



Web Services Description Language (WSDL)

Wanasanan Thongsongkrit



WSDL Development History at W3C

- WSDL 1.1
 - was submitted as a W3C Note by Ariba, IBM and Microsoft
 - March 2001
 - Merging 3 previous proposals:
 - Microsoft's SOAP Contract Language (SCL)
 - Service Description Language
 - IBM's Network Accessible Service Specification Language (NASSL)
- WSDL 2.0
 - March 2006
- not yet a W3C standard



What WSDL is for

- When you create a service, you typically do it because you want other people to use it.
- In order for them to do that, they need to know what information to send to the service, what information the service is going to send back, and where to find the service in the first place.
- It is much more helpful to have a standard, preferably human- and machine-readable, format for this information.



What is WSDL?

- o is an XML document
- o used to describe Web services: specifies
 - the location of the service
 - the operations (or methods) the service exposes



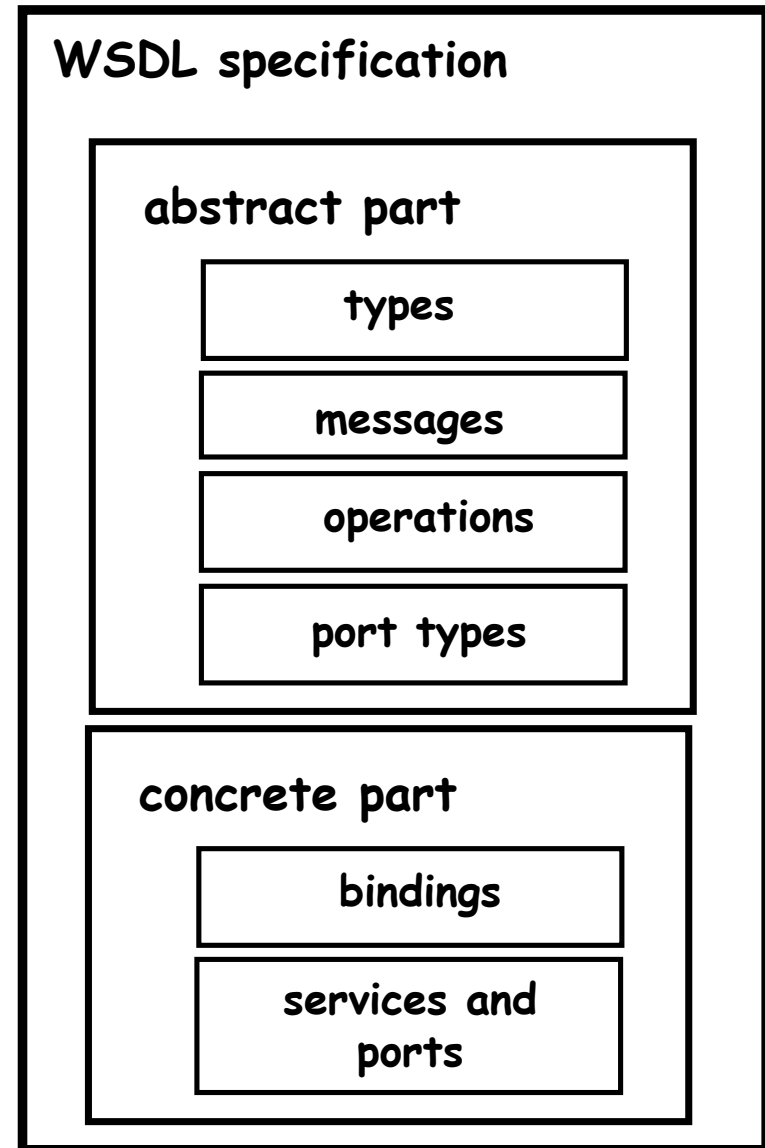
WSDL Data Structure

- WSDL does not presume that exchanges will take place using a particular form of communication
 - Different services could combine different interfaces using different binding, and could make them available at different addresses



The WSDL Document Structure

- Abstract part : describes
 - the messages it sends and receives
 - the operation associates a message exchange pattern with one or more messages
- Concrete part : specifies
 - transport and wire format details for one or more interfaces
 - a port (an endpoint) associates a network address with a binding
 - a service which groups together endpoints that implement a common interface



```

<?xml version="1.0"?>
<definitions name="Procurement"
  targetNamespace="http://example.com/procurement/definitions"
  xmlns:tns="http://example.com/procurement/definitions"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/" >

