subcomponents, connections, calls, flows, modes, and properties subclauses are op	0
Component Implem	p44l3	0	0	Component implementation category does not match it's component type category.	Checked when processing classifier implementations		(L3) The category of the component implementation must be identical to the category of the component ty	0
Component Implem	p44l4	0	0	The category of the component implementation being extended must match the category of the extending compone	Checked when processing classifier implementations		(L4) If the component implementation extends another component implementation, the category of both n	0
Component Implem	p44l5	0	0		0		(L5) The classifier being extended in a component implementation extension may include prototype bindi	0
Component Implem	p44l6	0	0	Component implementation must not contain mode subclause because component type contains requires modes s	Checked when processing classifier implementations,	BUT: not watching at ancesto	(L6) If the component type of the component implementation contains a requires_modes_subclause then	0
Component Implem	p44l7	0	0	If modes are declared in the component type, then modes cannot be declared in component implementations.	Checked when processing classifier implementations,	BUT: not watching at ancesto	(L7) If modes are declared in the component type, then modes cannot be declared in component impleme	0
Component Implem	p44l8	0	0		0		(L8) If modes or mode transitions are declared in the component type, then mode transitions can be adde	0
Component Implem	p44l9	0	0		0		(L9) The category of a subcomponent being refined must match the category of the refining subcompone	0
Component Implem	p44l10	0	0		0		(L10) For all other refinement declarations the categories must match (see the respective sections).	0
Component Implem	p44l11	0	0		0		(L11) Component implementations and component implementation extensions must not refine prototypes	0
								0
Subcomponents	p45n1	0	0		0		(N1) The defining identifier of a subcomponent declaration placed in a component implementation must b	0
Subcomponents	p45n2	0	0		0		(N2) The defining identifier of a subcomponent refinement must exist as a defining subcomponent identifi	0
Subcomponents	p45n3	0	0		0		(N3) The component type identifier or the component implementation name of a component classifier refe	0
Subcomponents	p45n4	0	0		0		(N4) The prototype identifier of a prototype reference must exist in the local name space of the componer	0
Subcomponents	p45n5	0	0		0		(N5) The prototype referenced by the prototype binding declarations must exist in the local namespace of	0
Subcomponents	p45n6	0	0		0		(N6) The modes named in the in modes statement of a subcomponent must refer to modes in the compo	0
Subcomponents	p45l1	0	0		0		(L1) The category of the subcomponent declaration must match the category of its corresponding compor	0
Subcomponents	p45l2	0	0		0		(L2) The component classifier reference of a subcomponent declaration may include prototype bindings f	0
Subcomponents	p45l3	0	0		0		(L3) In a subcomponent refinement declaration the component category may be refined from abstract to c	0
Subcomponents	p45l4	0	0		0		(L4) The Classifier_Substitution_Rule property specifies the rule to be applied when a refinement supplie	0
Subcomponents	p45l5	0	0		0		(L5) In the case of a signature match, the component type of the subcomponent being refined must have	0
Subcomponents	p45l6	0	0		0		(L6) The component category and optional component classifier or prototype reference can be followed b	0
Subcomponents	p45l7	0	0		0		(L7) The array size specification for the dimensions is optional. In this case the array declaration is consid	0
Subcomponents	p45l8	0	0		0		(L8) When refining a subcomponent array the number of dimensions of the array cannot be changed, but	0
Subcomponents	p45l9	0	0		0		(L9) When the subcomponent is declared as an array with array dimension sizes then a list of component	0
Subcomponents	p45l10	0	0		0		(L10) Selecting index ranges in one or more dimensions of an array is only possible if the size of the array	0
Subcomponents	p45l11	0	0		0		(L11) An array element implementation list is valid only if (a) the subcomponent classifier is a component t	0
Subcomponents	p45c1	0	0		0		(C1) The classifier of a subcomponent cannot recursively contain subcomponents with the same compone	0
								0
Abstract Compone	p46l1	0	0		0		(L1) An abstract component type declaration can contain feature declarations (including abstract feature d	0
Abstract Compone	p46l2	0	0		0		(L2) An abstract component implementation can contain subcomponent declarations of any category. Cer	0
Abstract Compone	p46l3	0	0		0		(L3) An abstract component implementation can contain a modes subclause, a connections subclause, a	0
