SCA Service Component Architecture

Web Service Binding Specification

SCA Version 1.00, March 21 2007

Technical Contacts:

Simon Holdsworth	IBM Corporation
Sabin Ielceanu	TIBCO Software Inc.
Anish Karmarkar	Oracle
Mark Little	Red Hat
Sanjay Patil	SAP AG
Michael Rowley	BEA

Status of this Document

This specification may change before final release and you are cautioned against relying on the content of this specification. The authors are currently soliciting your contributions and suggestions. Licenses are available for the purposes of feedback and (optionally) for implementation.

Table of Contents

1 Overview

1.1 Introduction

The SCA Web Service binding specified in this document applies to the services and references of composites [1]. It defines the manner in which a service can be made available as a web service, and in which a reference can invoke a web service.

This binding is a WSDL-based binding; that means it either references an existing WSDL binding or allows one to specify enough information to generate one. When an existing WSDL binding is not referenced, rules defined in this document allow one to generate a WSDL binding.

The Web Service binding can point to an existing WSDL [2] document, separately authored, that specifies the details of the WSDL binding and portType schema to be used to provide or invoke the web service. In this case the SCA web services binding allows anything that is valid in a WSDL binding, including rpc-encoded style and binding extensions. It is the responsibility of the SCA system provider to ensure support for all options specified in the binding. Interoperation of such services is not guaranteed.

The SCA Web Service binding also provides attributes that can be used to provide the details of a WSDL SOAP binding. This allows a WSDL document to be synthesized in the case that one does not already exist. In this case only WS-I compliant mapping is supported.

In most cases it is expected that a binding applied to a composite's reference will point to an existing WSDL document that describes the web service to be invoked. The binding applied to a composite's service may use either approach.

The SCA Web Service binding can be further customized through the use of SCA Policy Sets. For example, a requirement to conform to a WS-I profile [3] could be represented with a policy set.

2 Web Service Binding

2.1 Web Service Binding Schema

The Web Service binding element is defined by the following pseudo-schema.

```
<binding.ws wsdlElement="xs:anyURI"?
            wsdli:wsdlLocation="list of xs:anyURI"?
            ...>
   <wsa:EndpointReference>...</wsa:EndpointReference>*
   ...
</binding.ws>
```

- ***/binding.ws/@wsdlElement*** – optional attribute that specifies the URI of a WSDL element. The use of this attribute indicates that the SCA binding points to the specified element in an existing WSDL document. The URI can have the following forms:
    - Service:

      <WSDL-namespace-URI>#wsdl.service(<service-name>)

      In this case, all the endpoints in the WSDL Service that have equivalent PortTypes with the SCA service or reference must be available to the SCA service or reference.
    - Port (WSDL 1.1):

      <WSDL-namespace-URI>#wsdl.port(<service-name>/<port-name>)

      In this case, the identified port in the WSDL 1.1 Service must have an equivalent PortType with the SCA service or reference.
    - Endpoint (WSDL 2.0):

      <WSDL-namespace-URI>#wsdl.endpoint(<service-name>/<endpoint-name>)

      In this case, the identified endpoint in the WSDL 2.0 Service must have an equivalent PortType with the SCA service or reference.
    - Binding:

      <WSDL-namespace-URI>#wsdl.binding(<binding-name>)

      In this case, the identified WSDL binding must have an equivalent PortType with the SCA service or reference. In this case the endpoint address URI for the SCA service or reference must be provided via the URI attribute on the binding.
- ***/binding.ws/@wsdli:wsdlLocation*** – optional attribute that specifies the location of the WSDL document. This attribute can be specified in the event that the <WSDL-namespace-URI> in the 'endpoint' attribute is not dereferencable, or when the intended WSDL document is to be found at a different location than the one pointed to by the <WSDL-namespace-URI>. The use of this attribute indicates that the WSDL binding points to an existing WSDL document.
- ***/binding.ws/wsa:EndpointReference*** – optional WS-Addressing [6] EndpointReference that specifies the endpoint for the service or reference. When this element is present along with the wsdlElement attribute on the parent element, the wsdlElement attribute value MUST be of the 'Binding' form as specified above, i.e. <WSDL-namespace-URI>#wsdl.binding(<binding-name>).
- ***/binding.ws/@{any}*** - this is an extensibility mechanism to allow extensibility via attributes.

- ***/binding.ws/any*** – this is an extensibility mechanism to allow extensibility via elements.

