# WEB SERVICE INTEROPERABILITY AND ITS ROLES

-

We conclude our discussions on emerging Web services with some form of introduction to the works initiated by the Web Service Interoperability (WS-I [27] ) organization.

This important organization is an open -standards industry organization that is responsible for handling the interoperability issues of Web services by defining well-defined profiles for Web services interoperability. These profiles can be treated as best practice guidelines for most service interoperability concerns.

However, these profiles are not necessarily standards by themselves ; rather, these profiles are based on the existing standards, such as SOAP, WSDL, and UDDI. The term "guidelines" may be a more appropriate form of understanding.

In addition to these interoperable profile guidelines from WS-I, the organization provides tools to validate the messages to verify their conformity with the definedprofiles. As of today, this organization defined a basic profile [28] for an XML message exchange, using SOAP 1.1, WSDL 1.1, and XML schema as the core standards. Since this is appearing to maintain its form as the major interoperability guideline for Web service messages, we will spend some time in reviewing these details related to these best practices in order to better understand how they are defined.

**Introduction to Basic Profile Guidelines**

The current released version of basic profile is version 1.0. The WS-I board already accepted this Web service basic profile 1.0 as a candidate recommendation. This basicprofile 1.0 is built upon the following specifications:

SOAP 1.1, including material related to

Namespaces in XML

XML 1.0 Second Edition

WSDL 1.1, including material related to

XML Schema 2001 Part 1: Structures

XML Schema 2001 Part 2: Datatypes

UDDI 2.0 (which includes support for UDDI 1.0 interfaces)

SOAP Messages with Attachments, W3C note, 11 December 2000

It is important to understand the basic profile version and its base standards, including their respective versions. Based upon prior discussion, we are aware that these standards are evolving. The best practice usage model may, therefore, differ regarding combinations of these standards. The actual scope of the profile is bound by the standards it depends upon. It should be further understood that at development and test time, one may need to select a profile that matches the environment. Let us now begin to review more of the details surrounding this topic.

**Some Details on the Basic Profile, with Samples**

The basic premise of this profile is to ensure that Web services are able to communicate. The basic profile does not define any application semantics; rather, it ensures that the messages can be exchanged between any requester and the provider. The requirements are numbered (e.g., R2001) such that the validation tools and readers can refer to them as appropriate.

The basic profile provides interoperability in the following areas.

**Messaging**

The messaging aspect of Web service requires the most attention in order to achieve the required level of interoperability.

Web service messages are specified as XML messages. These XML messages are packaged into the SOAP envelope, and then transported through SOAP nodes to the final end-point destination. These messages may go through a number of intermediaries and processors.

This complexity requires a high level of interoperability among all the parties involved. The basic profile defines some core requirements around this message exchange pattern, including XML representation of SOAP messages, encoding requirements, SOAP processing model requirements, SOAP fault handling semantics, and SOAP binding over transport protocols.

The profile prefers the use of "literal" (nonencoded) XML messages rather than "encoded" XML messages. If we are using encoding, we must specify a soap:encodingStyle attribute to specify the required encoding schemes, including any restrictions such as "encodingStyle" which must not be present in <soap:envelope>, and in the child elements of the <soap:body>.

The "soap:mustUnderstand" attribute must accept only a "0" or a "1" value. The aforementioned sample validation rule helps us understand the depth and details of theprofile. We will review a real validation using the profile tool later in the book in order to understand the details.

**Service Description**

The profile defines the best usable scenarios for WSDL to describe interoperable messages. This section defines a number of correct usage scenarios for WSDL. This is to be considered an extended set of requirements.

**WSDL Document Structure**

This section covers a wide number of requirements on the possible construction of a WSDL document as a whole. This includes document validation, importing other schemas, encoding, placement of elements, and namespace management. The following example rules explain the correct usage patterns.

The R2007 requirement specifies that the <wsdl:import> must specify a location attribute. It must not be empty.

The above example relates to the importing of other XML schemas from other locations. The next example specifies the requirements on the above import statement placement.

The R2022 requirement defines that the <wsdl:import> elements must precede all other elements from the WSDL namespace, except <wsdl:documentation>.

There are a number of such mandatory, optional, and required rules on the document structure.

### Best Practices Usage of WSDL Types, Messages, portTypes, and Bindings

There are a lot of mandatory and optional rules that have been specified in this section related to the areas of WSDL portType definitions, message usage, and message type construction. The binding area covers the best usage pattern over different bindings, such as HTTP.

An incorrect example of a declaration for <wsdl:message> is shown below:

```
<message name="GetTradePriceInput"> <part name="tickerSymbol"
element="xsd:string"/> <part name="time" element="xsd:timeInstant"/> </message>
```

or, another form of the same declaration is shown below:

```
<message name="GetTradePriceInput"> <part name="tickerSymbol"
element="xsd:string"/> </message>
```

As mentioned, the above <wsdl:message> usage is incorrect, even though it is valid with reference to WSDL. Based on R2206, a wsdl:message in a <wsdl:description> containing a wsdl:part that uses the element attribute must refer to a global element declaration.

