DEVELOPING WEB SERVICES

SOAP

WSDL

Web Service Description Language

- **Web Service Description Language**
- XML-based language for describing Web service
  - What is available within the web service
  - What are the request and parameters

- Clients can consult the WSDL
- Server (web service provider) provides a WSDL

WSDL

Main Structure

```
<definitions>
  <types>
    Complex data types that will be used
  </types>
  <message>
    Data elements used for each operation
  </message>
  <portType>
    Operations that can be performed and the messages involved
  </portType>
  <binding>
   Binds the operation to a specific port type
  </binding>
  <service>
   Name of the web service (URI)
  </service>
</definitions>
```

WSDL

<definitions>...</definitions>

- Every WSDL has the root element **definitions**
- It contains the definition of one or more services
- Attributes associated to definitions
  - **name -** used to give a name to your wsdl (optional)
  - **targetNamespace -** WSDL documents can import other WSDL documents, and setting targetNamespace to a unique value ensures that the namespaces do not clash.
- **xmlns -** default namespace of the WSDL document (*http://schemas.xmlsoap.org/wsdl/*)
- xmlns:xsd and xmlns:soap are standard namespace definitions that are used for specifying SOAP-specific information as well as data types.
  - *http://www.w3.org/2001/XMLSchema* and *http://schemas.xmlsoap.org/wsdl/soap/*
- xmlns:tns stands for this namespace.

WSDL

<definitions>...</definitions>

```
<definitions name="StockService"
   targetNamespace="http://www.examples.com/stock/stock.wsdl"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:tns="http://www.examples.com/stock/stock.wsdl"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
>
      ......
      ......
</definitions>
```

WSDL

<types>...</types>

- The type elements provides information about any complex data types used in the WSDL document.
- When simple types are used the document does not need to have a types section.

WSDL

<types>...</types>

```
<types>
  <xsd:element name="TradePriceRequest">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="stockName" type="string"/>
       <xsd:element name="closingDate" type="date"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</types>

<!-- INPUT MESSAGE -->
<message name="getStockPrice">
  <part name="stockName" element="TradePriceRequest"/>
</message>
```

WSDL

<message>...</message>

- The message element is used to describe the data that gets exchanged between the web service and the client
- There should always be 2 types of messages
  - **Input** - used to describe the parameters the web service wants.
  - **Output** - used to describe the results of web service.
- Each message, will have a **<part>** element which is used to describe the parameter used by the input and output message.

WSDL

<message>...</message>

```
<!-- INPUT MESSAGE -->
<message name="getStockPrice">
  <part name="stockName" type="xsd:string"/>
</message>

<!-- OUTPUT MESSAGE -->
<message name="getStockPriceResponse">
  <part name="price" type="xsd:decimal"/>
</message>
```

WSDL

<portType>...</portType>

- portType element is used to define one complete operation which is offered by the web service
- A combination of 2 message elements Input and Output are known as a complete operation
- Patterns of Operation
  - One-way - contains single input operation
  - Request-response - Service receives a message and sends a response
    - One input followed by one output
  - Solicit-response - Service sends a message and receives a response
    - One output followed by one input
  - Notification - contains a single output

WSDL

<portType>...</portType>

```
<portType name="stockPrice_portType">
  <operation name="stockPrice">
    <input message="tns:getStockPrice"/>
    <output message="tns:getStockPriceResponse"/>
  </operation>
</portType>
```

WSDL

<binding>...</binding>

- Provides information on how *portType* operation will actually be transmitted and what protocol is used for the transmutation
- SOAP protocol uses the binding **<soap:binding>**, and the transport is SOAP messages on top of HTTP protocol
- You can specify multiple bindings for a single *portType*

WSDL

<binding>...</binding>

- The binding element has two attributes can
  - *name* - defines the name of the binding
  - *type* - points to the portType for the binding
- binding element children
- soap:binding
  - style attribute - rcp or document
  - transport attribute - defines the protocol that will be used
- soap:operation - Indicates the binding of a specific operation to a specific SOAP implementation
- soap:body - allows you to specify the details of the input and output

WSDL

<binding>...</binding>

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

WSDL

<service>...</service>

- The service element defines a list of ports
- The port element defines a port that is available at a specific address
  - soap:address element has a location attribute
    - *this is where the service actually exists*
- And the port is linked to a binding

```
<service name="server">
  <port binding="tns:stockPrice_Binding" name="serverPort">
    <soap:address location="ttp://www.examples.com/stock/" />
  </port>
</service>
```

http://cs.au.dk/~amoeller/WWW/webservices/wsdlexample.html

UDDI

Universal Description, Discovery, and Integration

- UDDI
  - **Universal Description, Discovery, and Integration**
  - XML based for publishing and finding WSDL
  - A directory that helps locate web services
  - Uses WSDL to describe interfaces to web services

UDDI

Structure

- UDDI business registry consists of 3 components
  - White Pages
    - Basic information: address, contact, and known identifiers
  - Yellow Pages
    - industrial categorizations based on standard taxonomies
- Green Pages
  - technical information about services provided
  - Describe how to access a Web Service