Script **generated by TTT**

Title: Distributed_Applications (23.06.2014)

Date: Mon Jun 23 09:16:29 CEST 2014

Duration: 46:14 min

Pages: 20

Web Services

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

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 URI.

WSDL Information Model

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.

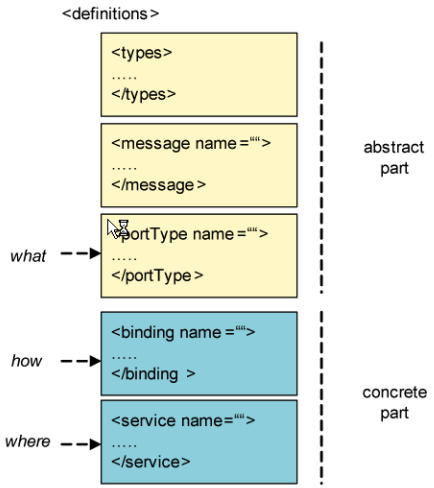
Service: a collection of related endpoints.

Parts of WSDL

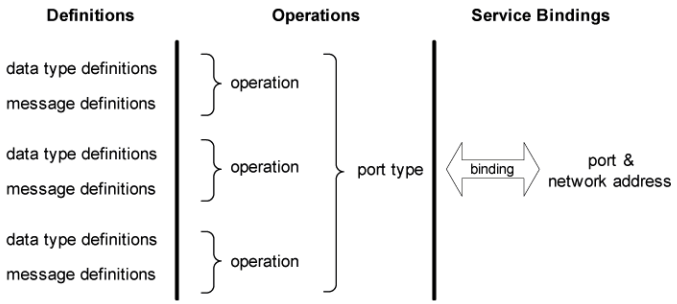
WSDL is divided in 2 parts

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.



Relationship of parts



definitions are generally expressed in XML.

operations describe actions for the messages supported by a Web Service; the equivalent of a method signature in Java.

service bindings connect port types to a port.

Generated by Targeteam

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 version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"
    xmlns:xsd1="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/">

<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>
```

Example for SOAP Request/Response

```
    xmlns= http://schemas.xmlsoap.org/wsdl/ >

<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>
    </schema>
</types>

<!-- Parameter der Nachricht -->
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
</message>
```

```
        <element name="TradePrice">
            <complexType>
                <all><element name="price" type="float"/></all>
            </complexType>
        </element>
    </schema>
</types>

<!-- Parameter der Nachricht -->
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<!-- Parameter der Antwort -->
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
</message>

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



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

<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>

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



```
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>

<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>

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

</definitions>
```


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

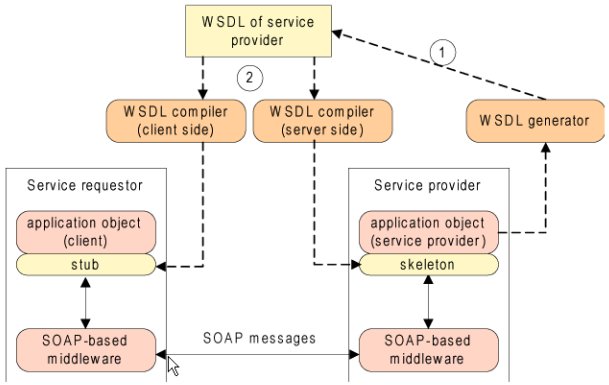
Definition: A WSDL document defines **services** as collections of network endpoints, or ports.

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.

Generating code from WSDL

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



WSDL documents can be generated from APIs (1).

Stubs and skeletons can be generated from WSDL document (2).

Common bad Practices

Analysis of existing WSDL documents shows that functionality of many Web services are hard to understand due bad practices.

developers take not sufficient 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.

Web Services

Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardization 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 **xmethods** for publicly available Web Services.

UDDI Business Registry System

Categorization of the information contained in a UDDI registry.

UDDI white pages: basic information such as company name, contact information, and of services these organizations provide.

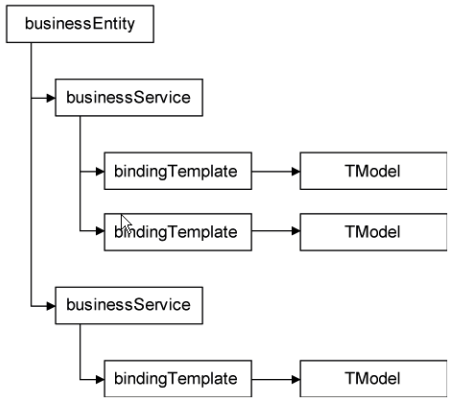
UDDI yellow pages: detailed business data and Web Services, organized by relevant business classification.

UDDI green pages: information how a given Web Service can be invoked.

Generated by Targeteam

UDDI Entities

UDDI allows to store and manipulate four main types of entities



businessEntity : represents the owner of a Web Service.

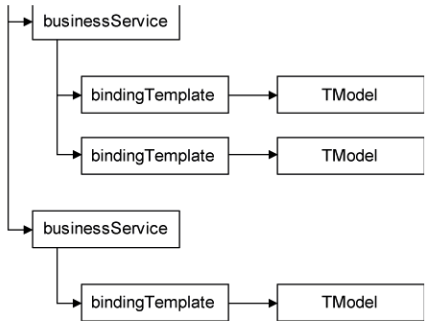
Attributes: name, unique key, zero or more services, descriptions, ...

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

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: 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,

UDDI Registry API

UDDI registries have 3 main types of users

service providers that publish services

requesters that look for services

other registries that need to exchange information.

UDDI supports the following sets of APIs

UDDI Inquiry API : operations to find registry entries such as `find_service` , or get details on specific entity, e.g. `get_serviceDetail` .

UDDI Publishers API : add, modify, and delete entries, e.g. `save_service` or `delete_service` .

UDDI Security API : get and discard authentication tokens to be used in communication with registry.

UDDI Ownership Transfer API : transfer ownership of structures between registries.

UDDI Subscription API : enables monitoring of changes in a registry by subscribing to track new, modified, and deleted entries.

UDDI Replication API : supports replication of information between registries.

Generated by Targeteam

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 **xmethods** for publicly available Web Services.