The correct implementation of the declaration is shown below:

```
<types xmlns:test="http://example.org/test/">> <xsd:element
name="SubscribeToQuotes" type="test:SubscribeToQuotesType" /> </types> <message
name="GetTradePriceInput"> <part name="body" element="test:SubscribeToQuotes"/>
</message>
```

WSDL 1.1 has not clearly specified the child element structure in the case of a document-literal approach. The basic profile avoids confusion by clearly mandating therequirements on the <soap:body> message. The child element of <soap:body> must be an instance of the global element declaration referenced by the corresponding <wsdl:message> part. Based upon the above declaration of the "SubscribeToQuotes" message, the correct SOAP message must be constructed as shown below:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:test=" http://example.org/test/"> <s:Header/> <s:Body> <test: SubscribeToQuotes xmlns:test=" http://example.org/test/"> ............... </test: SubscribeToQuotes> </s:Body> </s:Envelope>

Similar to the above examples, there are about 60 “70 requirements listed in the basicprofile to clarify the WSDL description mechanism.

### Use of XML Schema

WSDL 1.1 uses the XML schema as one of its type systems. The basic profile mandates the use of XML schema in accordance with the XML schema 1.0 recommendation.

### Service Publication and Discovery

The use of registries in Web service is an optional feature. The most common registry suitable for Web service publication is Universal Description Discovery and Integration (UDDI). [29] The basic profile lists some common rules for this publication; however, the usage model is not (yet) always compatible with UDDI 2.0.

### Security

The basic profile incorporates the networking services transport-level securities, such as HTTPS. The requirement is simple, in that it requires the SOAP address to specify an " https " address, rather than normal "http." The rest of the processing is a binding-level functionality.

### Tools

The tools play an important role in the basic profile validation. The basic profile requirements are complex for a service developer to verify, related to its accuracy. The tools should help in each stage of the development effort. The service developer can validate the WSDL for interoperability concerns. Furthermore, runtime efforts should provide for the facilities to introspect the XML messages, through the wire, for interoperability concerns, and accordingly advise as to any situations encountered in this stage.

These tools provide some interesting capabilities:

They validate the XML SOAP message against its corresponding WSDL and schema.

They log the response on validation in a conformance report for review (normally an HTML file).

On review we can determine each of the tests and validation results.

If there is a validation failure or a need for some explanation or any recommendation, they may point us to the corresponding basic profile recommendation.

The current specification for tools defines two important aspects.

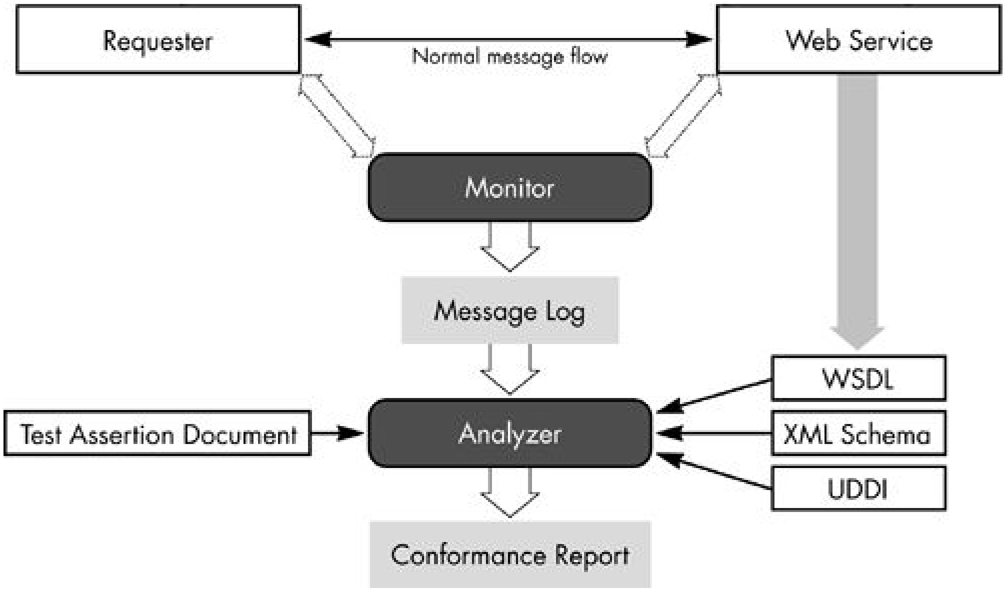
**Monitoring SOAP messages**

This allows tools to monitor the SOAP messages from a requester to a Web service. These SOAP messages are redirected to another port through tunneling mechanisms. As we can see, these monitors output the data in format as needed by the analyzer.

**Analyzing SOAP messages**

These tools are responsible for analyzing the logged messages in conjunction with the service WSDL, XML schema, and UDDI profiles. These messages are validated against the test assertion document, and finally a conformance report is produced. These types of analytical tools provide powerful validation capabilities when properly combined with these basic profile recommendations. This overall tools architecture is depicted in Figure 4.4.

Figure 4.4. The basic profile validation tools.



**Security Profiles and Other Areas to Consider**

In addition to the basic profile we have just discussed, there are other important works underway throughout the global technical communities to define a common interoperable security profile that matches the WS-Security standards. This work is also intended to define other testing tools to validate secure message exchange interactions. Another notable area is the development of sample applications and their usage scenarios.

We recommend using WS-I-defined tools to validate the XML message for better interoperability. In the coming years , we will most likely witness that many of the existing Web service vendors will be incorporating these tools in their products for the purpose of testing interoperability.

Source : http://elearningatria.files.wordpress.com/2013/10/ise-viii-grid-computing-06is845-notes.pdf