2.1.1 Endpoint URI resolution

The rules for resolving the URI at which an SCA service is hosted, or SCA reference targets, when used with binding.ws (in precedence order) are:

1. The URIs in the endpoint(s) of the referenced WSDL
   or
   The URI specified by the wsa:Address element of the wsa:EndpointReference,
2. The explicitly stated URI in the "uri" attribute of the binding.ws element, which may be relative,
3. The implicit URI as defined by the Assembly specification

The URI in the WSDL endpoint or in the wsa:Address of an EPR may be a relative URI, in which case it is relative to the URI defined in (2) or (3). The wsa:Address element can be the empty relative URI, in which case it uses the URI defined in (2) or (3) directly. This allows the EPR writer to specify reference parameters, metadata and other EPR contents while allowing the URI to be chosen by the deployer.

To reference a WSDL document and also specify an EPR, the wsdlElement attribute must refer to a binding element in the WSDL and not an endpoint or service.

2.1.2 Interface mapping

When binding.ws is used on a service or reference with an interface that is not defined by interface.wsdl, then a WSDL interface for the service or reference is derived from the interface by the rules defined for that interface type.

For example, for interface.java, the mapping to a WSDL portType is as defined in the SCA Assembly Specification [1].

Binding.ws implementations may use appropriate standards, for example WS-I AP 1.0 or MTOM, to map interface parameters to binary attachments transparently to the target component.

2.1.3 Production of WSDL description for an SCA service

Any service with one or more web service bindings with HTTP endpoints SHOULD return a WSDL description of the service in response to an HTTP GET request with the “?wsdl” suffix to that HTTP endpoint. If none of the web service bindings have HTTP endpoints, then some other means of obtaining the WSDL description of the service should be provided. This may include out of band mechanisms, for example publication to a UDDI registry.

Refer to section 2.3 for a detailed definition of the rules that SHOULD be used for generating the WSDL description of an SCA service with one or more web service bindings.

2.1.4 Additional binding configuration data

SCA runtime implementations may provide additional metadata that is associated with a web service binding, for example to enable JAX-WS [4] handlers to be executed as part of the target component dispatch. The specification of such metadata is SCA runtime-specific and is outside of the scope of this document.

2.1.5 Web Service Binding and SOAP Intermediaries

The Web Service binding does not provide any direct or explicit support for SOAP intermediaries [5].

2.1.6 Support for WSDL extensibility

When a Web Service binding is specified using the wsdlElement attribute, the details of the binding are specified by the WSDL element referenced by the value of the attribute. WSDL elements allow for extensibility via elements as well as attribute. The Web Service binding does not curtail the use of such extensibility in WSDL. Note that as a consequence of this, when using this form of Web Service binding, it is not possible to determine whether the binding is supported by the SCA runtime without parsing the referenced WSDL element and its dependent elements.

2.2 Web Service Binding Examples

The following snippets show the sca.composite file for the MyValueComposite file containing the service element for the MyValueService and reference element for the StockQuoteService. Both the service and the reference use a Web Service binding.

2.2.1 Example Using WSDL documents

This example shows a service and reference using the SCA Web Service binding, using existing WSDL documents in both cases. In each case there is a single binding element, whose name defaults to the service/reference name.

The service's binding is defined by the WSDL document associated with the given URI. This service must be conformant with the WS-I basic profile 1.1.

The reference's first binding is defined by the specified WSDL service in the WSDL document at the given location. The reference may use any of the WSDL service's ports/endpoints to invoke the target service. The reference's second binding is defined by the specified WSDL binding. The specific endpoint URI to be invoked is provided via the URI attribute.

```

<?xml version="1.0" encoding="ASCII"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0" name="MyValueComposite">
   <service name="MyValueService">
      <interface.java interface="services.myvalue.MyValueService"/>
      <binding.ws wsdlElement="http://www.myvalue.org/MyValueService#
                         wsdl.endpoint(MyValueService/MyValueServiceSOAP)"/>
         ...
   </service>

   ...

   <reference name="StockQuoteReference1">
      <interface.java interface="services.stockquote.StockQuoteService"/>
      <binding.ws wsdlElement="http://www.stockquote.org/StockQuoteService#
                         wsdl.service(StockQuoteService)"
                  wsdli:wsdlLocation="http://www.stockquote.org/StockQuoteService
                         http://www.stockquote.org/StockQuoteService.wsdl"/>
   </reference>

   <reference name="StockQuoteReference2">
      <interface.java interface="services.stockquote.StockQuoteService"/>
      <binding.ws wsdlElement="http://www.stockquote.org/StockQuoteService#
```

```
                                  wsdl.binding(StockQuoteBinding)"

                     wsdli:wsdlLocation="http://www.stockquote.org/StockQuoteService
                                  http://www.stockquote.org/StockQuoteService.wsdl"/>
                     uri="http://www.stockquote.org/StockQuoteService5"/>
   </reference>
</composite>
```

