



Web Services Description Requirements

W3C Working Draft 28 October 2002

This version:

<http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028>

Latest version:

<http://www.w3.org/TR/ws-desc-reqs>

Previous version:

<http://www.w3.org/TR/2002/WD-ws-desc-reqs-20020429>

Editor:

Jeffrey C. Schlimmer, Microsoft

This document is also available in these non-normative formats: XML, PS, PDF, and TXT.

Copyright © 2002 W3C® (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use, and software licensing rules apply.

Abstract

This document describes the Web Services Description Working Group's requirements for the Web Services Description specification.

Status of this Document

This is a W3C Last Call Working Draft of the Web Services Description Requirements document. It is a chartered deliverable of the Web Services Description Working Group (WG), which is part of the Web Services Activity. This document represents the current consensus within the Working Group about Web Services Description requirements. The Working Group does not intend to take this document further than Last Call, except to update this document in response to comments and requests from other Working Groups and the public.

The Last Call review period ends on 31 December 2002. Comments on this document should be sent to public-ws-desc-comments@w3.org (public archive). It is inappropriate to send discussion emails to this address.

Discussion of this document takes place on the public www-ws-desc@w3.org mailing list (public archive) per the email communication rules in the Web Services Description Working Group Charter.

Patent disclosures relevant to this specification may be found on the Working Group's patent disclosure page.

This is a public W3C Working Draft. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". A list of all W3C technical reports can be found at <http://www.w3.org/TR>.

Table of Contents

- 1 Notations [p.2]
- 2 Definitions [p.3]
 - 2.1 Non-normative definitions [p.3]
 - 2.2 Normative definitions [p.4]
- 3 Relationship to WG Charter [p.4]
- 4 Requirements [p.5]
 - 4.1 General [p.5]
 - 4.2 Simplicity [p.7]
 - 4.3 Interface Description [p.8]
 - 4.4 Description of Interactions with a Service [p.11]
 - 4.5 Messages and Types [p.12]
 - 4.6 Service Types [p.14]
 - 4.7 InterfaceBindings [p.15]
 - 4.8 Reusability [p.19]
 - 4.9 Extensibility [p.19]
 - 4.10 Versioning [p.20]
 - 4.11 Security [p.21]
 - 4.12 Mapping to the Semantic Web [p.21]
- 5 Requirements from other W3C WGs [p.21]
 - 5.1 XML Protocol [p.22]
 - 5.2 XForms [p.22]
 - 5.3 RDF [p.22]
 - 5.4 P3P [p.22]

Appendices

- A References [p.22]
- B Acknowledgments [p.23] (Non-Normative)

1 Notations

The following terminology and typographical conventions have been used in this document.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted in a manner similar to that described in [IETF RFC 2119] [p.22] . (Changes from [IETF RFC 2119] [p.22] are indicated with *emphasis*.)

MUST, REQUIRED, SHALL

The *requirement* is an absolute requirement. The specification *produced by the WG must address this requirement*.

SHOULD, RECOMMENDED

There may exist valid reasons *for the WG* to ignore *this requirement*, but the implications of *doing so* must be understood and weighed before doing so.

MAY, OPTIONAL

The *requirement* is truly optional. *The WG* may choose to omit the *requirement for the sake of scope or schedule*.

For the sake of process and clarity, each requirement is annotated with meta data.

- Each requirement has an identification number. The numbers are arbitrary and do not imply any ordering or significance.
- Draft requirements are annotated to indicate their review status within the WG:

[Draft]

A candidate requirement the WG is actively considering but has *not* yet reached consensus on.

- To indicate their source, requirements may be annotated with the initials of the original submitter, 'Charter' (from [WSD Charter] [p.23]), or 'WG' (from WG discussion).

2 Definitions

The definitions in this section are drawn primarily from [WSDL 1.1] [p.23] and are intended to be used for purposes of discussion. They are not intended to constrain the results of the WG.

2.1 Non-normative definitions

Web Service

[Definition: A **Web Service** is a software application identified by a URI [IETF RFC 2396] [p.22] , whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols.]

Client

[Definition: A **Client** is a software that makes use of a Web Service [p.3] , acting as its 'user' or 'customer'.]

2.2 Normative definitions

Message

[Definition: A **Message** is the basic unit of communication between a Web Service [p.3] and a Client [p.4] ; data to be communicated to or from a Web Service as a single logical transmission.]

Operation

[Definition: A sequence of Messages [p.4] related to a single Web Service [p.3] action is called an **Operation**.]

Interface (AKA Port Type)

[Definition: A logical grouping of operations [p.4] . An **Interface** represents an abstract Web Service [p.3] type, independent of transmission protocol and data format.]

InterfaceBinding

[Definition: An association between an Interface [p.4] , a concrete protocol and/or a data format. An **InterfaceBinding** specifies the protocol and/or data format to be used in transmitting Messages [p.4] defined by the associated Interface.]