Abstract Compone	p46l4	0	0		0		(L4) An abstract subcomponent can be contained in the implementation of any component category.	0
Abstract Compone	p46l5	0	0		0		(L5) If an abstract subcomponent is refined to a concrete category, the concrete category must be accept	0
Abstract Compone	p46l6	0	0		0		(L6) An abstract subcomponent can be declared as an array of subcomponents.	0
Abstract Compone	p46l7	0	0		0		(L7) If an abstract component type is refined to a concrete category, the features, modes, and flow specifi	0
Abstract Compone	p46l8	0	0		0		(L8) If an abstract component implementation is refined to a concrete category, the subcomponents, call s	0
								0
Prototypes	p47n1	0	0		0		(N1) The prototype identifier on the left-hand side of a prototype binding must exist in the local namespac	0
Prototypes	p47n2	0	0		0		(N2) The prototype identifier on the right-hand side of a prototype binding, if present, must exist in the loc	0
Prototypes	p47n3	0	0		0		(N3) Unique component classifier references must exist in the public section of the package being identifi	0
Prototypes	p47n4	0	0		0		(N4) Unique feature group type references must exist in the public section of the package being identified	0
Prototypes	p47l1	0	0		0		(L1) The component category declared in the component prototype binding must match the component ca	0
Prototypes	p47l2	0	0		0		(L2) The component category of the optional component classifier reference in the prototype declaration r	0
Prototypes	p47l3	0	0		0		(L3) If the component prototype only specifies a component category, then any component type and com	0
Prototypes	p47l4	0	0		0		(L4) If the component prototype declaration includes a component classifier reference, then the classifier	0
Prototypes	p47l5	0	0		0		(L5) The category of the component implementation that contains the prototype declaration places restric	0
Prototypes	p47l6	0	0		0		(L6) If the direction is declared for feature prototypes, then the prototype actual satisfies the direction acc	0
Prototypes	p47l7	0	0		0		(L7) In the case of feature group prototypes, the supplied feature group types must match the declared fe	0
Prototypes	p47l8	0	0		0		(L8) A classifier supplied in a feature prototype binding must match the classifier of the prototype declarat	0
Prototypes	p47l9	0	0		0		(L9) Component prototypes declared with square brackets specify that they expect a list of component cla	0
Prototypes	p47l10	0	0		0		(L10) The component category of the classifier reference or prototype reference in a prototype binding dec	0
Prototypes	p47l11	0	0		0		(L11) If a direction is specified for an abstract feature in a prototype declaration, then the direction of the p	0
Prototypes	p47l12	0	0		0		(L12) Component prototype bindings must only bind component prototypes, feature group prototype bindir	0
Prototypes	p47l13	0	0		0		(L13) Component prototype refinements must only refine component prototypes, feature group prototype r	0
								0
Annex Subclauses	p48n1	0	0		0		(N1) The annex identifier must be the name of an approved annex or a project-specific identifier different	0

NamespaceModel.java

Annex Subclauses	p48n2	0	0		0	(N2) The mode identifiers in the in_modes statement must refer to modes in the component type or comp	0
Annex Subclauses	p48l1	0	0		0	(L1) Annex subclauses can only be declared in component types, component implementations, and featu	0
Annex Subclauses	p48l2	0	0		0	(L2) A component type, component implementation, or feature group type declaration may contain at mos	0
Annex Subclauses	p48l3	0	0		0	(L3) Annex libraries must be declared in packages.	0
Annex Subclauses	p48l4	0	0		0	(L4) A package declaration may contain at most one annex library declaration for each annex.	0
							0
Data	p51l1	0	0		0	(L1) A data type declaration can contain provides subprogram access declarations as well as property as	0
Data	p51l2	0	0		0	(L2) A data type declaration must not contain a flow specification or modes subclause.	0
Data	p51l3	0	0		0	(L3) A data implementation can contain abstract, data and subprogram subcomponents, access connecti	0
Data	p51l4	0	0		0	(L4) A data implementation must not contain a flow implementation, an end-to-end flow specification, or a	0
							0
Subprograms and	p52n1	0	0		0	(N1) The defining identifier of a subprogram call sequence declaration must be unique within the local na	0
Subprograms and	p52n2	0	0		0	(N2) The defining identifier of a subprogram call declaration must be unique within the local namespace	0
Subprograms and	p52n3	0	0		0	(N3) If the called subprogram name is a subprogram classifier reference, its component type identifier or	0
Subprograms and	p52n4	0	0		0	(N4) The subprogram classifier reference of a subprogram call may be a subprogram type reference.	0