```

```

<message name="OrderMsg">
  <part name="productName" type="xs:string"/>
  <part name="quantity" type="xs:integer"/>
</message>

```

```

<portType name="procurementPortType">
  <operation name="orderGoods">
    <input message="OrderMsg"/>
  </operation>
</portType>

```

abstract part

messages

operation and port type

```

<binding name="ProcurementSoapBinding" type="tns:procurementPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="orderGoods">
    <soap:operation soapAction="http://example.com/orderGoods"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

```

concrete part

binding

```

<service name="ProcurementService">
  <port name="ProcurementPort" binding="tns:ProcurementSoapBinding">
    <soap:address location="http://example.com/procurement"/>
  </port>
</service>

```

port and service

```

</definitions>

```



The WSDL Document Structure

```
<definitions>  
  <types>  
    definition of types.....  
  </types>  
  <message>  
    definition of a message....  
  </message>  
  <portType>  
    definition of a port.....  
  </portType>  
  <binding>  
    definition of a binding....  
  </binding>  
  <service>  
    definition of a service  
  </service>  
</definitions>
```



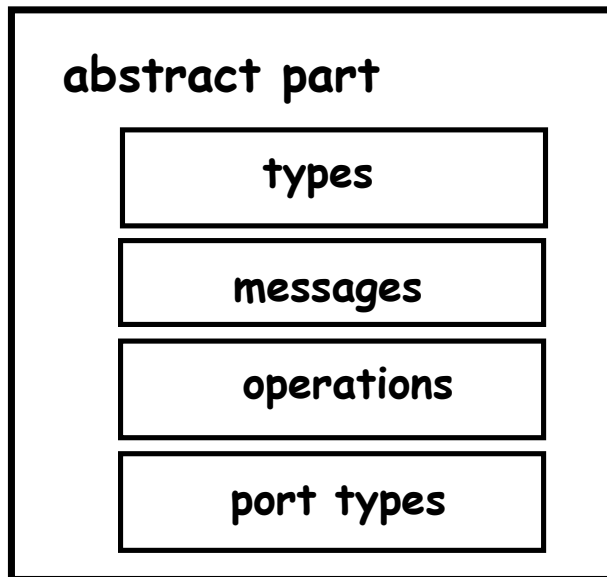

Definitions

```
<definitions name="Procurement"  
  targetNamespace="http://example.com/procurement/definitions"  
  xmlns:tns="http://example.com/procurement/definitions"  
  xmlns:xs="http://www.w3.org/2001/XMLSchema"  
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  
  xmlns="http://schemas.xmlsoap.org/wsdl/" >
```

- The definitions element defines namespaces.
 - `xmlns:xxx` defines the aliases or prefixes for namespaces of the actual elements that make up the WSDL
 - `targetNamespace`, defines the namespace to which the items defined by the WSDL belong



Abstract part



Element	Defines
<code><types></code>	The data types used by the web service (XML Schema syntax used in general)
<code><message></code>	The data elements of an operation
<code><portType></code>	The operations performed by the web service and the messages that are involved.



Abstract part

```
<message name="OrderMsg">  
  <part name="productName" type="xs:string"/>  
  <part name="quantity" type="xs:integer"/>  
</message>
```

messages

```
<portType name="procurementPortType">  
  <operation name="orderGoods">  
    <input message = "OrderMsg"/>  
  </operation>  
</portType>
```

operation and
port type

- Message element
 - Parts : name + type/element
- PortType (Interface) element defines operations, which are made up of messages
 - Operations : One-way/Request-response/ Solicit-response / Notification



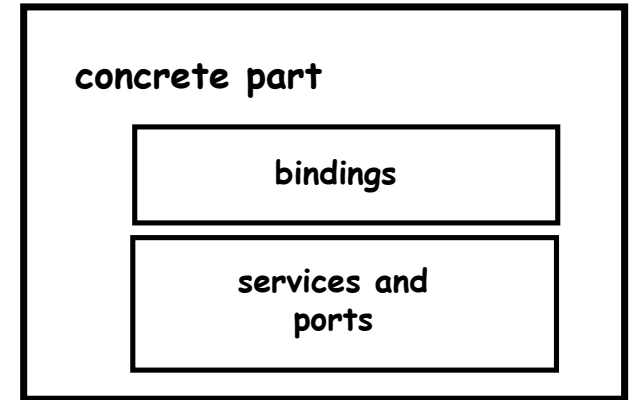
Operation Types

Type	Definition
One-way	The operation can receive a message but will not return a response
Request-response	The operation can receive a request and will return a response
Solicit-response	The operation can send a request and will wait for a response
Notification	The operation can send a message but will not wait for a response

