Ian Forster states: “Web service have little value if others cannot discover, access, and make sense of them.”

Definition: A **WSDL** document defines services as collections of network endpoints, or ports.

WSDL has a purpose similar to that of IDLs in conventional middleware platforms. A WSDL description describes 3 fundamental properties of a Web Service:

- **What** a service does: operations and the arguments needed to invoke them.
- **How** a service is accessed: details of data formats and protocols.
- **Where** a service is located: details of the protocol-specific network address, such as a URI.

**WSDL Information Model**

**Example for SOAP Request/Response**

**Generating code from WSDL**

**Common Best Practices**

A WSDL document uses the following elements in the definition of network services:

- **Types**: a container for non-built-in data type definitions using some type system, e.g. arrays and structures.
- **Message**: an abstract, typed definition of the data being transferred between the requestor and service;
  - method call (request/response): modeled as 2 messages.
- **Port Type**: an abstract set of operations supported by one or more endpoints; an operation specifies a specific input/output message sequence.
- **Operation**: an abstract description of an action supported by the service.
- **Binding**: specifies a concrete protocol and data format for the operations and messages defined by a particular PortType, such as SOAP or Corba.
- **Port**: a single endpoint defined as a combination of a binding and a network address.
- **Service**: a collection of related endpoints.

**Parts of WSDL**

**Relationship of parts**
WSDL is divided into 2 parts:

- An abstract part which describes what is offered; it consists of types, message, operations and port types.
- A concrete part which describes how and where it is offered; it consists of bindings, services, and ports.

Example for SOAP Request/Response:

WSDL definition of a simple service providing stock quotes; the service supports the single operation GetLastTradePrice(ticker symbol) and returns the price as a float.

```xml
<?xml version="1.0"?>
<definitions name="StockQuote"

targetNamespace="http://example.com/stockquote.wsdl"

xmlns:tns="http://example.com/stockquote.wsdl"

xmlns:xsd="http://example.com/stockquote.xsd"

xmlns:soap="http://schemas.xmlsoap.org/soap/

xmlns=http://schemas.xmlsoap.org/soap/"

<types>

<schema targetNamespace="http://example.com/stockquote.xsd"

 xmlns="http://www.w3.org/2000/10/XMLSchema"

 element name="TradePriceRequest">

 <complexType>

  <all>element name="tickerSymbol" type="string"/> </all>

 </complexType>

 </element>

 <element name="TradePrice">

 <complexType>

  <all>element name="price" type="float"/> </all>

 </complexType>

 <element>

 </complexType>

 </schema>

</types>
```

Example for SOAP Request/Response:

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/

<wsdl:types>

<wsdl:element name="TradePriceRequest">

  </complexType>

  <all>element name="tickerSymbol" type="string"/> </all>

 </complexType>

 </element>

 <element name="TradePrice">

  </complexType>

  <all>element name="price" type="float"/> </all>

 </complexType>

 </schema>

</wsdl:definitions>
```

```
<!-- Parameter der Nachricht -->
<message name="GetLastTradePriceInput">

  <part name="body" element="xsd1:TradePriceRequest"/>

 </message>
```
Example for SOAP Request/Response

```xml
<complexType name="TradePrice">
  <complexContent>
    <restriction base="xsd:float">
      <totalDigits value="4"/>
    </restriction>
  </complexContent>
</complexType>
```

Example for SOAP Request/Response

```xml
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
```

Example for SOAP Request/Response

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

Example for SOAP Request/Response

```xml
<service name="StockQuoteSoapService">
  <documentation>Our defined service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>
```

Web Services Description Language (WSDL)

Ian Forster states: "Web service have little value if others cannot discover, access, and make sense of them."

**Definition:** A **WSDL** document defines services as collections of network endpoints, or ports. WSDL has a purpose similar to that of IDLs in conventional middleware platforms. A WSDL description describes 3 fundamental properties of a Web Service:

- **What** a service does: operations and the arguments needed to invoke them.
- **How** a service is accessed: details of data formats and protocols.
- **Where** a service is located: details of the protocol-specific network address, such as a URL.

**WSDL Information Model**

**Example for SOAP Request/Response**

**Generating code from WSDL**

**Common bad Practices**
Generating code from WSDL

Use of a WSDL compiler to automatically generate code (e.g., a Java interface) from a WSDL file.

- WSDL of service provider
  - WSDL compiler (client side)
  - WSDL compiler (server side)
  - WSDL generator
  - Service requestor
    - application object (client)
  - SOAP-based middleware
  - SOAP messages
  - Service provider
    - application object (service provider)
    - skeleton
  - SOAP-based middleware

WSDL documents can be generated from APIs.
Stubs and skeletons can be generated from WSDL document.

Web Services

Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardized by the W3C community.

Motivation - Example
Service Oriented Architecture - SOA
Web Services - Characteristics
Web Services Architecture
Simple Object Access Protocol (SOAP)
Web Services Description Language (WSDL)
Universal Description, Discovery, and Integration (UDDI)
REST
Web Service Composition
Adopting Web Services
Mashups

Universal Description, Discovery, and Integration (UDDI)

Provides the definition of a set of services supporting the description and discovery of businesses, organizations, and Web Service providers, the Web services they make available, the technical interface to access those services.

UDDI itself is a Web Service, has a WSDL interface and can be described by a UDDI registry.

- UDDI Business Registry System
- UDDI Entities
- UDDI Registry API
- UDDI registry methods for publicly available Web Services.

Common bad Practices

Analysis of existing WSDL documents shows that functionality of many Web services are hard to understand due to bad practices:
- Developers take insufficient care of names and comments.
- Port types are tied to concrete protocols.
- Semantically unrelated operations are placed in a single port type.
- Overload output messages to transport results and error information.
Categorization of the information contained in a UDDI registry:

- UDDI white pages: basic information such as company name, contact information, and services these organizations provide.
- UDDI yellow pages: detailed business data and Web Services, organized by relevant business classification.
- UDDI green pages: information on how to invoke a Web Service.

**UDDI Entities**

**businessEntity**
- Represents the owner of a Web Service.
- Attributes: name, unique key, zero or more services, descriptions, ...

**businessService**
- Represents a group of one or more Web Services.
- Attributes: name, unique key, one binding template per Web Service, descriptions, ...

**bindingTemplate**
- Represents a single Web Service; contains all the information to locate and invoke the service.
- Attributes: name, unique key, an access point that indicates the URL of the Web Service

**TModel**
- Represents WSDL interface types.
- Attributes: name, unique key, an URL that points to the data associated with the TModel, description.
provides the definition of a set of services supporting the description and discovery of
businesses, organizations, and Web Service providers,
the Web services they make available,
the technical interface to access those services.

UDDI itself is a Web Service; has a WSDL interface and can be described by a UDDI registry.

UDDI Business Registry System
UDDI Entities
UDDI Registry API

UDDI registry [methods] for publicly available Web Services.