Subprograms and	p52n5	0	0		0	(N5) If the called subprogram name is a subprogram subcomponent reference, the subprogram subcomp	0
Subprograms and	p52n6	0	0		0	(N6) If the called subprogram name is a requires subprogram access reference, the requires subprogram	0
Subprograms and	p52l1	0	0		0	(L1) A subprogram type declaration can contain parameter, out event port, out event data port, and featu	0
Subprograms and	p52l2	0	0		0	(L2) A subprogram implementation can contain abstract, subprogram, and data subcomponents, a subpr	0
Subprograms and	p52l3	0	0		0	(L3) Only one subprogram call sequence can apply to a given mode.	0
Subprograms and	p52c1	0	0		0	(C1) The reference to a provides subprogram access of a processor in a subprogram call (processor . pr	0
Subprograms and	p52c2	0	0		0	(C2) A subprogram call may reference a subprogram classifier. A project may enforce a consistency rule t	0
							0
Subprogram Group	p53n1	0	0		0	(N1) The defining identifier of a subprogram group type must be unique within the package namespace o	0
Subprogram Group	p53n2	0	0		0	(N2) Each subprogram group provides a local namespace. The defining subprogram identifiers of subpro	0
Subprogram Group	p53n3	0	0		0	(N3) The local namespace of a subprogram group type extension includes the defining identifiers in the lo	0
Subprogram Group	p53n4	0	0		0	(N4) The defining subprogram identifiers of subprogram access feature declarations in feature group refi	0
Subprogram Group	p53n5	0	0		0	(N5) The package name of the unique subprogram group type reference must refer to a package name i	0
Subprogram Group	p53l1	0	0		0	(L1) A subprogram group type can contain provides and requires subprogram access, and provides and	0
Subprogram Group	p53l2	0	0		0	(L2) A subprogram group implementation can contain abstract, data, subprogram group, and subprogram	0
Subprogram Group	p53l3	0	0		0	(L3) A subprogram group type or implementation may contain zero or more subcomponent declarations.	0
							0
Threads	p54l1	0	0		0	(L1) A thread type declaration can contain port, feature group, requires data access declarations, as well	0
Threads	p54l2	0	0		0	(L2) A thread component implementation can contain abstract, data, subprogram, and subprogram group	0
Threads	p54l3	0	0		0	(L3) The Complete out event port, and Error out event data port are predeclared, i.e., are implicitly identifi	0
Threads	p54c3	0	0		0	(C3) Either the Compute_Entrypoint, Compute_Entrypoint_Source_Text Compute_Entrypoint_Call_Sequ	0
Threads	p54c4	0	0		0	(C4) The Period property must have a value if the Dispatch_Protocol property value is periodic, sporadic,	0
							0
Thread Groups	p55l1	0	0		0	(L1) A thread group component type can contain provides and requires data access, as well as port, featu	0
Thread Groups	p55l2	0	0		0	(L2) A thread group component implementation can contain abstract, data, subprogram, subprogram grou	0
Thread Groups	p55l3	0	0		0	(L3) A thread group implementation can contain a connections subclause, a flows subclause, a modes su	0
Thread Groups	p55l4	0	0		0	(L4) A thread group must not contain a subprogram calls subclause.	0
							0
Processes	p56l1	0	0		0	(L1) A process component type can contain port, feature group, provides and requires data access, provi	0
Processes	p56l2	0	0		0	(L2) A process component implementation can contain abstract, data, subprogram, subprogram group, th	0
Processes	p56l3	0	0		0	(L3) A process implementation can contain a connections subclause, a flows subclause, a modes subcla	0
Processes	p56l4	0	0		0	(L4) A thread group must not contain a subprogram calls subclause.	0
Processes	p56c1	0	0		0	(C1) The complete source text associated with a process component must form a complete and legal prog	0
							0
Processors	p61l1	0	0		0	(L1) A processor component type can contain port, feature group, provides subprogram access, provides	0
Processors	p61l2	0	0		0	(L2) A processor component implementation can contain declarations of memory, bus, virtual bus, virtual	0
Processors	p61l3	0	0		0	(L3) A processor implementation can contain a modes subclause, flows subclause, and a properties subc	0
Processors	p61l4	0	0		0	(L4) A processor implementation can contain bus access, subprogram access, subprogram group access	0
Processors	p61l5	0	0		0	(L5) A processor implementation must not contain a subprogram calls subclause.	0
							0
Virtual Processors	p62l1	0	0		0	(L1) A virtual processor component type can contain port, feature group, provides subprogram access, ar	0
Virtual Processors	p62l2	0	0		0	(L2) A virtual processor component implementation can contain declarations of virtual bus, virtual process	0
Virtual Processors	p62l3	0	0		0	(L3) A virtual processor implementation can contain a modes subclause, flows subclause, and a propertie	0