EndPoint (AKA Port)

[Definition: An association between a fully-specified InterfaceBinding [p.4] and a network address, specified by a URI [IETF RFC 2396] [p.22] , that may be used to communicate with an instance of a Web Service [p.3] . An **EndPoint** indicates a specific location for accessing a Web Service using a specific protocol and data format.]

Service

[Definition: A collection of EndPoints [p.4] is called **Service**.]

3 Relationship to WG Charter

The Web Services Description WG Charter [WSD Charter] [p.23] has two sections describing what is in-scope and what is out-of-scope of the problem space defined for the WG. The WG considers all the requirements in Section 1 of [WSD Charter] [p.23] to be in-scope per the Charter.

Reviewers and readers should be familiar with the Web Services Description WG Charter [WSD Charter] [p.23] because it provides the critical context for the requirements and any discussion of them.

4 Requirements

4.1 General

R001

The description language **MUST** allow any programming model, transport, or protocol for communication between peers. (From the Charter. Last revised 23 Apr 2002.)

R004

The WG specification(s) **MUST** describe constructs using the [XML Information Set] [p.23] model (similar to the SOAP 1.2 specifications [SOAP 1.2 Part 1] [p.22]). (From JS. Last revised 21 Feb 2002.)

R099

Processors of the description language **MUST** support XML Schema (<http://www.w3.org/2001/XMLSchema>). See also [XML Schema Part 1] [p.23] . (From WG discussion. Last discussed 21 Feb 2002.)

R100

The description language **MUST** allow other type systems besides XML Schema (<http://www.w3.org/2001/XMLSchema>) via extensibility. (From WG discussion. Last discussed 21 Feb 2002.)

R098

The WG specification(s) schema and examples **MUST** be written in XML Schema and **SHOULD** be written in the latest public W3C XML Schema Recommendation. (From WG discussion. Last revised 28 Feb 2002.)

R005

The WG specification(s) **MUST** correct errors/inconsistencies in [WSDL 1.1] [p.23] . (From KL. Last revised 10 Apr 2002.)

R007

The WG specification(s) **MUST** provide detailed examples, including on-the-wire messages. (From KL. Last revised 10 Apr 2002.)

R003

The WG specification(s) **SHOULD** use available XML technologies. (From JS. Last revised 10 Apr 2002.)

R105

The WG specification(s) SHOULD support Web Services that operate on resource constrained devices. (From YF. Last discussed 10 Apr 2002.)

R010

The WG specification(s) SHOULD use consistent terminology across all sections of the specification(s). (From KL. Last revised 10 Apr 2002.)

R124

The WG MUST register a MIME type for WSDL (perhaps application/wsdl+xml). (From WG discussion. Last revised 27 Jun 2002.)

R006

[Rejected, KL] Provide better specification for document name and linking. WSDL 1.1 Section 2.1.1 is over simple. More detailed specification should be provided to define how the import mechanism works, especially how it is related to the import and include mechanism defined in the XML Schema specification [XML Schema Part 1] [p.23] . (Last revised 10 Apr 2002. Redundant with R005, don't need each individual issue listed in the requirements doc. The WG already has two issues in its issues document for clarifying import, and adding include.)

R009

[Rejected, KL] Enable easy Interaction with Upper layers in the Web Services stack. Additional technologies will be required in the future to complete the Web Services architecture. As one of the fundamental layers of the Web Services stack, though WSDL should not depend on any other layers, one of the design goals of WSDL should be easy interaction with upper layers, such as Services composition layers. (Last revised 10 Apr 2002. Success is not measurable.)

R103

[Rejected, YF] WSDL specifications should be clear and easy to understand. This clarity implies that considerable editorial effort will be required in the structuring of the narrative through both outline/overview and normative reference material. (Last revised 10 Apr 2002. A specification should be precise. Clear and easy to understand are both very subjective)

R008

[Rejected, KL] Support up-to-date XML Schema. In all [WSDL 1.1] [p.23] examples, the October 2000 version of the XML schema is used: <http://www.w3.org/2000/10/XMLSchema>. We understand that the 10/2000 schema was the most up-to-dated schema available at the time WSDL1.1 was released. However, in future versions of WSDL specification, the W3C Recommendation version of the XML schema should be used. The recommendation was released in May 2001 [XML Schema Part 1] [p.23] : <http://www.w3.org/2001/XMLSchema>. (Last discussed 21 Feb 2002. Replaced with R098, R099, and R100.)

4.2 Simplicity

R013

The WG specification(s) **MUST** be simple to understand and implement correctly. The description language **MUST** be simple to use. (From the Charter. Last discussed 7 Mar 2002.)

R014

The WG specification(s) **SHOULD** be compatible with existing Web infrastructure. (From the Charter. Last discussed 7 Mar 2002.)

R011

[Rejected, Charter] Focus must be put on simplicity, modularity and decentralization. (Last discussed 21 Feb 2002. Replaced with R013, R102, R027.)

