

WSDL Port Capabilities Proposal

Authors: Jim Webber, Arjuna Technologies Ltd.

Savas Parastatidis, North-East Regional e-Science Centre, University of Newcastle upon Tyne

Version: 0.1 – DRAFT

Deleted: School of Computing Science,

1. Introduction

This paper constitutes a proposal for extending the current WSDL specification, such that services can advertise their support for arbitrary Web services protocols to address non-functional requirements such as security and reliability.

Formatted: PaperParagraph

2. Motivation

The interface and communication protocol aspects of a Web service are defined in WSDL. An interface element contains the description of operations through operation elements; a bindings element is used to associate interfaces to particular communication protocols through port elements; and service elements are used to expose the supported combinations of interfaces and bindings.

Deleted: Web Service

In the current scheme, there is no way for a Web services architect to attach some Quality of Service related information to the description of ports. The advertised QoS characteristics may refer to capabilities of a particular port or they may define communication requirements. For example, a Web service may require the use of a WS-Transaction SOAP actor for a particular port. WSDL has no means of advertising this requirement. Web service consumers may attempt to communicate with the particular port of the Web service only to receive a SOAP fault. Equally, a consumer may not be aware that a Web service supports WS-SecureConversation for a particular port and, hence, may not utilise the capability, though it exists, to the detriment of overall application performance.

Deleted: Web Service

Deleted: Web Service

Deleted: for

Deleted: Web Service

Deleted: Web Service

Deleted: Also

Deleted: Web Service

Deleted: , not utilising

Deleted: .

3. Proposal

We propose the introduction of two new WSDL elements as children of the [port] information item: **[requires]** and **[supports]**. Their structure is defined in the following XML schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.w3.org/????/??/wsdl"
  xmlns:wsdl="http://www.w3.org/????/??/wsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:complexType name="CapabilitiesType">
    <xs:sequence>
      <xs:any processContents="lax" namespace="##other" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="requires" type="wsdl:CapabilitiesType"/>
  <xs:element name="supports" type="wsdl:CapabilitiesType"/>
</xs:schema>
```

The `requires` and `supports` elements are hints to WSDL processors about the capabilities of ports. Their content is not defined by WSDL. Protocol specifications have to define elements (and their semantics) that can be included as contents of the `requires` and `supports` elements in WSDL documents.

4. Example

Let's assume a BankAccount Web service that is described by the following WSDL.

Deleted: Web Service

```

<wsdl:definitions
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:bank="http://example.bank.com"
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext"
  xmlns:wstx="http://schemas.xmlsoap.org/ws/2002/08/wstx"
  targetNamespace="http://example.bank.com">
  <wsdl:types>
    <xs:schema/>
  </wsdl:types>

  <wsdl:message name="creditAccountMsg">
    <wsdl:part name="accountId" type="xs:string"/>
    <wsdl:part name="ammount" type="xs:positiveInteger"/>
  </wsdl:message>

  <wsdl:interface name="BankAccountInterface">
    <wsdl:operation name="credit">
      <wsdl:input message="bank:creditAccountMsg"/>
    </wsdl:operation>
  </wsdl:interface>

  <wsdl:binding name="BASOAPBinding" type="bank:BankAccountInterface">
    <soap:binding style="document"/>
    <wsdl:operation name="bank:credit">
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>

  <wsdl:service name="BankAccountService">
    <wsdl:port name="BankAccountServicePort"
      binding="bank:BASOAPBinding">
      <soap:address location="http://example.bank.com/Service"/>
      <wsdl:supports>
        <wsse:secure-conversation>
      </wsdl:supports>
      <wsdl:requires>
        <wstx:supports
          tx-type="http://schemas.xmlsoap.org/ws/2002/08/wstx"/>
      </wsdl:requires>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Deleted: transactional

The BankAccountServicePort port is defined as supporting the use of WS-SecureConversation SOAP actor and advertises its requirement for the WS-Transaction protocol. The `wsse:secure-conversation` element and its semantics would be defined by the WS-SecureConversation specification. Similarly, the `wstx:transactional` element would be defined by the WS-Transaction specification. For example, `wstx:supports` may suggest that all the interactions with the operations supported by the port may take place within the context of an atomic transaction (specified by the `tx-type` attribute), though the operations do not mandate it. It is an easy inference that other attributes like `wstx:requires`, `wstx:requires-new` and so forth could also be used here.

It is noted that Web service architects may wish to qualify particular operations of an interface with `wsdl:supports` and `wsdl:requires` elements instead of applying them to an entire port. If such a situation is to be allowed, then these elements must be introduced as children of the [operation] infor-

Deleted: are defined

Deleted: by

Deleted: is

Deleted: transactional

Deleted: must

Formatted: Paper CodeInParagraph

Formatted: Paper CodeInParagraph

Deleted: Web Service

mation item as well. An example where such a situation may be considered is a Web service port where only a subset of operations must be executed within the context of an atomic transaction.

Deleted: Web Service

Deleted: whose

Deleted: number of