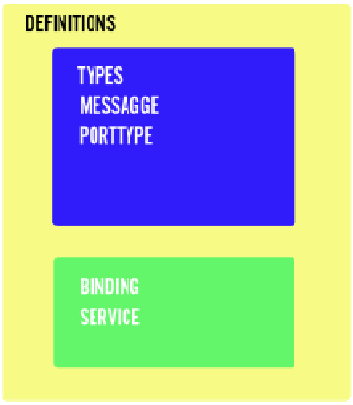
WSDL: Web Services Description Language

Il *Web Services Description Language* (**WSDL**) è un documento XML in grado di descrivere le funzioni e i parametri di un *Web Service*.
Questo documento è quindi una descrizione di come interagire con il servizio in questione ed è quindi usato per la creazione dei client dei servizi web.
All'interno del file WSDL Sono descritte:

- le **operazioni** messe a disposizione dal servizio;
- il **protocollo** di comunicazione da utilizzare per accedere al servizio;
- il **formato** dei messaggi accettati in input;
- gli **output** restituiti e il loro formato;
- gli **endpoint** di ogni funzione.

Il WSDL è solitamente utilizzato in combinazione con SOAP e XML Schema per rendere disponibili i servizi web, e permettere il dialogo automatico macchina/macchina.
Ad esempio, un software client può leggere il documento WSDL relativo ad un Web Service e, conseguentemente, comporre messaggi SOAP per usufruirne correttamente.
Gli elementi principali che compongono un documento WSDL sono:

- *types* - definisce i tipi di dato che possono essere scambiati tra client e web service
- *message* - descrive i messaggi che possono essere scambiati tra il web service e i client
- *portType* - definisce i punti di connessione verso il webservice (ogni operazione esposta ha un elemento *portType*);
- *binding* - descrive un operazione esposta dal servizio web con gli elementi in input, output ed i loro vincoli
- *service* - fornisce una descrizione testuale del servizio (leggibile dall'uomo), e informa i client da dove accedere a quest'ultimo.



Di seguito uno "scheletro" della struttura di base di un file WSDL:

- **<?xml ... ?>** - dichiarazione versione XML
- **<wsdl:definitions ... >** - dichiarazione del WSDL
- **<wsdl:types>... </wsdl:types>** - definizione dei tipi di dato
- **<message> ... </message>** - elenco dei messaggi
- **<portType> ... </portType>** - elenco delle porte
- **<binding> ... </binding>** - associazione tra operazioni e trasporto
- **<service> ... </service>** - definizione del servizio

SOAP: Simple Object Access Protocol

SOAP è l'acronimo di *Simple Object Access Protocol* ed è un protocollo per lo scambio di messaggi tra componenti software (solitamente definiti *client* e *web service*) che avviene secondo le regole della sintassi XML. La parola Object indica che l'uso del protocollo deve essere fatto secondo il paradigma della programmazione orientata agli oggetti.
Il protocollo definisce un set di regole che il client deve rispettare per richiedere la risposta al server che ospita ed espone il *web service*. Attraverso lo scambio di messaggi SOAP definiamo, quindi, delle RPC-Call (*Remote Procedure Call*) cioè delle **chiamate di procedure remote**: in pratica un componente software locale svolge un'operazione attraverso un'elaborazione compiuta, totalmente o in parte, attraverso l'ausilio di un sistema remoto (il web service).

La trasmissione e la negoziazione di questi messaggi XML è regolata secondo i protocolli HTTP o SMTP, all'interno dei quali viene incapsulato il messaggio SOAP.
Elementi che compongono il framework SOAP sono:

- *Initial SOAP sender* - è il nodo che genera il messaggio;
- *SOAP sender* - è il componente incaricato di spedire i messaggi SOAP;
- *SOAP intermediary* - è il componente che si occupa di processare l'header del messaggio SOAP, girandolo al corretto receiver;
- *SOAP receiver* - è il componente incaricato di ricevere i messaggi SOAP;
- *Ultimate SOAP receiver* - è il nodo che riceve il messaggio, ovvero il destinatario.

Messaggio SOAP

Un messaggio SOAP è strutturato da un *header* ed un *body*.
Il segmento *header* è facoltativo e contiene meta-informazioni quali ad esempio il routing, la sicurezza, le transazioni e parametri richiesti da una procedura.
Il segmento *body*, invece, è obbligatorio e trasporta il contenuto informativo (payload). Questo deve seguire uno schema definito dall'XML Schema.