R016

[Rejected, JS] Be simple to understand and implement correctly; comparable to other widespread Web solutions. (Last discussed 21 Feb 2002. Replaced with R013.)

R017

[Rejected, JS] Specification shall be as lightweight as possible, keeping parts that are mandatory to a minimum. (Last discussed 7 Mar 2002. Covered by R013.)

R018

[Rejected, JS] Optional parts of the specification should be orthogonal to each other allowing non-conflicting configurations to be implemented. (Last discussed 7 Mar 2002. Good goal, but unnecessary as a specific requirement.)

R019

[Rejected, YF] Facilitate the creation of simple applications (fast and easy writing for simple apps). (Last discussed 7 Mar 2002. Merged in R013.)

R020

[Rejected, YF] Be possible to compare easily two WSDL Web Services. (Last discussed 7 Mar 2002. May raise intractable semantic issues.)

R102

[Rejected, YF] Since WSDL is intended to be a foundation service description language, its definition should remain simple and stable over time. Explicit use of modularity and layering in the resulting design will help assure longevity. Such a framework will allow subsequent extension of the design while leaving the foundation of the design intact. (Last discussed 7 Mar 2002. Adequately

covered by 'simple' in R013.)

R104

[Rejected, YF] The WSDL specification must clearly identify conformance requirements in a way that enables the conformance of an implementation of the specification to be tested (see also the W3C Conformance requirements (W3C members only)). (Last discussed 7 Mar 2002. Adequately covered by 'correct' in R013.)

4.3 Interface Description

R021

The description language MUST describe the Messages accepted and generated by the Web Service. (From the Charter. Last revised 21 Feb 2002.)

R022

The description language MUST allow describing application-level error Messages (AKA faults) generated by the Web Service. (From the Charter. Last revised 28 Feb 2002.)

R054

The description language MUST describe Messages independent from their use in message exchange patterns and/or InterfaceBindings. (From YF. Last revised 17 Oct 2002.)

R041

The description language MUST allow describing sets of Operations that form a logical group. (From JS. Last revised 28 Feb 2002.)

R116

The description language MUST allow describing abstract policies required or offered by Services. (From GD. Last revised 11 Apr 2002.)

R083

The description language MUST separate design-time from run-time information. (From JS. Last discussed 11 Apr 2002.)

R026

The description language MUST provide human-readable comment capabilities. (From the Charter. Last discussed 28 Feb 2002.)

R123

The content model for human-readable comment capabilities **MUST** be open. (From RD. Last discussed 11 June 2002.)

R042

The description language **SHOULD** allow deriving one Interface from another by extension of the logical group of Messages. (From JS. Last discussed 11 June 2002.)

R117

The description language **SHOULD** allow specifying QoS-like policies and mechanisms of a Web Service. For instance, an indication of how long it is going to take a Web Service to process the request. (From WG discussion. Last discussed 12 April 2002.)

R109

[Rejected, JS] The language must describe Interfaces separate from their concrete protocol, transport, data format or wire format deployment. (See also R046.) (Last discussed 7 Mar 2002. Covered by R071. ?I think we wrote this to respond to the partition description across multiple files (R071) but then discarded the other requirement (described in the wording of this requirement) that underlies the definition of an Interface versus an InterfaceBinding?)

R032

[Rejected, WS] In a lot of cases, it is important for the server to expose some service-wide properties/attributes. These properties/attributes have the service-level scope and could be used to describe either some QoS parameters or some application specific characteristics. As an example, a service may want to expose an attribute which describes the version number of the service. Hence, WSDL should be able to model service level attributes/properties. (Last discussed 11 April 2002. Covered by R117, R116, R075.)

R112

[Rejected, SK] A Web Service description should be able to define extensible mechanisms for capturing meta-information associated with a message. A WS description allows it to publish the message interactions it is capable of handling. However, this description alone does not capture any meta-information associated with the message interaction definitions. The message interactions are meaningful in a given business domain, or more precisely, as defined as part of W3C work on ontology. Some of the examples of the meta-information are:

1. Some messages of a WS may require authentication information.
2. Some messages of a WS may deal with in a particular Business Domain. For instance, submitPO, may be an overloaded message where one such message primarily deals with RosettaNet.

3. QoS parameters

(Last discussed 11 April 2002. Covered by R117, R116, and others.)

