



Additional validation rules

This document lists assertions appearing in the SCA specification documents (V1.0). These assertions are not considered by the XSD scheme, but we must validate them in the code.

Currently we consider only the Service Component Architecture document. The line number of assertions are between brackets.

Status: in progress

Version	Date	Author(s)	Comments
v1.0	06/22/07	Stéphane Drapeau (Obeo)	Initial version

Component

- The component name must be unique across all the components in the composite (L142).

Service

- The name of the service has to match a name of a service defined by the implementation (L164).
- If an interface is specified it must provide a compatible subset of the interface provided by the implementation (L177)

Reference

- The name of the reference has to match a name of a reference defined by the implementation (L 192).
- Multiple target services are valid when the reference has a multiplicity of 0..n or 1..n. (L1581) (L203) (L211). idem Wire.
- WiredByImpl: if « true » is set, then the reference should not be wired statically within a composite, but left unwired (L220).
- If an interface is specified it must provide a compatible subset of the interface provided by the implementation (L226).

Wire

- For a composite used as a component implementation, wires can only link sources and targets that are contained in the same composite (irrespective of which file or files are used to describe the composite). Wiring to entities outside the composite is done through services and references of the composite with wiring defined by the next higher composite. (L1630)
- source: Valid URI schemes are <component-name>/<reference-name> where the source is a component reference. The specification of the reference name is optional if the source component only has one reference (L1621)
- target: Valid URI schemes are <component-name>/<service-name> where the target is a service of a component. The specification of the service name is optional if the target component only has one service with a compatible interface(L1625)
- A wire may only connect a source to a target if the target implements an interface that is compatible (see document) with the interface required by the source.(L1634)
- A Wire can connect between different interface languages (e.g. Java interfaces and WSDL portTypes) in either direction, as long as the operations defined by the two interface types are equivalent. They are equivalent if the operation(s), parameter(s), return value(s) and faults/exceptions map to each other.(L1648)

Property

- The name of the property has to match a name of a property defined by the implementation (L279).

- The property type specified must be compatible with the type of the property declared by the implementation (L274).

Composite

- Components within the composite cannot be referenced directly by the using component. The using component can only connect wires to the services and references of the used composite and set values for any properties of the composite. The internal construction of the composite is invisible to the using component(L1852)
- A composite used as a component implementation must honor a completeness contract. The concept of completeness of the composite implies (L1857):
 - the composite must have at least one service or at least one reference.
 - each service offered by the composite must be wired to a service of a component or to a composite reference.

Composite Reference

- The name of the reference must be unique across all the composite references in the composite (L1325).
- Composite references involve the promotion of one or more references of one or more components (within the composite???) (L1061).
- promote attribute: The value is a list of values of the form <component-name>/<reference-name> separated by spaces. The specification of the reference name is optional if the component has only one reference.(L1328).
- Target = a list of one or more of target Service, depending on multiplicity setting (L1342).
- If wireByImpl is set to true, then the reference should not be wired statically within a using composite, but left unwired (L1351).
- If an interface is specified it must provide an interface which is the same or which is a compatible superset of the interface declared by the promoted component reference (L1358).
- The value specified for the multiplicity attribute has to be compatible with the multiplicity specified on the component reference, i.e. It has to be equal or further restrict (L1364).
- The same component reference maybe promoted more than once, using different composite references, but only if the multiplicity defined on the component reference is 0..n or 1..n. The multiplicity on the composite reference can restrict accordingly. (L1392)
- Two or more component references may be promoted by one composite reference, but only when (L1395):
 - the interfaces of the component references are the same, or if the composite reference itself declares an interface then all the component references must have interfaces which are compatible with the composite reference interface
 - the multiplicities of the component references are compatible, i.e one can be the restricted form of the another, which also means that the composite reference carries the restricted form either implicitly or explicitly
 - the intents declared on the component references must be compatible – the intents which apply to the composite reference in this case are the union of the required intents specified for each of the promoted component references. If any intents contradict (e.g. mutually incompatible qualifiers for a particular intent) then there is an error.

- The promote attribute identifies one or more promoted component references. The value is a list of values of the form <component-name>/<reference-name> separated by spaces. The specification of the reference name is optional if the component has only one reference. (L1328)

Composite Service

- The name of the service must be unique across all the composite services in the composite (L1498).
- Composite services involve the promotion of one service of one of the components within the composite (L1059).
- The promote attribute value is of the form <component-name>/<service-name>. The service name is optional if the target component only has one service (L1501).
- If an interface is specified it must be the same or a compatible subset of the interface provided by the promoted component service, i.e. provide a subset of the operations defined by the component service. (L1512)

Binding

- I think there are others rules that I did not see...
- The rule which forbids mixing of wires specified with the target attribute with the specification of endpoints in binding subelements of the reference also applies to wires specified via separate wire elements. (L1592)
- uri is optional for references defined in composites used as component implementations, but required for references defined in composites contributed to SCA domains.(L2307)
- The name attribute allows distinction between multiple binding elements on a single service or reference. The default value of the name attribute is the service or reference name. When a service or reference has multiple bindings, only one can have the default value; all others must have a value specified that is unique within the service or reference.(L2319)

Constraining Type

- When an implementation is constrained by a constrainingType it must define all the services, references and properties specified in the corresponding constrainingType. The constraining type's references and services will have interfaces specified and may have intents specified. An implementation may contain additional services, additional optional references and additional optional properties, but cannot contain additional non-optional references or additional non optional properties (a non-optional property is one with no default value applied).(L2182)
- When a component is constrained by a constrainingType (via the "constrainingType" attribute), the entire componentType associated with the component and its implementation is not visible to the containing composite. The containing composite can only see a projection of the componentType associated with the component and implementation as scoped by the constrainingType of the component. For example, an additional service provided by the implementation which is not in the constrainingType associated with the component cannot be promoted by the containing composite.(L2188)
- The constrainingType can include required intents on any element. Those intents are applied to any component that uses that constrainingType. In other words, if requires="reliability"

exists on a `constrainingType`, or its child service or reference elements, then a constrained component or its implementation must include `requires="reliability"` on the component or implementation or on its corresponding service or reference. Note that the component or implementation may use a qualified form of an intent specified in unqualified form in the `constrainingType`, but if the `constrainingType` uses the qualified form, then the component or implementation must also use the qualified form, otherwise there is an error.(L2196)