Virtual Processors	p62l4	0	0		0	(L4) A virtual processor implementation must not contain a subprogram calls subclause.	0
Virtual Processors	p62l5	0	0		0	(L5) A virtual processor implementation can contain subprogram access, subprogram group access, port,	0
Virtual Processors	p62c1	0	0		0	(C1) In a fully bound system every virtual processor must be directly or indirectly bound to, or directly or in	0

NamespaceModel.java

Virtual Processors	p62c2	0	0		0	(C2) In a fully deployed system a requires virtual bus binding of a virtual processor specified by the Requi	0
							0
Memory	p63l1	0	0		0	(L1) A memory type can contain bus access declarations, feature groups, a modes subclause, and prope	0
Memory	p63l2	0	0		0	(L2) A memory implementation can contain abstract, memory, and bus subcomponent declarations.	0
Memory	p63l3	0	0		0	(L3) A memory implementation can contain a modes subclause and property associations.	0
Memory	p63l4	0	0		0	(L4) A memory implementation can contain bus access connection declarations. Bus access connections	0
Memory	p63l5	0	0		0	(L5) A memory implementation must not contain flows subclause, or subprogram calls subclause.	0
							0
Buses	p64l1	0	0		0	(L1) A bus type can have requires bus access declarations, a modes subclause, and property associatior	0
Buses	p64l2	0	0		0	(L2) A bus type must not contain any flow specifications.	0
Buses	p64l3	0	0		0	(L3) A bus implementation can contain virtual bus and abstract subcomponent declarations.	0
Buses	p64l4	0	0		0	(L4) A bus implementation can contain a modes subclause and property associations.	0
Buses	p64l5	0	0		0	(L5) A bus implementation must not contain flows subclause, or subprogram calls subclause.	0
							0
Virtual Buses	p65l1	0	0		0	(L1) A virtual bus type can have property associations.	0
Virtual Buses	p65l2	0	0		0	(L2) A virtual bus type must not contain flow specifications.	0
Virtual Buses	p65l3	0	0		0	(L3) A virtual bus implementation can contain virtual bus subcomponent declarations.	0
Virtual Buses	p65l4	0	0		0	(L4) A virtual bus implementation can contain a modes subclause and property associations.	0
Virtual Buses	p65l5	0	0		0	(L5) A virtual bus implementation must not contain a connections subclause, flows subclause, or subprog	0
Virtual Buses	p65c1	0	0		0	(C1) In a fully deployed system virtual buses must be directly or indirectly bound to processors or buses th	0
							0
Devices	p66l1	0	0		0	(L1) A device type can contain port, feature group, provides subprogram access, provides subprogram gr	0
Devices	p66l2	0	0		0	(L2) A device component implementation must not contain a subprogram calls subclause.	0
Devices	p66l3	0	0		0	(L3) A device implementation can contain abstract, data, virtual bus, and bus subcomponents, bus acces	0
							0
Systems	p71l1	0	0		0	(L1) A system component type can contain subprogram, subprogram group, data and bus access declara	0
Systems	p71l2	0	0		0	(L2) A system component implementation can contain abstract, data, subprogram, subprogram group, pr	0
Systems	p71l3	0	0		0	(L3) A system implementation can contain a modes subclause, a connections subclause, a flows subclau	0
Systems	p71l4	0	0		0	(L4) A thread group must not contain a subprogram calls subclause.	0
Systems	p71n1	0	0		0	(N1) The defining identifier of a feature must be unique within the namespace of the associated compone	0
Systems	p71n2	0	0		0	(N2) Thread features may not be declared using the predeclared ports names Complete or Error.	0
Systems	p71n3	0	0		0	(N3) Each refining feature identifier that appears in a feature refinement declaration must also appear in a	0
Systems	p71n4	0	0		0	(N4) A feature is referenced in one of two ways. Within the component implementations for a component	0
Systems	p71n5	0	0		0	(N5) The path of a contained property association for a feature must refer to an element of a feature grou	0
Systems	p71l1	0	0		0	(L1) Each feature can be refined at most once in the same type extension.	0
Systems	p71l2	0	0		0	(L2) A feature refinement declaration of a feature and the original feature must both be declared as port, p	0
Systems	p71l3	0	0		0	(L3) Feature arrays must only be declared for threads, devices, and processors.	0
Systems	p71l4	0	0		0	(L4) If the feature refinement specifies an array dimension, then the feature being refined must have an a	0
Systems	p71l5	0	0		0	(L5) If the refinement specifies an array dimension size, then the feature being refined must not have an a	0
Systems	p71l6	0	0		0	(L6) A contained property association must only be used when the feature is a feature group.	0
Systems	p71l7	0	0		0	(L7) In the case of a feature with a classifier reference, the classifier of the refined feature declaration in a	0
							0