R035

[Rejected, KL] Distinction between interface definition and implementation definition. A description of a Web Service can be logically divided into three parts: Data type definition, Service Interface definition and Service Implementation definition. The data type definition can be viewed as part of the Service Interface Definition. Analogous to defining an abstract interface in a programming language and having many concrete implementations, a service interface definition can be instantiated and referenced by multiple service implementers. [WSDL 1.1] [p.23] specification implies such a division by providing the mechanism for dividing a service definition into multiple WSDL documents. WSDL1.1 Section 2.1.2, Authoring Style, shows an example of separating a complete service definition into three documents: data type definition, abstract definitions and specific service bindings. However, this distinction is not clear and reference to each unit is very difficult. To facilitate easier allocation of responsibilities among different organizations (such as standard bodies and service providers) or among different teams within an organization (such as teams related to the different stages of a service's life-cycle: design time/development time, configuration time and run time), a better distinction between Interface definition and Implementation definition should be made in the specification. Elements such as Message, PortType, Operation are abstract interface definitions, and are usually defined at design time. Elements such as InterfaceBinding and Services usually get their value at configuration/deployment/run time. Mixing all these elements together is at least confusing to many people. (Last discussed 11 April 2002. Covered by R083.)

R089

[Rejected, KB] Describe Web Services Operations in an abstract format using the XML type system. (Last discussed 11 April, 2002. Covered by R099.)

R090

[Rejected, KB] Group logically related Operations together into abstract Interface types. (Last discussed 11 April, 2002. Covered by R041.)

R023

[Rejected, Charter] The data exchanged is usually typed and structured. This increases interoperability by having applications agreeing on semantics and also provides some level of error detection. It is expected that developers will want to use different mechanisms for describing data types and structures, depending on the purpose of the Web service. The WG should allow different mechanisms, and must define one based on XML Schema. (Last discussed 21 Feb 2002. Covered by R021, R090, R100.)

R033

[Rejected, YF] Support abstract interfaces. (Last discussed 28 Feb 2002. Replaced by R109.)

R034

[Rejected, YF] Support interfaces derived from abstract interfaces. (Last discussed 28 Feb 2002. Replaced by R109.)

R101

[Rejected, KL] The final WSDL specification should be divided into two parts: the first part only focuses on the core interface definition language, and the second part addresses the binding extensions. This requirement concurs with the Charter's requirement for two separate deliverables. (Last discussed 28 Feb 2002. Concern that this over constrains the specification process.)

4.4 Description of Interactions with a Service

R036

The description language **MUST** allow describing the functionality associated with one-way messages (to and from the service described), request-response, solicit-response, and faults. (From the Charter. Last revised 28 Feb 2002.)

R044

The description language **SHOULD** allow describing both application data and context data of a Service. (From PF. Last discussed 12 April 2002.)

R097

The description language **SHOULD** allow describing asynchronous message exchange patterns. (From IS. Last discussed 11 April 2002.)

R110

The description language **SHOULD** allow indicating how long a Web Service is going to take to process the request. This is just a hint to the Client [p.4] , and the Web Service would not be obligated to respect what it advertised. (From WV. Special case of R117.)

R094

The description language **MAY** allow describing events and output-oriented Operations. The description language **MAY** be very specific about events, defining a special type of a Message or even a separate definition entity. (From IS. Last discussed 12 April 2002.)

R040

[Rejected, JS] Describe arbitrary Message exchanges. (Last discussed 11 April 2002. Out of scope.)

R045

[Rejected, PF] WSDL is typically used to capture the Web Server requirements on the Client [p.4] . For example, the Web Server will expect to see certain SOAP headers. When WSDL is used in higher protocols, such as an orchestration language, each side of the exchange may wish to publish their requirements, and the Client [p.4] may have a requirement on the Web Server. For example, the Client [p.4] may require the Web Server to set a particular header on the response. In WSDL today, there is an option to try to map this into the 'out-in' or 'out' interactions, by treating them as the 'conjugates' of the corresponding 'in-out' or 'in-only' Operations. However, this is unsatisfactory, as these interactions are not well defined, and there is no way to specify that an out-in is actually the conjugate of an in-out, or simply another Operation that has the same messages in the opposite order. It would be more satisfactory if the concept of 'conjugates' was exposed directly so that the Client [p.4] side of an interaction could publish their requirements. This could be used by proposal such as flow or orchestration languages. (Last discussed 12 April 2002. Out of scope as a feature - move to use cases.)

R037

[Rejected, JJM] Must describe SOAP 1.2 MEP (Message Exchange Pattern) (charter says: "must [...] describe [...] one-way Messages, [...] request-response") (Last discussed 28 Feb 2002. Covered by R036.)

R038

[Rejected, JS] Must be able to describe simple one-way Messages, i.e., either incoming or outgoing (event) Messages. (Last discussed 28 Feb 2002. Covered by R036.)

R039

[Rejected, JS] Must be able to describe simple request-response-fault Message exchange. (Last discussed 28 Feb 2002. Covered by R036.)

R122

The description language MAY allow restricting and/or describing the possible flow of Messages between the Web Service and a Client. The description language MAY in particular allow describing what applicative Fault refers to what incorrect call flow. (Last discussed 11 June 2002. Beyond WG scope.)

4.5 Messages and Types

R046

The description language MUST describe Messages independent from transfer encodings. (From JS. Last discussed 17 Oct 2002.)

