National e-Science Centre

Introduction to Web Services Protocols

epcc

Communication and standards

- Efficient (or indeed any) communication is dependent on a shared vocabulary and grammar.
- Because web services deals with inter-organisation communication these must be universal standards.

Underlying standards

- The basic standards for web services are:
- XML (Extensible Markup Language)
- SOAP (simple object access protocol)
- WSDL (web services description language)
- UDDI (universal description, discovery and integration)

The state of standards

- XML 1.0 fairly stable, although Schema are in the process of replacing DTDs (currently Schema 1.1 being worked on).
- SOAP 1.2
- WSDL 2.0 (coming out, 1.2 current)
- UDDI version 3 (Aug 2003)
- BPEL 1.1 (Business Process Execution Language)
- choreography description language (web services work flows)

  started January 2003.

Standards are still volatile and in the process of development.

Web Services Architecture

- Web Services involve three major roles
  - Service Provider
  - Service Registry
  - Service Consumer
- Three major operations surround web services
  - Publishing – making a service available
  - Finding – locating web services
  - Binding – using web services

Making a service available (1)

In order for someone to use your service they have to know about it.

- To allow users to discover a service it is published to a registry (UDDI).
- To allow users to interact with a service you must publish a description of it’s interface (methods & arguments).
- This is done using WSDL.

Making a service available (2)

National e-Science Centre

- Once you have published a description of your service you must have a host set up to serve it.
- A web server is often used to deliver services (although custom application – application communication is also possible).
- This is functionality which has to be added to the web server. In the case of the apache web server a ‘container’ application (Tomcat) can be used to make the application (servlet) available to apache (deploying).

The old transfer protocols are still there.

- Like the grid architecture web services is layered on top of existing, mature transfer protocols.
- HTTP, SMTP are still used over TCP/IP to pass the messages.
- Web services, like grids, can be seen as a functionality enhancement to the existing technologies.

XML

- All Web Services documents are written in XML

- XML Schema are used to define the elements used in Web Services communication

SOAP

National
e-Science
Centre

- Actually used to communicate with the Web Service
- Both the request and the response are SOAP messages
- The body of the message (whose grammar is defined by the WSDL) is contained within a SOAP “envelope”
- “Binds” the client to the web service

WSDL

- Describes the Web Service and defines the functions that are exposed in the Web Service
- Defines the XML grammar to be used in the messages
  - Uses the W3C Schema language

UDDI

- UDDI is used to register and look up services with a central registry
- Service Providers can publish information about their business and the services that they offer
- Service consumers can look up services that are available by
  - Business
  - Service category
  - Specific service

Section

XML

What is XML

- XML stands for extensible markup language
- It is a hierarchical data description language
- It is a sub set of SGML a general document markup language designed for the American millitary.
- It is defined by w3c.

How does XML differ from HTML?

- HTML is a presentation markup language – provides no information about content.
- There is only one standard definition of all of the tags used in HTML.
- XML can define both presentation style and give information about content.
- XML relies on custom documents defining the meaning of tags.