Abstract Features	p81l1	0	0		0	(L1) The feature direction in a refined feature declaration must be identical to the feature direction in the fe	0
Abstract Features	p81l2	0	0		0	(L2) If the direction of an abstract feature is specified, then the direction must be satisfied by the refineme	0
Abstract Features	p81l3	0	0		0	(L3) An abstract feature with a feature prototype identifier and the prototype being referenced must both s	0
Abstract Features	p81l4	0	0		0	(L4) An abstract feature refinement declaration of a feature with a feature prototype reference must only a	0
							0
Feature Groups an	p82n1	0	0		0	(N1) The defining identifier of a feature group type must be unique within the package namespace of the p	0
Feature Groups an	p82n2	0	0		0	(N2) Each feature group type provides a local namespace. The defining identifiers of prototype, feature, a	0
Feature Groups an	p82n3	0	0		0	(N3) The local namespace of a feature group type extension includes the defining identifiers in the local n	0
Feature Groups an	p82n4	0	0		0	(N4) The defining feature identifiers of feature group declarations must be unique in the local name space	0
Feature Groups an	p82n5	0	0		0	(N5) The defining feature group identifier of feature_refinement declarations in component types must exi	0
Feature Groups an	p82n6	0	0		0	(N6) The package name of the unique feature group type reference must refer to a package name in the g	0
Feature Groups an	p82n7	0	0		0	(N7) The prototype reference in a feature group declaration must refer to a prototype of the component ty	0
Feature Groups an	p82l1	0	0		0	(L1) A feature group type may contain zero or more elements, i.e., feature or feature groups. If it contains	0
Feature Groups an	p82l2	0	0		0	(L2) A feature group type can be declared to be the inverse of another feature group type, as indicated by	0
Feature Groups an	p82l3	0	0		0	(L3) Only feature group types without inverse of or feature group types with features and inverse of can be	0
Feature Groups an	p82l4	0	0		0	(L4) A feature group type that is an extension of another feature group type without an inverse of cannot c	0
Feature Groups an	p82l5	0	0		0	(L5) The feature group type that is an extension of another feature group type with features and inverse o	0
Feature Groups an	p82l6	0	0		0	(L6) A feature group declaration with an inverse of statement must only reference feature group types witl	0
Feature Groups an	p82l7	0	0		0	(L7) A feature group refinement may be refined to only add property associations. In this case inclusion o	0

NamespaceModel.java

Feature Groups an	p82l8	0	0		0	(L8) The number of feature or feature groups contained in the feature group and its complement must be	0
Feature Groups an	p82l9	0	0		0	(L9) Each of the declared features or feature groups in a feature group must be a pair-wise complement v	0
Feature Groups an	p82l10	0	0		0	(L10) If both feature group types have zero features, then they are considered to complement each other;	0
Feature Groups an	p82l11	0	0		0	(L11) Ports are pair-wise complementary if they satisfy the port connection rules specified in Section 9.2.1	0
Feature Groups an	p82l12	0	0		0	(L12) Access features are pair-wise complementary if they satisfy the access connection rules in Section !	0
Feature Groups an	p82l13	0	0		0	(L13) If an in or out direction is specified as part of a feature group declaration, then all features inside the	0
							0
Ports	p83n1	0	0		0	(N1) A defining port identifier must adhere to the naming rules specified for all features (see Section 8).	0
Ports	p83n2	0	0		0	(N2) The defining identifier of a port refinement declaration must also appear in a feature declaration of a	0
Ports	p83n3	0	0		0	(N3) The unique component type identifier of the data classifier reference must be the name of a data cor	0
Ports	p83n4	0	0		0	(N4) The prototype identifier of a prototype reference, if specified, must exist in the namespace of the con	0
Ports	p83l1	0	0		0	(L1) Ports can be declared in subprogram, thread, thread group, process, system, processor, virtual proce	0
Ports	p83l2	0	0		0	(L2) Data and event data ports may be incompletely defined by not specifying the data component classif	0
Ports	p83l3	0	0		0	(L3) Data, event, and event data ports may be refined by adding a property association. The data compor	0
Ports	p83l4	0	0		0	(L4) The port category of a port refinement must be the same as the category of the port being refined, or	0
Ports	p83l5	0	0		0	(L5) The port direction of a port refinement must be the same as the direction of the feature being refined.	0
							0
Subprogram and S	p84n1	0	0		0	(N1) The defining identifier of a provides or requires subprogram or subprogram group access declaratior	0
Subprogram and S	p84n2	0	0		0	(N2) The defining identifier of a provides or requires subprogram or subprogram group refinement must e	0