R085

The description language SHOULD allow describing Messages that include references (URIs) to typed referents, both values and Services. (From PP. Last discussed 11 April, 2002.)

R051

[Rejected, JS] Be able to describe Messages that include arrays and nested arrays. (Last discussed 11 April 2002. Subsumed by R100.)

R047

[Rejected, JS] Be able to describe the semantic content of messages. (Last revised 11 April 2002. Out of scope.)

R096

[Rejected, IS] Be able to describe references to other Web Services (remote) or other Interfaces (EndPoints, local to this WSDL doc) that can be used as parts in Message definitions. Currently (as of [WSDL 1.1] [p.23]) Message parts refer to data types (described in one or the other schema). The part must also be able to refer to a remote Web Service (WSDL URL/Service/Port) or a local Web Service/EndPoint qualified names. This has to be made clear as part of the standard for WS Client [p.4] s and Web Service providers. (Last discussed 11 April 2002, covered by R085.)

R055

[Rejected, YF] Support grouping functionalities (Operations) that share the same Message-exchange pattern and transport InterfaceBinding. (Last discussed 11 April, 2002. Unclear what problem this "solution" is targeted at.)

R053

[Rejected, JR] Be able to classify/categorize [individual] Operations. With the usage of XML schema in the ELEMENT attribute of the PART element (current WSDL spec), it is possible to use a type system as a kind of taxonomy for a semantically enriched description of parameters. To automatically search a suitable Web Service respectively Operation from a set of Web Service descriptions, it is not enough only to consider the parameters but also a kind of Operation "type" (something like a taxonomy on Operations). So I would suggest a kind of ELEMENT or TYPE attribute for Operations. (Last discussed 11 April 2002. Out of scope.)

R093

[Rejected, IS] Be able to accommodate namespace clusters with data types (schemas) and Interface definitions (Message / EndPoint / InterfaceBinding). I.e., Service may have several namespaces with types and several other namespaces with Message/EndPoint definitions. That is pretty important for expressing proper OO model of a Service. Very few framework implementations pay attention to this. (In many cases namespaces are flattened out which results in name conflicts.) I guess it is so because namespaces of various type definitions and Message / EndPoint / InterfaceBinding definitions have never been emphasized as a requirement really. (Last discussed 11 April, 2002. This

requirement seems to be addressed to poor/incomplete implementations of namespaces.)

R048

[Rejected, JS] Must be able to describe Messages using XML Schema simple and complex types. (Last discussed 11 April 2002. Covered by R099.)

R049

[Rejected, JS] Be able to describe Messages using other info sets. (Last discussed 11 April, 2002. Covered by R100.)

4.6 Service Types

R118

The description language SHOULD group Interfaces into a Service type. (From JS. Last discussed 12 April 2002.)

R058

The description language SHOULD allow deriving one Service type from another by extension of the logical group of InterfaceBindings. (From JS. Last discussed 12 April 2002.)

R106

[Rejected, PM] Ability to associate a network address with an InterfaceBinding at runtime. For example, it is possible to have a Interface that supports Operations like "Register" and "Notify" where a user will provide an email address that a Web Service can send notifications to when the user registers with the Service. So the network address for the "Notify" Operation needs to be dynamically populated at runtime. (Last discussed 12 April 2002, Covered by R083 and R085, move to use cases.)

R057

[Rejected, JS] Be able to name an instance of a EndPoint independent of its address. (Last discussed 12 April 2002. Needs clarification.)

R056

[Rejected, JS] Be able to describe a logical group of fully-specified InterfaceBindings without specifying a network address that may be used to communicate with the instance of the InterfaceBinding. That is, be able to describe a Service type. (Prescribes a specific means to fulfill R106.) (Last discussed 12 April 2002, probably covered by R118.)

4.7 InterfaceBindings

R081

The description language MUST describe EndPoint location using URIs. (From JS.)

R114

The description language MUST allow unambiguously mapping any on-the-wire Message to an Operation. (From WG discussion. Last revised 4 Apr 2002.)

R060

The description language MUST allow specifying an association between an Interface and one or more concrete protocols and/or data formats. (From the Charter. Last revised 12 Apr 2002.)

R068

The description language MUST allow binding of transport characteristics independently of data marshalling characteristics. (From PF. Last discussed 12 April 2002.)

R052

The description language MUST allow describing InterfaceBindings to other protocols besides those described in the specification. (From JS. Last revised 11 April 2002.)

R111

The WG MUST provide a normative description of the InterfaceBinding for HTTP/1.1 [IETF RFC 2616] [p.22] GET and POST. (From the Charter. Last revised 28 Mar 2002.)

R066

The description language MUST allow binding Interfaces to transports other than HTTP/1.1 [IETF RFC 2616] [p.22] . (From JS. Last discussed 12 April 2002.)