A service can proactively initiate the interaction (unlike IDLs) - operation can invoke thus a service can also behave like a client



Concrete part



```
<binding name="ProcurementSoapBinding" type="tns:procurementPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="orderGoods">
    <soap:operation soapAction="http://example.com/orderGoods"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

— binding

```
<service name="ProcurementService">
  <port name="ProcurementPort" binding="tns:ProcurementSoapBinding">
    <soap:address location="http://example.com/procurement"/>
  </port>
</service>
```

— port and service



InterfaceBinding

```
<binding name="ProcurementSoapBinding" type="tns:procurementPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="orderGoods">
    <soap:operation soapAction="http://example.com/orderGoods"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

- **binding** attributes are
 - name (any name)
 - type (points to the port type defined in the abstract part)
- **soap:binding** attributes are
 - style("rpc" / "document")
 - Transport(Communication protocol : SOAP, Transport protocol : HTTP/SMTP)
- **operation** element
 - defines each operation that the port exposes.
 - must also specify how the input and output are encoded (literal(for document)/SOAP(for rpc))



Port and Service

```
<service name="ProcurementService">  
  <port name="ProcurementPort" binding="tns:ProcurementSoapBinding">  
    <soap:address location="http://example.com/procurement"/>  
  </port>  
</service>
```

- A service may have more than one endpoint, with each one defined by its own port element.
- The port element corresponds to a particular binding, and includes information on how to access it (URI).
- Different ports represent different bindings for the same port type - allows the same functionality to be accessible via multiple transport protocols and interaction styles