2.2.2 Examples Without a WSDL Document

The next example shows the simplest form of the binding element without WSDL document, assuming all defaults for portType mapping and SOAP binding synthesis. The service and reference each have a single binding element, whose name defaults to the service/reference name.

The service is to be made available at a location determined by the deployment of this component. It will have a single port address and SOAP binding, with a simple WS-I BP 1.1 compliant binding, and using the default options for mapping the Java interface to a WSDL port type.

The reference indicates a service to be invoked which must have a SOAP binding and portType that matches the default options for binding synthesis and interface mapping. One particular use of this case would be where the reference is to an SCA service with a web service binding which itself uses all the defaults.

```
<?xml version="1.0" encoding="ASCII"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
           name="MyValueComposite">

   <service name="MyValueService">
      <interface.java interface="services.myvalue.MyValueService"/>
      <binding.ws/>
      ...
   </service>

   ...

   <reference name="StockQuoteService">
      <interface.java interface="services.stockquote.StockQuoteService"/>
      <binding.ws uri="http://www.sqs.com/StockQuoteService"/>
   </reference>
</composite>
```

The next example shows the use of the binding element without a WSDL document, with multiple SOAP bindings with non-default values. The SOAP 1.2 binding name defaults to the service name, the SOAP 1.1 binding is given an explicit name. The reference has a web service binding which uses SOAP 1.2, but otherwise uses all the defaults for SOAP binding. The reference binding name defaults to the reference name.

```
<?xml version="1.0" encoding="ASCII"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
           name="MyValueComposite">

   <service name="MyValueService">
      <interface.java interface="services.myvalue.MyValueService"/>
      <binding.ws name="MyValueServiceSOAP11" requires="soap/1.1"/>
      <binding.ws requires="soap/1.2"/>
```

```
        ...
    </service>

    ...

    <reference name="StockQuoteService">
       <interface.java interface="services.stockquote.StockQuoteService"/>
       <binding.ws uri="http://www.sqs.com/StockQuoteService"
                   requires="soap/1.2"/>
    </reference>
</composite>
```

2.2.3 Example PolicySet Providing The Conversation Intent

This policy set applies to binding.ws and provides the conversation intent. The conversation intent is provided by using WS-ReliableMessaging protocol which has a concept of a Sequence. This Sequence (which appears as a wsrm:Sequence SOAP header in the message) is used as a correlation mechanism, on the wire, to implement conversational semantics.

```
<policySet name="WSRM-Sequence-based-conversation"
           provides="sca:conversation"
           appliesTo="sca:binding.ws">
   <wsp:Policy>
     <wsrmp:RMAssertion
                xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200608"/>
   </wsp:Policy>
</policySet>
```

2.3 WSDL Generation

This section defines the rules that SHOULD be used for generation of a WSDL document that describes an SCA service with one or more web service bindings that require a SOAP binding.

A WSDL document may be generated for an SCA service with non-SOAP web service bindings, or other bindings. For non-SOAP web service bindings that do not refer to an existing WSDL document, or non-web service bindings, the generation rules below may be considered a template, and a similar approach taken.

2.3.1 Intents

The following intents affect WSDL generation:

- soap
  This indicates that a SOAP binding is required. The SOAP binding may be of any SOAP version, including multiple versions.
- soap.1_1
  A SOAP 1.1 binding only is required.
- soap.1_2
  A SOAP 1.2 binding only is required.

2.3.2 WSDL Service and Ports

A separate WSDL document is generated for each SCA service. Each has its own unique target namespace. This is to ensure that bindings on different services of the same component do not clash. The WSDL service has one or more ports for each web service binding on the SCA service that has a SOAP requirement, or that refers to an existing WSDL binding, depending on the requirements of the web service binding. Each of those ports has a single binding.