R028

The description language MUST allow describing the structure of incoming and outgoing SOAP 1.2 messages [SOAP 1.2 Part 1] [p.22] , including the contents, encoding, target, and optionality of SOAP 1.2 Header and Body blocks, SOAP RPC blocks, and SOAP Faults. (From JJM. Last revised 12 Apr 2002.)

R113

The description language MUST allow describing which SOAP features are offered by or required by an Operation or a Service. (From GD. Last revised 4 Apr 2002.)

R065

The WG MUST provide a normative description of the InterfaceBinding for SOAP 1.2 over HTTP/1.1. (From JS. Last revised 28 Mar 2002.)

R062

The WG specification(s) MUST ensure that the SOAP 1.2 InterfaceBinding is capable of describing transports other than HTTP. (From the Charter. Last revised 28 Mar 2002.)

R125

The normative description of the InterfaceBinding for SOAP 1.2 MUST support the SOAP 1.2 MEP for HTTP GET in and HTTP SOAP out. (From TAG. Last discussed 26 Sep 2002.)

R031

The WG specification(s) SHOULD support SOAP 1.2 intermediaries. (From JJM. Last discussed 11 April 2002.)

R025

[Rejected, Charter] The WG will make sure that SOAP 1.2 extensibility mechanism can be expressed. (Last discussed 11 April 2002. Covered by R113.)

R107

[Rejected, JJ] Based on the XML Protocol Usage Scenario (2.14 S21 Incremental parsing/processing of SOAP messages) and other requirements (a SOAP processor returning a large amount of data as attachment or message) there is a need for a SOAP processor and the SOAP client proxies to be constructed with the notion of data streaming in mind so that applications can scale well. (Especially in the case of dynamic proxy and stub creation scenarios.) This requirement for the SOAP processors imposed a requirement on the WSDL to be descriptive enough (like MIME binding or some kind of extension) to describe so that the Service Provider will do incremental parsing and processing of data (input) and the client can process the return message or attachment the same way. Without this description most of the toolkits will find it difficult to use this SOAP processor advantages for scalability and/or fail in interoperability. (Last discussed 12 April 2002. Covered by R117.)

R082

[Rejected, JS] Be able to describe the address for specific EndPoint instances within a Service. (Last discussed 12 April. Covered by R081.)

R086

[Rejected, PP] Support all HTTP methods (verbs), including WebDAV and allow the use of non-standard HTTP methods. (Last discussed 12 April 2002. Out of Scope.)

R029

[Rejected, JJM] Describe SOAP 1.2 Header and Body's content type. (Charter says: "must define [a mechanism for describing data types and structures] based on XML Schema" and "take into account ending work going on in XML Protocol".) (Last discussed 28 Mar 2002. Covered adequately by R028.)

R030

[Rejected, JJM] Describe SOAP 1.2 RPC parameters types (ibid.). (Last discussed 28 Mar 2002. Duplicate of R028.)

R061

[Rejected, Charter] It is expected that in the near-term future, Web Services will be accessed largely through SOAP Version 1.2 [SOAP 1.2 Part 1] [p.22] (the XML-based protocol produced by the XML Protocol Working Group) carried over HTTP/1.1 [IETF RFC 2616] [p.22] , or by means of simple HTTP/1.1 GET and POST requests. Therefore, (a) the WG will provide a normative InterfaceBinding for SOAP Version 1.2 over HTTP, and (b) the WG should provide a normative InterfaceBinding for HTTP/1.1 GET and POST requests. (Last discussed 28 Mar 2002. Covered by R065 and R111, respectively.)

R063

[Rejected, JJM] Ensure that SOAP 1.2 bindings to SMTP or BEEP (for example) can be described. (Charter says: "ensure that other SOAP bindings can be described".) (Last discussed 28 Mar 2002. Adequately covered by R062.)

R064

[Rejected, JS] Be able to describe the wire format of Messages, including, but not limited to, XML, ASCII, binary, or some combination. (Last discussed 28 Mar 2002. Out of scope; should unambiguously refer to wire format but not describe wire format per se.)

R069

[Rejected, KL] Better Specification for InterfaceBinding Extensions. In addition to the core service definition framework, [WSDL 1.1] [p.23] introduces specific InterfaceBinding extensions for SOAP 1.1, HTTP GET/POST, and MIME, and nothing precludes the use of other InterfaceBinding extensions. To keep the core service definition framework simple, a separate and more detailed specification or technical report should be dedicated for various InterfaceBindings. (Last discussed 28 Mar 2002. Technical requirement merged into R066; editorial prescription over constrains the specification process.)

R077

[Rejected, JS] Be able to describe SOAP 1.2 Messages [SOAP 1.2 Part 1] [p.22] . (Last discussed 28 Mar 2002. Covered by R028.)

R078

[Rejected, JS] The WG will provide a normative description of SOAP 1.2 Messages. (Last discussed 28 Mar 2002. Covered by R065.)

R079

[Rejected, JS] Be able to describe SOAP 1.2 Header elements and Body elements. (Last discussed 28 Mar 2002. Covered by R028.)