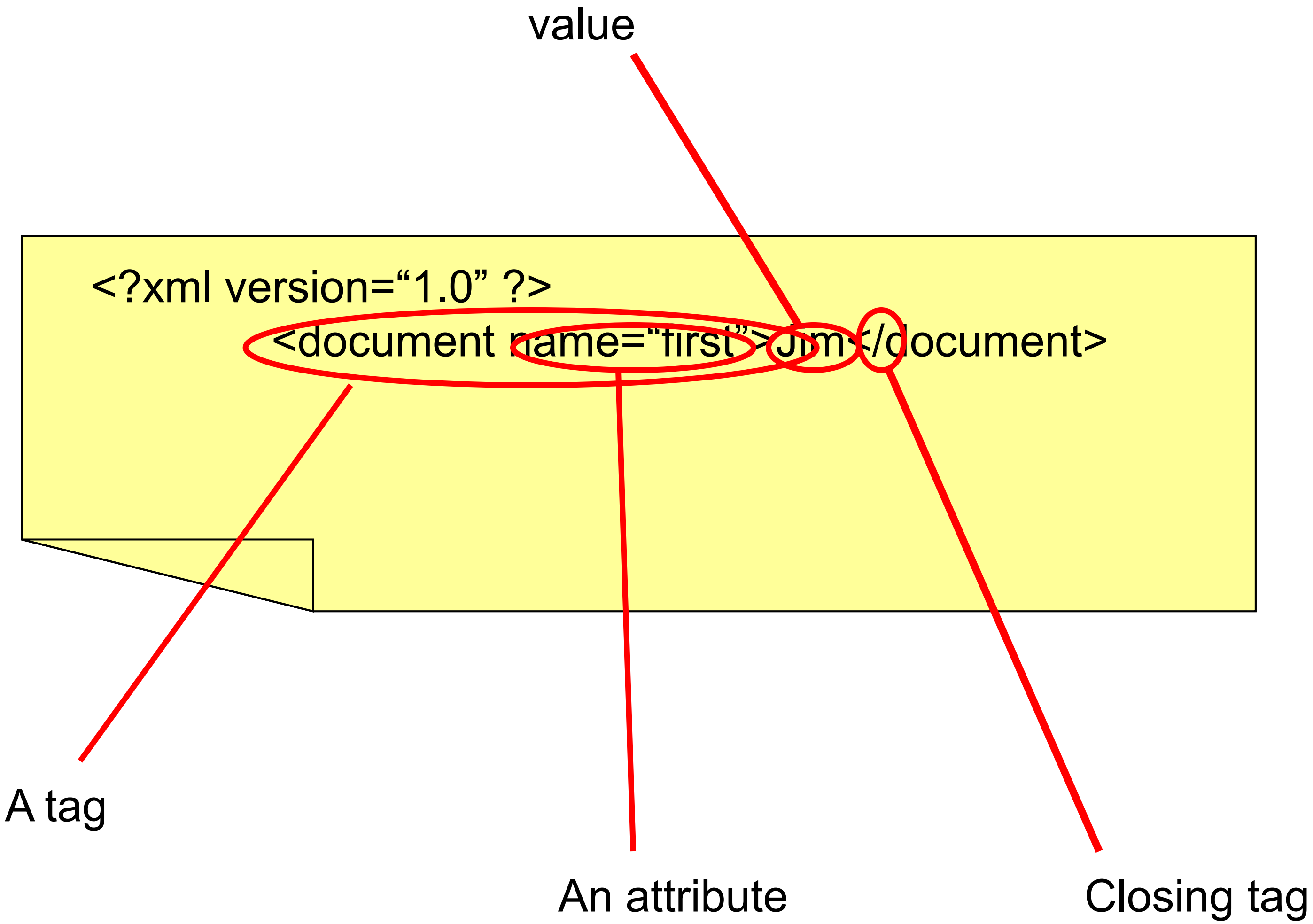
What is a Schema?

- **A schema is the definition of the meaning of each of the tags within a XML document.**

- ***Analogy: A HTML style sheet can be seen as a limited schema which only specifies the presentational style of HTML which refers to it.***

- ***Example: in HTML the tag <strong> pre-defined. In XML you would need to define this in the context of your document.***

A minimal XML document



```
<?xml version=“1.0” ?>
        <document name=“first”>Jim</document>
```

Valid and well formed

- A correct XML document must be both valid and well formed.
- Well formed means that the syntax must be correct and all tags must close correctly (eg <…> </…>).
- Valid means that the document must conform to some XML definition (a DTD or Schema).

(Otherwise there can be no definition of what the tags mean)

Using namespaces in XML

- To fully qualify a namespace in XML write the namespace:tag name. eg.

  <my_namespace:tag> </my_namespace:tag>

- In a globally declared single namespace the qualifier may be omitted.

- More than one namespace:

  <my_namespace:tag> </my_namespace:tag>

  <your_namespace:tag>
  </your_namespace:tag>

  can co-exist if correctly qualified.

Namespaces in programming languages

- **In C/C++ defined by #includes and classes (eg. myclass::variable).**
- **In PERL defined by package namespace, $local and $my (eg. myPackage::variable).**
- **In JAVA defined by includes and package namespace (eg. java.lang.Object)**
- Defines the scope of variables

Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
xmlns=“document" >
<xs:element name = “DOCUMENT”>
          <xs:element name=“CUSTOMER"> </xs:element>
</xs:element>
</xs:schema>
```

Simple schema saved as order.xsd