Subprogram and S	p84n3	0	0		0	(N3) The component type identifier or component implementation name of a subprogram or subprogram (	0
Subprogram and S	p84n4	0	0		0	(N4) The prototype identifier of a subprogram or subprogram group access classifier reference, if present	0
Subprogram and S	p84l1	0	0		0	(L1) If a subprogram access refers to a component classifier or a component prototype, then the category	0
Subprogram and S	p84l2	0	0		0	(L2) If a subprogram group access refers to a component classifier or a component prototype, then the ca	0
Subprogram and S	p84l3	0	0		0	(L3) An abstract feature can be refined into a subprogram access or a subprogram group access. In this (	0
Subprogram and S	p84l4	0	0		0	(L4) A subprogram or subprogram group access declaration that does not specify a component classifier	0
Subprogram and S	p84l5	0	0		0	(L5) A subprogram or subprogram group access declaration may be refined by adding a property associa	0
Subprogram and S	p84l6	0	0		0	(L6) A provides subprogram access cannot be refined to a requires subprogram access and a requires su	0
Subprogram and S	p84c1	0	0		0	(C1) A provides subprogram access feature indicates that a subprogram is made available to be reference	0
							0
Subprogram Paran	p85n1	0	0		0	(N1) The defining identifier of a parameter must be unique within the namespace of the subprogram type	0
Subprogram Paran	p85n2	0	0		0	(N2) The defining parameter identifier of a parameter refinement declaration must also appear in a featur	0
Subprogram Paran	p85n3	0	0		0	(N3) The data classifier reference must refer to a data component type or a data component implementat	0
Subprogram Paran	p85n4	0	0		0	(N4) The prototype identifier, if present, must exist in the namespace of the subprogram classifier that cor	0
Subprogram Paran	p85l1	0	0		0	(L1) Parameters can be declared for subprogram component types.	0
Subprogram Paran	p85l2	0	0		0	(L2) A parameter declaration that does not specify a data classifier reference is incomplete. Such a refere	0
Subprogram Paran	p85l3	0	0		0	(L3) A parameter declaration may be refined by adding a property association. Inclusion of the data class	0
Subprogram Paran	p85l4	0	0		0	(L4) The parameter direction of a parameter refinement must be the same as the direction of the feature b	0
							0
Data Component A	p86n1	0	0		0	(N1) The defining identifier of a provides or requires data access declaration must be unique within the na	0
Data Component A	p86n2	0	0		0	(N2) The defining identifier of a provides or requires data access refinement must exist as a defining iden	0
Data Component A	p86n3	0	0		0	(N3) The component type identifier or component implementation name of a data access classifier referei	0
Data Component A	p86n4	0	0		0	(N4) The prototype identifier, if present, must exist in the namespace of the classifier that contains the dat	0
Data Component A	p86l1	0	0		0	(L1) If a data access refers to a component classifier or a component prototype, then the category of the (	0
Data Component A	p86l2	0	0		0	(L2) A data access declaration may be refined by refining the data classifier, by adding a property associa	0
Data Component A	p86l3	0	0		0	(L3) A provides data access cannot be refined to a requires data access and a requires data access canr	0
Data Component A	p86l4	0	0		0	(L4) An abstract feature can be refined into a data access. In this case, the abstract feature must not have	0
Data Component A	p86c1	0	0		0	(C1) A data access declaration that does not specify a data classifier reference is incomplete. Such a refe	0
Data Component A	p86c2	0	0		0	(C2) If the source code of a component does access shared data, then the component type declaration mi	0
Data Component A	p86c3	0	0		0	(C3) A data access refinement may refine an abstract feature declaration. If the abstract feature declaratic	0
							0
Bus Component Ac	p87n1	0	0		0	(N1) The defining identifier of a provides or requires bus access declaration must be unique within the nai	0
Bus Component Ac	p87n2	0	0		0	(N2) The defining identifier of a provides or requires bus refinement must exist as a defining identifier of a	0
Bus Component Ac	p87n3	0	0		0	(N3) The component type identifier or component implementation name of a bus access classifier referen	0
Bus Component Ac	p87n4	0	0		0	(N4) The prototype identifier, if present, must exist in the namespace of the classifier that contains the bus	0
Bus Component Ac	p87l1	0	0		0	(L1) If a bus access refers to a component classifier or a component prototype, then the category of the c	0
Bus Component Ac	p87l2	0	0		0	(L2) A bus access declaration may be refined by refining the bus classifier, by adding a property associati	0