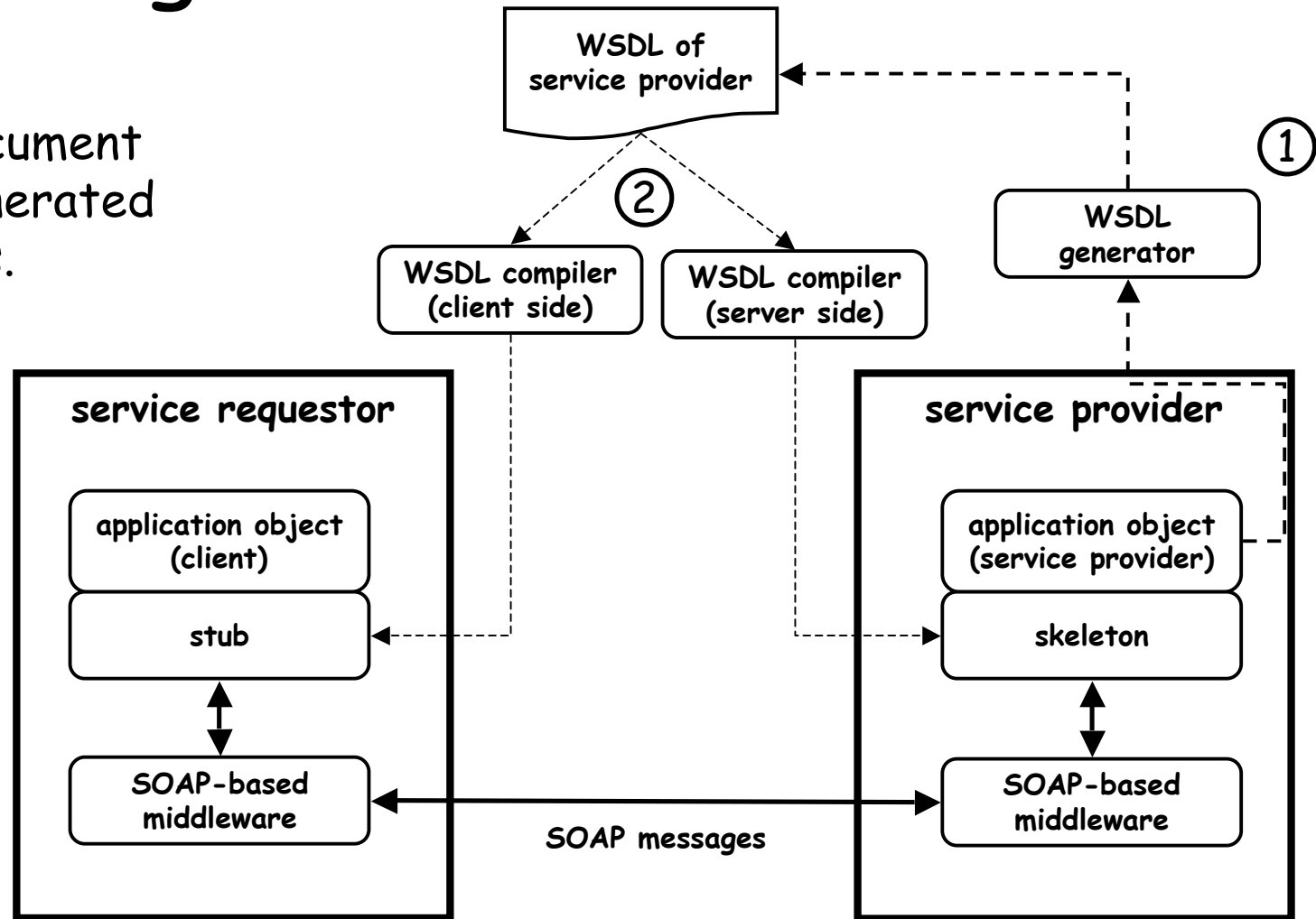
Reusability

- o abstract interfaces: reusable
 - WSDL documents may import other WSDL documents
 - Some documents define abstract interfaces, while other documents import those definitions and make them concrete by defining bindings and addresses



Using WSDL

- WSDL document can be generated from APIs.





WSDL and Other Standards

- Generic service description language - WSDL
- Specific standards
 - Electronic Data Exchange (EDI) used in manufacturing
 - SWIFT used in financial world
- Possibility
 - Hybrid : other standards + WSDL
 - Independent
 - WSDL as a wrapper for these standards so that the systems that support them become Web enabled and can use generic tools for Web access.
 - WSDL may not describe the entire service but only the interface and protocol bindings. The details of the operations will be hidden under a description that uses a specification other than WSDL.



WSDL 2.0

- o inherited most of the WSDL 1.1 architectural principles, including layers of description, flexible authoring style or modularization capability, and extensibility.
 - Abstract Interface:
 - Protocol Binding
 - Service Endpoints

