

SCA Model using XSD

This document lists the corrections on the XSD files (from OSOA: <http://www.osoa.org/xmlns/sca/1.0/>) required to enable the validation of the XSD model and to reflect properly the SCA specification (V1.0).

Status: in progress

Version	Date	Author(s)	Comments
v1.0	06/06/07	Damien Fournier, Philippe Merle (INRIA)	Initial version

Validation

The SCA assembly model specifications describes a set of XSD files that defines the language of SCA assembly descriptions. XML Schema Definition <http://www.w3.org/XML/Schema> provides a means for defining the structure, content and semantics of XML documents. Giving a first look at the XSD files which define a SCA model, we observe that the files cannot be validated according to the XSD specification (tested via the XML Schema Validator available at W3C website <http://www.w3.org/2001/03/webdata/xsv>). The following errors are detected for schema defining the SCA model:

- Resource scheme are not found
- Attempt to extend with an attribute or an element already declared
- Non-deterministic content model (violate the “Unique Particle Attribution” rule)

Such non-validated model can introduce errors when parsing and handling assembly definitions. We need to ensure that XSD files reflect correctly the SCA Assembly Model Specification, especially to edit and process SCA assemblies, and to verify their structural properties. Here, we propose some corrections to enable validation of the XSD model.

Correction in sca.xsd

The file sca.xsd declares schema locations for the SCA specifications. The schema describing the JMS binding referenced at line 16 does not exist.

```
<include schemaLocation="sca-binding-jms.xsd"/>
```

Actually, this schema is given in the SCA JMS binding specification but is not available in the SCA namespace. To resolve this error we just have to create a sca-binding-jms.xsd file with the code given in the specification. Currently we just comment this line.

Correction in sca-core.xsd

The file sca-core.xsd defines the model for the main concepts of the Service Component Architecture described in the assembly model specification. In this XML schema definition file, the complex types Reference, Service, ComponentReference and ComponentService cause the schema to be non-deterministic. This error is caused by an element “choice”.

```
<choice minOccurs="0" maxOccurs="unbounded">
  <element ref="sca:binding" />
  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
</choice>
...
<any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
```

We observe that the elements “any” used twice are in conflict. Actually the element “choice” is useless. This part can simply be replaced by defining an element of type “binding” which has 0 or more occurrences. The correction applied between lines 54-61, 78-83, 202-208 and 228-233 is

written below.

```
<element ref="sca:binding" minOccurs="0" maxOccurs="unbounded" />
...
<any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
```

Name of imported composite is required for "include" elements. This is not currently described in the schema, and thus, need to be set to required (schema line 264).

Note : SCA components and composites can be configured via external properties. According to specification a property requires a value for attribute "type" or "element". The definition of a SCA property is not correct in the schema. But, because choice between two attributes can not be expressed with XSD, this constraint cannot be verified with the schema.

Correction in sca-implementation-java.xsd

The schema definition given by the file sca-implementation-java.xsd extends the schema defining main SCA concepts in order to describe components implemented with Java. This definition tries to override attributes (line 20 & 21) which is not allowed in XML schema definition.

```
<attribute name="requires" type="sca:listOfQNames" use="optional"/>
<attribute name="policySets" type="sca:listOfQNames" use="optional"/>
```

The attributes "requires" and "policySets" are already defined in the sca-core.xsd schema. The definition of these attributes in sca-implementation-java.xsd needs to be deleted.

Correction in sca-implementation-composite.xsd

As Java implementation, the file sca-implementation-composite.xsd adds definition of composites implementation to the SCA model. The attributes "requires" and "policySets" are also defined, and need to be deleted (lines 19 & 20).

Correction in sca-binding-sca.xsd

The definition in sca-binding-sca.xsd adds description of SCA bindings to the SCA model by extending the "binding" type.

```
<complexType name="SCABinding">
  <complexContent>
    <extension base="sca:Binding">
      <sequence>
        <element name="operation" type="sca:Operation"
minOccurs="0"
maxOccurs="unbounded" />
      </sequence>
      <attribute name="uri" type="anyURI" use="optional"/>
      <attribute name="name" type="QName" use="optional"/>
      <attribute name="requires" type="sca:listOfQNames"
use="optional"/>
      <attribute name="policySets" type="sca:listOfQNames"
```

```

        use="optional"/>
        <anyAttribute namespace="##any" processContents="lax"/>
    </extension>
</complexContent>
</complexType>

```

In this schema, the SCA binding overrides elements and attributes of a binding defined by the SCA core model. Redefining a type is not allowed in the XML schema definition norm. Thus, the element “operation” and the attributes “uri”, “name”, “requires” and “policySets” are useless in this definition. As a result a SCA binding is just a renaming of a binding defined in the SCA core model.

```

<complexType name="SCABinding">
    <complexContent>
        <extension base="sca:Binding">
            <anyAttribute namespace="##any" processContents="lax"/>
        </extension>
    </complexContent>
</complexType>