R080

[Rejected, JS] Be able to describe SOAP 1.2 Faults. (Last discussed 28 Mar 2002. Covered by R028.)

R087

[Rejected, FC] [WSDL 1.1] [p.23] defines services and operations and their bindings to various protocols. However, the details of how an operation is identified (either generally or specifically in particular bindings) is, shall we say, rather vague. As a result, some implementations use the namespace & element of the first child of Body (in SOAP RPC), others use SOAPAction header (in SOAP over HTTP), others use only the namespace, others the element name, others attempt to match the message type, etc. As a result, interoperability suffers.

It seems like a normative model (at least) for operation determination is necessary for interoperability between clients and servers from different vendors. This may be a requirement to define such a requirement for all defined bindings, as opposed to something that can be completely specified in the description. But I believe that such a requirement exists. (Last discussed 4 Apr 2002. Pulled out part that is not covered by R065 into R114.)

R091

[Rejected, KB] Apply specific wire-format serializations (InterfaceBindings) for Service types. (Last discussed 4 Apr 2002. Covered by R065, R111, and R067.)

R092

[Rejected, KB] Apply in an orthogonal manner specific transport(s) for an InterfaceBinding. (Last discussed 4 Apr 2002. Confusion about the intention of this requirement; perhaps a requirement for partial InterfaceBindings?)

R108

[Rejected, MW] Must be able to describe messages that include binary data, where the binary data is transmitted efficiently. (Last discussed 4 Apr 2002. Consider this requirement to be discussing attachments, and consider attachments as part of providing a quality InterfaceBinding to SOAP per R065, R062. If there are attachments for other InterfaceBindings, then it's up to those bindings to

provide appropriate support.)

4.8 Reusability

R071

The description language **MUST** allow partitioning a description across multiple files. (From JS.)

R072

The description language **MUST** allow using a description fragment in more than one description. (From JS. Last discussed 12 April 2002.)

R073

[Rejected, YF] Support reusability of WSDL documents or parts of documents. (Last discussed 12 April 2002. Covered by R072.)

4.9 Extensibility

R012

The description language **MUST** support the kind of extensibility actually seen on the Web: disparity of document formats and protocols used to communicate, mixing of XML vocabularies using XML namespaces, development of solutions in a distributed environment without a central authority, etc. In particular, the description language **MUST** support distributed extensibility. (From the Charter. Last discussed 12 April 2002.)

R067

The description language **MUST** allow for extension in description language components, including at least message, port type, binding, and service. (From WG discussion. Last discussed 17 Oct 2002.)

R074

The description language **MUST** allow indicating whether a given extension is required or optional. (From JS. Last discussed 12 April 2002.)

R121

The description language **SHOULD** be able to be easily integrated into other markup languages. This may involve the following types of considerations: media types [IETF RFC 2046] [p.22] : which should be used for a compound type, schema wildcarding in the host markup language, containment semantics: how the interpretation of WSDL is affected by different containing elements, fragment identifiers: how references that cross namespace boundaries work. (From MB. Last discussed 11 June 2002. Beyond WG scope.)

R015

[Rejected, JJM] Must support an open content model. (Charter says: "must support distributed extensibility" and "will look into extending Interface descriptions in a decentralized fashion".) (Last discussed 12 April 2002. Prescribes a specific (but plausible) means to fulfill R012 and R067.)

R027

[Rejected, Charter] Developers are likely to want to extend the functionality of an existing Web Service. The WG will look into extending interface descriptions in a decentralized fashion, i.e., without priori agreement with the original interface designers. (Last discussed 12 April 2002. Covered by R058.)

R043

[Rejected, JS] Be able to extend Interfaces using mechanisms not explicitly identified in the spec. (Last discussed 12 April 2002. Merged into R067.)

R050

[Rejected, JS] Be able to extend Message descriptions using mechanisms not explicitly identified in the spec. (Merged into R067.)

R059

[Rejected, JS] Be able to extend Service descriptions using mechanisms not explicitly identified in the spec. (Merged into R067.)

R095

[Rejected, IS] Extensible meta definitions. Be able to include *typed* metadata attributes for any definition element: Message, Operation, Interface, InterfaceBinding, EndPoint, and Service. The attributes may also be hierarchical (i.e., defined in another namespace). (Last discussed 12 April 2002. Attributes is overly prescriptive; definition elements requirement merged in R067; use of namespaces covered by R012.)

4.10 Versioning

R075

The description language MUST allow identifying versions of Services. (From PF. Last discussed 12 April 2002.)

R119

The description language MUST allow identifying versions of descriptions. (From PF. Last discussed 12 April 2002.)

R076

[Rejected, FC] It would be good to allow for versioning of something smaller than a WSDL document. I suspect that tools vendors will "compose" these documents, and they may sometimes contain information about a number of unrelated services (or, more correctly, services that are related in ways other than application semantics (tool vendor, server location, etc)). It would be good if Web Services themselves were versioned, the Web Services being the semantic "unit" being defined. (Last discussed 12 April 2002. Duplicate of R075.)