Con il termine *SOAP Envelope* identifichiamo il documento XML che contiene il messaggio SOAP in tutte le sue componenti.

Esempio 1

server.php

```
<?php
        // server
        class MySoapServer {
                public function getMessage() {
                        return 'CIAO A TUTTI!';
                }

                public function addNumbers($num1,$num2) {
                        return $num1+$num2;
                }
        }

        $options = array('uri'=>'http://www.gianpi.eu/SOAP/0/client.php');
        $server = new SoapServer(NULL,$options);
        $server->setClass('MySoapServer');
        $server->handle();
?>
```

client.php

```
<?php
        // client
        $options= array(
                'location'=>   'http://www.gianpi.eu/SOAP/0/server.php',
                'uri'       =>   'http://www.gianpi.eu/SOAP/0/client.php'
        );

        $client = new SoapClient(NULL,$options);
        echo '<br>'.$client->getMessage();
        echo '<br>'.$client->addNumbers(3,6);
?>
```

Esempio 2

test.wsdl

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:tns="http://www.html.it/php_ws_soap" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"
        targetNamespace="http://www.html.it/php_ws_soap">

        <types>
                <xs:schema targetNamespace="http://www.html.it/php_ws_soap">
                        <xs:element name="name" type="xs:string"/>
                        <xs:element name="weburl" type="xs:string"/>
                </xs:schema>
        </types>

        <message name="getWebUrl">
                <part name="name" type="xs:string"/>
        </message>
        <message name="returnWebUrl">
                <part name="weburl" type="xs:string"/>
        </message>

        <portType name="WebServiceTest">
                <operation name="getWebUrl">
                        <input message="tns:getWebUrl"/>
                        <output message="tns:returnWebUrl"/>
                </operation>
        </portType>

        <binding name="WebServiceSOAPBinding" type="tns:WebServiceTest">
                <soap:binding transport="http://schemas.xmlsoap.org/soap/http" type="rpc"/>
                <operation name="getWebUrl">
                        <!-- Localizzazione del service deployato sul server. -->
                        <soap:operation soapAction= "http://www.gianpi.eu/SOAP/2/server.php/
                        getWebUrl"/>
                        <input>
                                <soap:body use="encoded"
                                  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                                  namespace="http://www.html.it/php_ws_soap"/>
                        </input>
                        <output>
                                <soap:body use="encoded"
                                  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                                  namespace="http://www.html.it/php_ws_soap"/>
                        </output>
                </operation>
        </binding>

        <service name="GetWebUrl">
                <port name="WebUrl" binding="tns:WebServiceSOAPBinding">
                        <soap:address location="http://www.gianpi.eu/SOAP/2/server.php"/>
                </port>
        </service>

</definitions>
```

server.php

```
<?php
    class SearchEngineWS {
        function getWebUrl($name){
            echo($name."<br>");
            $engines = array(
                'google' => 'www.google.it',
                'yahoo' => 'www.yahoo.it'
            );
            return $engines[$name] ? $engines[$name] : "Search Engine
            unknown";
        }
    }

    // OPZIONALMENTE: Definire la versione del messaggio soap. Il secondo
    parametro non è obbligatorio.
    $server= new SoapServer("test.wsdl", array('soap_version' => SOAP_1_2));
    $server->setClass("SearchEngineWS");

    // Infine la funzione handle processa una richiesta SOAP e manda un
    messaggio di ritorno al client che l'ha richiesta.
    $server->handle();
?>
```

client.php

```
<?php
    /**
    * Client che chiede al Web service l'indirizzo internet di un motore di ricerca.
    * PHP 5 mette a disposizione l'oggetto SoapClient per definire un client.
    */
    // Se chiamiamo questo file in un browser otteniamo come risposta:
    www.google.it.
    try {
        $gsearch = new SoapClient('test.wsdl');
        echo("SOAP Client creato con successo!<br>");

        $result = $gsearch->getWebUrl('google');
        echo("Servizio Disponibile<br>");
        print_r("Stampa del risultato: ".$result." <br>");
    }
    catch (SoapFault $exception) {
        print_r($exception);
    }
?>
```