Additional ports and bindings may be generated in this WSDL document for non-web service bindings, or web service bindings with non-SOAP requirements. The manner in which that is done is undefined.

The binding elements themselves may be generated as defined below, or may be imported from existing WSDL documents in the case that the web service binding refers to the binding element of such a document.

The target namespace of the WSDL document, and of the service, ports and generated binding elements is:

Base System URI for HTTP / Component Name / Service Name

2.3.3 WSDL Bindings

The binding elements in the generated WSDL document are either defined within the document, derived from the requirements of the binding, or are imported from existing WSDL documents.

Generated bindings have the following fixed assumptions:

- use=”literal” for input and output messages
- style=”document” for the binding
- All faults map to soap:faults
- No header or headerFault elements are generated
- The transport is “http://schemas.xmlsoap.org/soap/http”, unless the system provides intents for alternative transports
- The soap version is determined from the soap intents as defined above

2.3.3.1 SOAP versions

Where a web service binding requires a specific SOAP version, then a single WSDL port and SOAP binding of the appropriate version is generated.

Where no specific SOAP version is required, then one or more WSDL ports with associated SOAP bindings may be generated, depending on the level(s) supported in the target runtime.

2.3.4 WSDL PortType

An SCA service has a single interface. This interface is always imported into the generated WSDL document. This may be done directly for WSDL-defined interfaces, or indirectly via a WSDL generated from the interface type for the service.

2.3.5 WSDL Generation Rules

The following is the formal definition of the generation of a WSDL document from an SCA service with one or more web service bindings, with either a SOAP requirement or existing WSDL document:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="componentName/serviceName"
             targetNamespace="HTTP Base URI/componentName/serviceName"
             {(if any bindings require SOAP 1.1)
```

```
                  xmlns:soap11="http://schemas.xmlsoap.org/wsdl/soap/"
                  }
                  {(if any bindings require SOAP 1.2)
                  [xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"]
                  }
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <import namespace="SCA service interface namespace"
          location="SCA service interface location"/>

  {(for each binding.ws element in the service with a WSDL, do the following:)
  <import namespace="existing WSDL binding namespace"
          location="existing WSDL binding location"/>
  }

  {(for each binding.ws element in the service without a WSDL, do the following
    for each SOAP version required:)
  <binding name="/service/binding.ws[n]/@name+[/soapVersionPrefix]+'Binding'"
           type="SCA service interface portType name">
    <soapVersionPrefix:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    {(for each operation in the interface do the following:)
      <operation name="name-of-the-operation">
        <soapVersionPrefix:operation/>
        <input>
          <soapVersionPrefix:body use="literal"/>
        </input>
        {(if there is an output)
          <output>
            <soapVersionPrefix:body use="literal"/>
          </output>
        }
        {(if there is a fault)
          <fault>
            <soapVersionPrefix:fault name="name-of-the-fault"/>
          </fault>
        }
      </operation>
    }
  </binding>
  }

  <service name="/service/@name">
    {(for each binding.ws element in the service do the following for each SOAP
      version required:)
      <port name="/service/binding.ws[n]/@name+[/soapVersionPrefix]+'Port'"
            binding="/service/binding.ws[n]/@name+[/soapVersionPrefix]+'Binding'">
        <soapVersionPrefix:address location="/service/binding.ws[n]/@uri"/>
      </port>
    }
  </service>
</definitions>
```

3 Web Services Binding Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) Copyright SCA Collaboration 2006 -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.osoa.org/xmlns/sca/1.0"
    xmlns:sca="http://www.osoa.org/xmlns/sca/1.0"
    xmlns:wsdli="http://www.w3.org/2004/08/wsdl-instance"
    xmlns:wsa="http://www.w3.org/2004/12/addressing"
    elementFormDefault="qualified">

    <import namespace="http://www.w3.org/2004/08/wsdl-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>
</schema>
```

4 References

[1] SCA Assembly Model Specification
http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications

[2] WSDL Specification
WSDL 1.1: http://www.w3.org/TR/wsdl
WSDL 2.0: http://www.w3.org/TR/wsdl20/

[3] WS-I Profiles
http://www.ws-i.org/Profiles/BasicProfile-1.1.html
http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html
http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html
http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html

[4] JAX-WS Specification
http://jcp.org/en/jsr/detail?id=224

[5] SOAP specification
http://www.w3.org/TR/2003/REC-soap12-part1-20030624/
http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[6] Web Services Addressing 1.0 – Core
http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/