```

Correction in sca-binding-webservice.xsd

The sca-binding-webservice.xsd schema extends the model with description of web service bindings. Web service binding defines how a SCA assembly can be exposed as a web service and/or invoke web services. As a SCA binding, a web service binding extends the binding definition from the SCA core model. The web service binding definition we use is directly extracted from the specification document, because it is updated with the specification, contrary to the online schema.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) Copyright SCA Collaboration 2006 -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace ="http://www.oesa.org/xmlns/sca/1.0"
    xmlns:sca ="http://www.oesa.org/xmlns/sca/1.0"
    xmlns:wsdli ="http://www.w3.org/2004/08/wsdli-instance"
    xmlns:wsa ="http://www.w3.org/2004/12/addressing"
    elementFormDefault ="qualified">
    <import namespace="http://www.w3.org/2004/08/wsdli-instance"
        schemaLocation ="wsdli.xsd" />
    <import namespace="http://www.w3.org/2004/12/addressing"
        schemaLocation ="ws-addr.xsd" />
    <include schemaLocation="sca-core.xsd"/>
    <element name="binding.ws" type="sca:WebServiceBinding"
        substitutionGroup ="sca:binding"/>
    <complexType name="WebServiceBinding">
        <complexContent>
            <extension base="sca:Binding">
                <sequence>
                    <element ref="wsa:EndpointReference" minOccurs="0"
                        maxOccurs ="unbounded"/>
                    <any namespace="##other" processContents="lax" minOccurs="0"
                        maxOccurs ="unbounded"/>
                </sequence>
                <attribute name="wsdlElement" type="anyURI" use="optional"/>
                <attribute ref="wsdli:wsdlLocation" use="optional"/>
                <anyAttribute namespace="##any" processContents="lax"/>
            </extension>
        </complexContent>
    </complexType>

```

```
</complexType>
</schema>
```

However, the definition provided by the specification document has some errors. First, as schema for SCA policy, imports are wrong. We correct these references (lines 9 to 12) with the following input.

```
<import namespace="http://www.w3.org/2004/08/wsd1-instance"
        schemaLocation="http://www.w3.org/2004/08/wsd1-instance" />
<import namespace="http://www.w3.org/2005/08/addressing"
        schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd" />
```

Second, the possible extension of the web service binding type causes the model to violate the unique particle attribution rule of the XML schema definition norm. We need to delete possibility to extend this binding type (line 22 in the schema) to ensure strict validation of the schema.

Correction in sca-policy.xsd

The SCA policy document defines constructs to express non-functional requirements associated to SCA components such as quality of service (QoS), capabilities or constraints. This document is based on the Web Services Policy Framework (WS-Policy) and Web Services Policy Attachment (WS-PolicyAttachment) specification submitted to the W3C. The reference to the imported WS-Policy schema (line 11) is wrong in the SCA policy model.

```
<import namespace="http://schemas.xmlsoap.org/ws/2004/09/policy"
        schemaLocation="http://schemas.xmlsoap.org/ws/2004/09/ws-
policy.xsd"/>
```

The schema location must be corrected with the following input.

```
<import namespace="http://schemas.xmlsoap.org/ws/2004/09/policy"
        schemaLocation="http://schemas.xmlsoap.org/ws/2004/09/policy/ws-
policy.xsd"/>
```

Applying those corrections allow us to validate SCA model schema against the XML schema definition norm. The new schema provided by SCOrWare contribute to the SCA specification by giving a model definition which can be used to verify correctness of SCA assembly descriptions.

Structure

The model defined by XDS has also some structure problems. Part of the schema definition does not correspond to the written SCA specifications.

Service and Reference

The definition provided by the SCA assembly model specification makes distinctions between a Reference in a composite and a Reference in a component. The former promotes a component contained in a composite while the later describes the operations required by a component. The schema defines a component reference (noted ComponentReference) as a restriction of a composite

reference (noted Reference). The difference between a Reference and a ComponentReference is symbolized by a “promote” attribute. This attribute refers to a component which is promoted by the composite. It is defined as a required attribute for a Reference.

Extending by restriction a Reference implies that the attribute “promote” is needed in the extension. Thus, the attribute “promote” is also required in a ComponentReference. However, as described in the specification (SCA Assembly model, page 5) a reference or a service, in a component, has not a promote attribute. This points out a structural problem in the definition between a composite reference and a component reference. Instead of specifying a component reference as a restriction of a composite reference, we can define a composite reference and a component reference as an extension of a basic reference type. A reference extends this type by adding an attribute “promote” while a component reference adds an attribute “autowire”. Similarly, service expressed in a composite or service expressed in a component have the same problem. We also define Service and ComponentService as extensions of a basic service type.