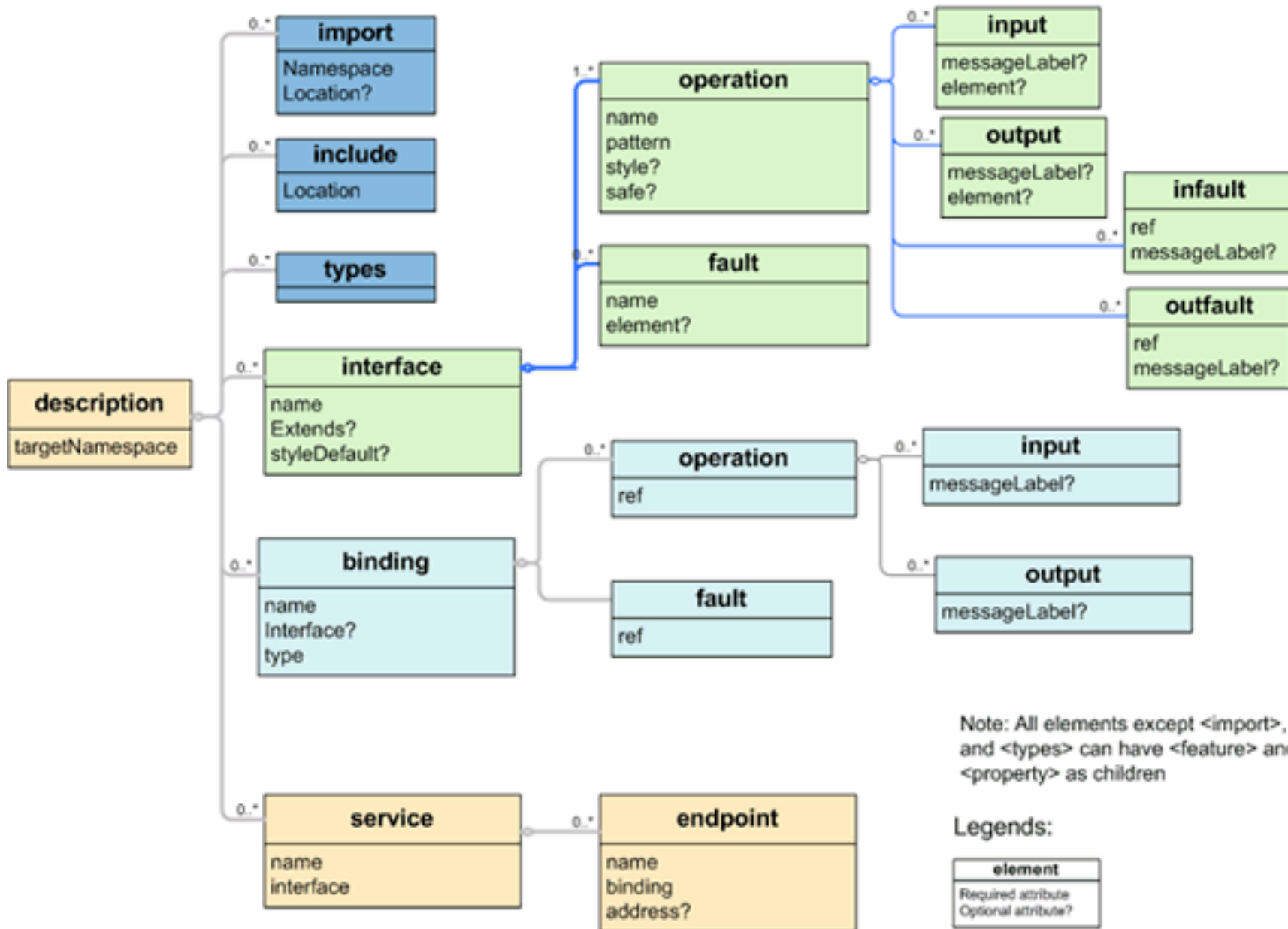


WSDL 1.1 vs. WSDL 2.0

o Flexible Authoring Styles

WSDL 1.1	<ul style="list-style-type: none">▪ <code><wsdl:import></code> enables importing WSDL definitions defined in separate files with the same or different namespaces
WSDL2.0	<ul style="list-style-type: none">▪ <code><wsdl:import></code> for WSDL definitions with a different namespace▪ <code><wsdl:include></code> for WSDL definitions with the same namespace.

WSDL 2.0 Component Model





WSDL 1.1 vs. WSDL 2.0

o Namespace

WSDL 1.1	http://schemas.xmlsoap.org/wsdl/
WSDL2.0	http://www.w3.org/2004/08/wsdl



WSDL 1.1 vs. WSDL 2.0

o Message Exchange Patterns (MEPs)

WSDL 1.1	One-way Request-response Solicit-response Notification	
WSDL2.0	In-bound	In-Only Robust In-Only In-Out In-Optional-Out
	Out-bound	Out-Only Robust Out-Only Out-In Out-Optional-In

● ● ● | Example : Message exchange patterns

```
<wsdl:operation name="finalizeIssue"
  pattern="http://www.w3.org/2006/01/wsdl/in-only">
  <wsdl:input message="tns:finalizeIssueRequestMessage" />
</wsdl:operation>

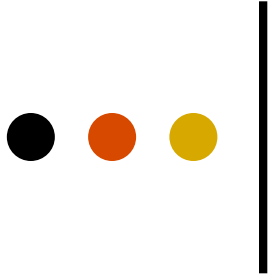
<wsdl:operation name="createNewAd"
  pattern="http://www.w3.org/2006/01/wsdl/in-out">
  <wsdl:input message="tns:createNewAdRequestMessage" />
  <wsdl:output message="tns:createNewAdResponseMessage" />
</wsdl:operation>
```




WSDL 1.1 vs. WSDL 2.0

o Supported Schema

WSDL 1.1	supports XML schema
WSDL 2.0	supports also other schemas, such as RELAX NG and Schematron



Example: Amazon WSDL

<http://www.awszone.com/>



More details

- Understanding Web services
 - <https://www6.software.ibm.com/developerworks/education/ws-understand-web-services2/index.html>
- First Look at WSDL 2.0
 - <https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/74bae690-0201-0010-71a5-9da49f4a53e2>
- Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language
 - <http://www.w3.org/TR/wsdl20/#intro>