4.11 Security

R115

The WG specification(s) SHOULD define an equivalence relation on Service descriptions. (From SW. Last discussed 17 Oct 2002.)

R084

[Rejected, JS] Compliance must not preclude building implementations that are resistant to attacks. (Last revised 10 Apr 2002. Vague.)

R088

[Rejected, DM] The specification MAY document how a WSDL document can be signed, using XMLDSig, so that a potential user of the WSDL document can establish trust in the information conveyed about the web service. (Last revised 10 Apr 2002.)

4.12 Mapping to the Semantic Web

R070

The WG specification(s) MUST allow providing a mapping from the description language to [RDF] [p.22] . (From the Charter. Last revised 11 April, 2002.)

R120

The description language MUST ensure that all conceptual elements in the description of Messages are addressable by a URI reference [IETF RFC 2396] [p.22] . (From the Semantic Web. Last discussed 11 June 2002.)

5 Requirements from other W3C WGs

These are requirements submitted by other W3C Working Groups and Activities.

R024

[Rejected, Charter] The WG will also take into account the encoding work going on in the XML Protocol Working Group. (Last discussed 11 April 2002, This is not a requirement on the specifications we produce, it is a requirement on the behavior of the Working Group.)

R002

[Rejected, JS] Coordinate with W3C XML Activity and XML Coordination Group. (Last discussed 11 April 2002, This is not a requirement on the specifications we produce, it is a requirement on the behavior of the Working Group.)

5.1 XML Protocol

5.2 XForms

5.3 RDF

5.4 P3P

A References

RDF

Resource Description Framework (RDF) Model and Syntax Specification, Ora Lassila, R. Swick, Editors. World Wide Web Consortium, 22 February 1999. This version of the Resource Description Framework Model and Syntax Recommendation is <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>. The latest version of Resource Description Framework Model and Syntax is available at <http://www.w3.org/TR/REC-rdf-syntax>.

IETF RFC 2046

Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, N. Freed, N. Borenstein Author. Internet Engineering Task Force, November 1996. Available at <http://www.ietf.org/rfc/rfc2046.txt>.

IETF RFC 2119

Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, Author. Internet Engineering Task Force, June 1999. Available at <http://www.ietf.org/rfc/rfc2119.txt>.

IETF RFC 2396

Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, Authors. Internet Engineering Task Force, August 1998. Available at <http://www.ietf.org/rfc/rfc2396.txt>.

IETF RFC 2616

Hypertext Transfer Protocol -- HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Authors. Internet Engineering Task Force, June 1999. Available at <http://www.ietf.org/rfc/rfc2616.txt>.

SOAP 1.2 Part 1

SOAP Version 1.2 Part 1: Messaging Framework, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, and H. F. Nielsen, Editors. World Wide Web Consortium, 26 June 2002. This version of the SOAP Version 1.2 Part 1 Specification is <http://www.w3.org/TR/2002/WD-soap12-part1-20020626>.

The latest version of SOAP Version 1.2 Part 1 is available at <http://www.w3.org/TR/soap12-part1>.

WSD Charter

Web Services Description Working Group Charter, J. Marsh, P. Le Hégarret. World Wide Web Consortium, 26 January 2002. Available at <http://www.w3.org/2002/01/ws-desc-charter>.

WSDL 1.1

Web Services Description Language (WSDL) 1.1, E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, Authors. World Wide Web Consortium, 15 March 2002. This version of the Web Services Description Language Specification is <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>. The latest version of Web Services Description Language is available at <http://www.w3.org/TR/wsdl>.

XML 1.0

Extensible Markup Language (XML) 1.0 (Second Edition), T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Editors. World Wide Web Consortium, 10 February 1998, revised 6 October 2000. This version of the XML 1.0 Recommendation is <http://www.w3.org/TR/2000/REC-xml-20001006>. The latest version of XML 1.0 is available at <http://www.w3.org/TR/REC-xml>.

XML Information Set

XML Information Set, J. Cowan and R. Tobin, Editors. World Wide Web Consortium, 24 October 2001. This version of the XML Information Set Recommendation is <http://www.w3.org/TR/2001/REC-xml-infoiset-20011024>. The latest version of XML Information Set is available at <http://www.w3.org/TR/xml-infoiset>.

XML Schema Part 1

XML Schema Part 1: Structures, H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, Editors. World Wide Web Consortium, 2 May 2001. This version of the XML Part 1 Recommendation is <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>. The latest version of XML Schema Part 1 is available at <http://www.w3.org/TR/xmlschema-1>.

B Acknowledgments (Non-Normative)

This document is the work of the W3C Web Services Description Working Group.

The people who have contributed to discussions on www-ws-desc@w3.org are also gratefully acknowledged.