Bus Component Ac	p87l3	0	0		0	(L3) A provides bus access cannot be refined to a requires bus access and a requires bus access cannot	0
Bus Component Ac	p87l4	0	0		0	(L4) An abstract feature can be refined into a bus access. In this case, the abstract feature must not have	0
Bus Component Ac	p87c1	0	0		0	(C1) A bus access declaration that does not specify a bus classifier reference is incomplete. Such a refere	0
Bus Component Ac	p87c2	0	0		0	(C2) If a bus access feature is a refinement of an abstract feature, then the direction of the abstract feature	0
Bus Component Ac	p87n1	0	0		0	(N1) The defining identifier of a defined connection declaration must be unique in the local namespace of	0

NamespaceModel.java

Bus Component Ac	p87n2	0	0		0	(N2) The connection identifier in a connection refinement declaration must refer to a named connection d	0
Bus Component Ac	p87l1	0	0		0	(L1) A connection refinement must contain at least one of the following: a connection source and destinat	0
Bus Component Ac	p87l2	0	0		0	(L2) If a semantic connection may be active in a particular mode, then the ultimate source and ultimate de	0
Bus Component Ac	p87l3	0	0		0	(L3) If a semantic connection may be active in a particular mode transition, then the ultimate source comp	0
							0
Feature Connectior	p91n1	0	0		0	(N1) A source or destination reference in a feature connection or feature connection refinement declaratic	0
Feature Connectior	p91n2	0	0		0	(N2) The subcomponent reference may refer to a subcomponent or a subcomponent array.	0
Feature Connectior	p91l1	0	0		0	(L1) If the feature connection declaration represents a connection between features of sibling component	0
Feature Connectior	p91l2	0	0		0	(L2) If the feature connection declaration represents a connection between features up the containment h	0
Feature Connectior	p91l3	0	0		0	(L3) If the feature connection declaration represents a connection between features down the containmer	0
Feature Connectior	p91l4	0	0		0	(L4) If the feature connection declaration specifies a directional connection, then the direction of the conn	0
Feature Connectior	p91l5	0	0		0	(L5) The individual connections of a semantic connection must be bidirectional or have the same directior	0
							0
Port Connections	p92n1	0	0		0	(N1) The connection identifier in a port connection refinement declaration must refer to a named port or fe	0
Port Connections	p92n2	0	0		0	(N2) A source or destination reference in a port connection or port connection refinement declaration mus	0
Port Connections	p92n3	0	0		0	(N3) The subcomponent reference may also consist of a reference to a subcomponent array.	0
Port Connections	p92n4	0	0		0	(N4) The event_or_event_data identifier of event source specifications (self.event_or_event_data_identifi	0
Port Connections	p92l1	0	0		0	(L1) In the case of a directional port connection the connection end representing the source of the flow m	0
Port Connections	p92l2	0	0		0	(L2) In the case of a bidirectional port connection either connection end can be the source. If the bidirectic	0
Port Connections	p92l3	0	0		0	(L3) If the source connection end is a data access feature it must have read access rights; if the destinatic	0
Port Connections	p92l4	0	0		0	(L4) The feature identifier of a subcomponent reference may refer to a feature array, if the subcomponent	0
Port Connections	p92l5	0	0		0	(L5) The following are acceptable sources and destinations of port connections. The left column shows cc	0
Port Connections	p92l6	0	0		0	(L6) If the port connection declaration represents a connection between ports of sibling components, then	0
Port Connections	p92l7	0	0		0	(L7) If the port connection declaration represents a connection between ports up the containment hierarch	0
Port Connections	p92l8	0	0		0	(L8) If the port connection declaration represents a connection between ports down the containment hiera	0
Port Connections	p92l9	0	0		0	(L9) The individual connections of a semantic port connection must be bidirectional or have the same dire	0
Port Connections	p92l10	0	0		0	(L10) Self.<identifier> must only be referenced as the source of a connection.	0
Port Connections	p92l11	0	0		0	(L11) A data port cannot be the destination of more than one semantic port connection unless each semai	0
Port Connections	p92l12	0	0		0	(L12) A semantic connection cannot contain connection declarations with both immediate and delayed Tin	0
Port Connections	p92l13	0	0		0	(L13) For connections between data ports, event data ports and data access, the data classifier of the sou	0
Port Connections	p92l14	0	0		0	(L14) The following rules are supported: вЂў вЂў вЂў вЂў Classifier_Match: The source data type and c	0
Port Connections	p92l15	0	0		0	(L15) If more than one port connection declaration in a semantic port connection has a property associatic	0
