



# WSDL 2.0

Web Services Description Language 2.0

*A New Hope?  
or The Empire Strikes Back?*

# Tony Rogers

[tony.rogers@ca.com](mailto:tony.rogers@ca.com)

Senior Architect, Development  
eTrust, CA

Co-chair, W3C WS-Description Working Group  
Editor, W3C WS-Addressing Working Group  
Co-chair, OASIS UDDI Spec Technical Committee

- History of WSDL
- WSDL 2.0 – a <description>
- WSDL 2.0 and 1.1
- The steps to a standard

# History

- WSDL 1.1 – W3C Note 2001
- WSDL 2.0
  - 2002                    initial charter
  - . 2003
  - .. 2004   re-chartered
  - ... 2005
  - .... 2006 Candidate Recommendation!

# WSDL 2.0 – the beginning

*The WS Description Working Group is chartered to describe:*

- the messages*
- the message exchange patterns*
- the protocol binding*

*for the interface, the boundary across which applications (web services and their clients) communicate*

Charter for the WS Description Working Group, 2002.

# WSDL 2.0 – the documents

- Part 0 – the Primer
- Part 1 – the Core Language
- Part 2 – Adjuncts
- SOAP 1.1 Binding
- RDF Mapping
- Media Types Note
- Alternative Schema Languages Note
- ...

# WSDL 2.0 – Basics

- description
- documentation
- types
- interface
- binding
- service

# WSDL 2.0 – description

- The root of every WSDL 2.0 document
- Defines the target namespace
  - like the target namespace for Schema
- Specifies namespace prefixes

# Example: description

```
<?xml version="1.0" encoding="utf-8" ?>  
<description xmlns="http://www.w3.org/2006/01/wsdl"  
  targetNamespace="http://greath.example.com/2004/wsdl/resSvc"  
  xmlns:tns="http://greath.example.com/2004/wsdl/resSvc" ...>  
  ...  
</description>
```

# WSDL 2.0 – documentation

- Every good language needs comments!
- <documentation> element

# Example: documentation

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<description . . . >
```

```
<documentation>
```

This document describes the GreatH Web service. Additional application-level requirements for use of this service -- beyond what WSDL 2.0 is able to describe -- are available at <http://greath.example.com/2004/reservation-documentation.html>

```
</documentation>
```

```
. . .
```

```
</description>
```

# WSDL 2.0 – types

- Defines the type of every message
  - each must be a single top-level element
  - including faults
- Source of greatest confusion:
  - `wsdl:include`
  - `wsdl:import`
  - `xs:include`
  - `xs:import` – top-level or inside `xs:schema`

# Example: types

```
<?xml version="1.0" encoding="utf-8" ?>
<description ... >
  <types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://greath.example.com/2004/schemas/resSvc"
      xmlns="http://greath.example.com/2004/schemas/resSvc">
      <xs:element name="checkAvailability" type="tCheckAvailability"/>
      <xs:complexType name="tCheckAvailability">
        <xs:sequence>
          <xs:element name="checkInDate" type="xs:date"/>
          <xs:element name="checkOutDate" type="xs:date"/>
          <xs:element name="roomType" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>
  ...
</description>
```

# WSDL 2.0 – interface

- Abstract description of functionality
- Specifies operations:
  - pattern (MEP)
  - messages (including faults)
  - style
  - “safety”

# Example: interface

```
<?xml version="1.0" encoding="utf-8" ?>
<description ... >
  <types ... />

  <interface name = "reservationInterface" >
    <fault name = "invalidDataFault" element = "ghns:invalidDataError"/>
    <operation name="opCheckAvailability"
      pattern="http://www.w3.org/2006/01/wsdl/in-out"
      style="http://www.w3.org/2006/01/wsdl/style/iri"
      wsdlx:safe = "true">
      <input messageLabel="In" element="ghns:checkAvailability" />
      <output messageLabel="Out" element="ghns:checkAvailabilityResp" />
      <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
    </operation>
  </interface>
  ...
</description>
```

# WSDL 2.0 – binding

- Depends on extensions to core WSDL
- Message format
- Protocol
  
- Mapping of SOAP MEP
- Fault code

# Example: binding

```
<?xml version="1.0" encoding="utf-8" ?>
<description ... >
  <types ... />
  <interface .../>

  <binding name="reservationSOAPBinding"
    interface="tns:reservationInterface"
    type="http://www.w3.org/2006/01/wsdl/soap"
    wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP">
    <operation ref="tns:opCheckAvailability"
      wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>
    <fault ref="tns:invalidDataFault" wsoap:code="soap:Sender"/>
  </binding>
  ...
</description>
```

# WSDL 2.0 – service

- Concrete implementation of an interface
- One service -> one interface
- Each endpoint
  - address
  - binding

# Example: service

```
<?xml version="1.0" encoding="utf-8" ?>  
<description ... >  
  <types ... />  
  <interface ... />  
  <binding ... />  
  
  <service name="reservationService"  
    interface="tns:reservationInterface">  
  
    <endpoint name="reservationEndpoint"  
      binding="tns:reservationSOAPBinding"  
      address="http://greath.example.com/2004/reservation"/>  
  
  </service>  
</description>
```

# WSDL 2.0 – the terminology

- Infoset
  - all the XML in one “lump”
- Component Model
  - all the defaults resolved
  - all the “meaning” in one “lump”
- terminology used in the standards documents

# WSDL 2.0 - extensions

- WSDL 2.0 is meant to be extended
- Predefined extension:
  - `wsdlx:safety`
- Extensions defined in protocol bindings
  
- Features and Properties

# WSDL 2.0 versus 1.1

## WSDL 1.1:

- definitions: renamed description
- types: not heavily changed, but refined
- message: removed
- operation: revamped
- port type: renamed interface, and extended
- binding: more facilities
- port: renamed endpoint, simplified
- service: restricted to single interface

# WSDL 2.0 versus 1.1

- WSDL 2.0 is simpler
- WSDL 2.0 offers more structure
- WSDL 2.0 is extensible in standard ways
  
- WSDL 1.1 can be auto-translated (mostly)

# Progress to standard

- Charter
- Working Draft/s
- Last Call/s
- Candidate Recommendation
- Implementations and Interop
- Proposed Recommendation
- W3C Recommendation

# Features at risk

- Features and Properties
- HTTP IRI interface (REST)
- MEPs
  - Robust in-only
  - In-optional out
  - Out-only
  - Robust out-only
  - Out-in
  - Out-optional in

# Summary

- History of WSDL
- WSDL 2.0 – a <description>
- WSDL 2.0 and 1.1
- The steps to a standard



# References

- <http://www.w3.org/2002/ws/desc/>  
The WS-Description Working Group page
- <http://www.w3.org/TR/wsd120-primer/>  
The WSDL 2.0 Primer
- <http://www.w3.org/TR/2001/NOTE-wsd1-20010315>  
The WSDL 1.1 NOTE