Port Connections	p92l16	0	0		0	(L16) A processor port specification must only be used in event connections within threads and subprogra	0
Port Connections	p92c1	0	0		0	(C1) There cannot be cycles of immediate connections between threads, devices, and processors.	0
Port Connections	p92c2	0	0		0	(C2) The processor port identifier of a processor port specification (processor.processor_port_identifier) m	0
Port Connections	p92c3	0	0		0	(C3) The Supports_Classifier_Subset_Matches property may be associated with a bus or virtual bus. This	0
Port Connections	p92c4	0	0		0	(C4) The Supports_Type_Conversions property may be associated with a bus or virtual bus. This specifie	0
							0
Parameter Connec	p93n1	0	0		0	(N1) The connection identifier in a parameter connection refinement declaration must refer to a named pa	0
Parameter Connec	p93n2	0	0		0	(N2) A source (destination) reference in a parameter connection declaration must reference a parameter	0
Parameter Connec	p93l1	0	0		0	(L1) The source of a parameter connection must be an incoming data or event data port of the containing	0
Parameter Connec	p93l2	0	0		0	(L2) The following source/destination pairs are acceptable for parameter connection declarations: threadp	0
Parameter Connec	p93l3	0	0		0	(L3) A parameter cannot be the destination feature reference of more than one parameter connection dec	0
Parameter Connec	p93l4	0	0		0	(L4) The data classifier of the source and destination must match. The matching rules as specified by the	0
							0
Access Connection	p94n1	0	0		0	(N1) The connection identifier in an access connection refinement declaration must refer to a named acce	0
Access Connection	p94n2	0	0		0	(N2) An access reference in an access connection declaration must reference an access feature of a sub	0
Access Connection	p94l1	0	0		0	(L1) The category of the source and the destination of a access connection declaration must be the same	0
Access Connection	p94l2	0	0		0	(L2) In the case of a bidirectional semantic access connection either connection end can be the source.	0
Access Connection	p94l3	0	0		0	(L3) In the case of a directional data or bus access connection the connection end representing the comp	0
Access Connection	p94l4	0	0		0	(L4) In a partial AADL model the ultimate source or destination may be a provides access feature of a cor	0
Access Connection	p94l5	0	0		0	(L5) If the access connection declaration represents an access connection between access features of sil	0
Access Connection	p94l6	0	0		0	(L6) If the access connection declaration represents a feature mapping up the containment hierarchy, the	0
Access Connection	p94l7	0	0		0	(L7) If the access connection declaration represents a feature mapping down the containment hierarchy,	0
Access Connection	p94l8	0	0		0	(L8) A requires access cannot be the source or destination feature reference of more than one access co	0
Access Connection	p94l9	0	0		0	(L9) For access connections the classifier of the provider access must match to the classifier of the requir	0
Access Connection	p94l10	0	0		0	(L10) If more than one access feature in a semantic access connection has an Access_Right property ass	0
Access Connection	p94l11	0	0		0	(L11) The category of the access connection source and destination must be identical. If the component c	0
							0
Feature Group Cor	p95n1	0	0		0	(N1) The connection identifier in a feature group connection refinement declaration must refer to a feature	0
Feature Group Cor	p95n2	0	0		0	(N2) A source or destination reference in a feature group connection declaration must reference a feature	0

NamespaceModel.java

Feature Group Cor	p95l1	0	0		0	(L1) If the feature group connection declaration represents a component connection between sibling com	0
Feature Group Cor	p95l2	0	0		0	(L2) The Classifier_Matching_Rule property specifies the rule to be applied to match the feature group cl	0
Feature Group Cor	p95l3	0	0		0	(L3) The following rules are supported for feature group connection declarations that represent a connect	0
Feature Group Cor	p95l4	0	0		0	(L4) The following rules are supported for feature group connection declarations that represent a connect	0
Feature Group Cor	p95l5	0	0		0	(L5) If the feature group connection declaration represents a connection between feature group of sibling	0
Feature Group Cor	p95l6	0	0		0	(L6) If the feature group connection declaration represents a connection between feature groups up the c	0
Feature Group Cor	p95l7	0	0		0	(L7) If the feature group connection declaration represents a connection between feature groups down th	0
Feature Group Cor	p95l8	0	0		0	(L8) A feature group connection must be bidirectional or be consistent with the direction of the source an	0