```
<?xml version=“1.0”?>
<DOCUMENT xmlns=“document”
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
Xsi:schemaLocation=“order.xsd”>
<DOCUMENT>
          <CUSTOMER>sam smith</CUSTOMER>
          <CUSTOMER>sam smith</CUSTOMER>
</DOCUMENT>
```

XML document derived from schema.

Section

National e-Science Centre

SOAP

Request Response Web Services

- **Currently the most common implementation of Web Services**
- **Work in a very simple 'request – response' paradigm**
- **For Example:**
  - A Weather Service– simple request for weather in an area, simple response with the weather report
  - An Airline special offers service – travel agents would simply make requests for latest offers and would receive the offers as a response

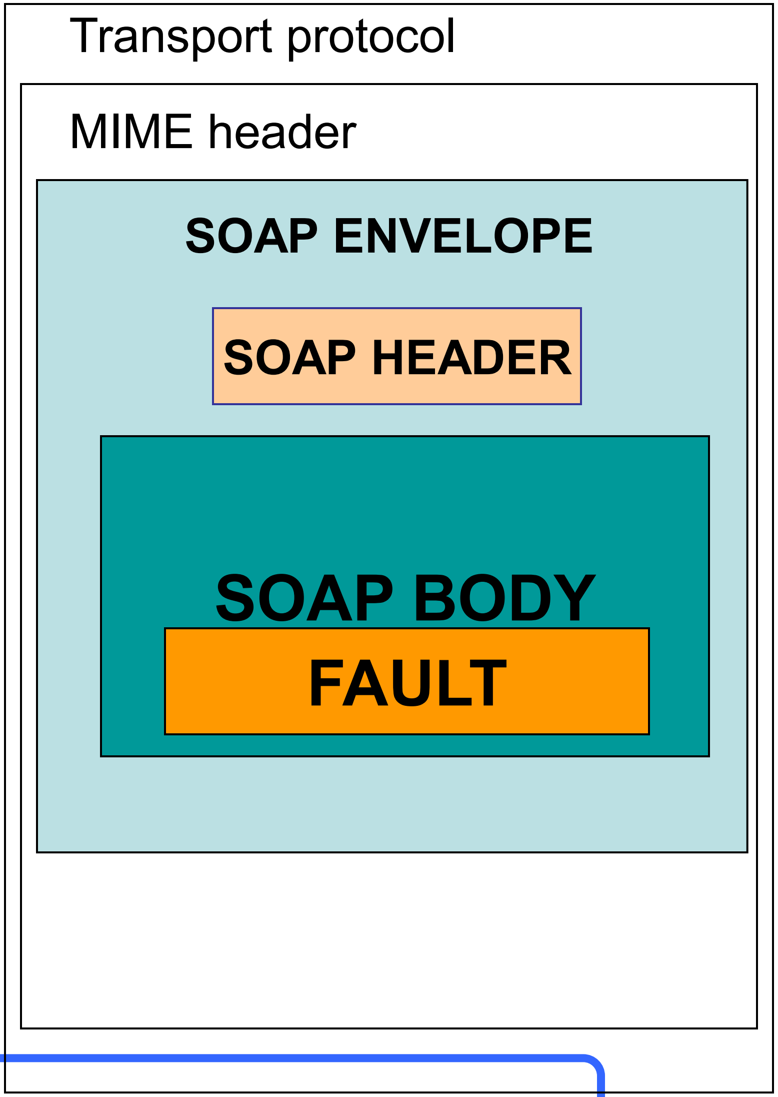
SOAP messages

National e-Science Centre

- SOAP provides a standard ‘envelope’ within which a message can be delivered.
- SOAP is mechanism (protocol) for transferring information (messages) between applications which may be widely distributed.
- SOAP says nothing about the content of the message – the sender and the receiver must understand the message for themselves.
- SOAP is part of a communication stack.

SOAP Structure(1)

- **Each SOAP message will have:**
  - An `Envelope`
  - A `Header` (optional)
  - A `Body`
  - The `Body` may contain a `Fault` element



SOAP Structure(2)

- The envelope wraps the entire soap document
- The header contains allows additional information to be passed as well as the body of the document – e.g. authentication
- The body element contains the core of the SOAP document – this will contain either the RPC call or the XML message itself
- The fault information will contain any exception information

Anatomy of a SOAP message

```
<?xml version='1.0'
   encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP_ENV="http://schemas.xmlsoap.org/soap/envelope/"
         xmlns:xsi="http://www.w3c.org/1999/XMLSchema-instance"
         xmlns:xsd="http://www.w3c.org/1999/XMLSchema">
         <SOAP-ENV:Header>

         </SOAP-ENV:Header


         <SOAP_ENV:Body>


         </SOAP-ENV:Body>


</SOAP-ENV:Envelope>
```

SOAP protocol binding

```
SOAPAction = “urn:soaphttpclient-action-uri”
Host = localhost
Content-Type = text/xml; charset=utf-8
Content-Length = 701
```

```
<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
         xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
         xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>




</SOAP-ENV:Envelope>
```

SOAP RPC

- SOAP RPC messages contain XML that represents a method call or method response
- The SOAP XML will be converted into a method call on the server and the response will be encoded into SOAP XML to be returned to the client

SOAP Faults

- SOAP errors are handled using a specialised envelope known as a Fault Envelope

- A SOAP Fault is a special element which must appear as an immediate child of the body element

- <faultcode> and <faultstring> are required.

A SOAP fault

National e-Science Centre

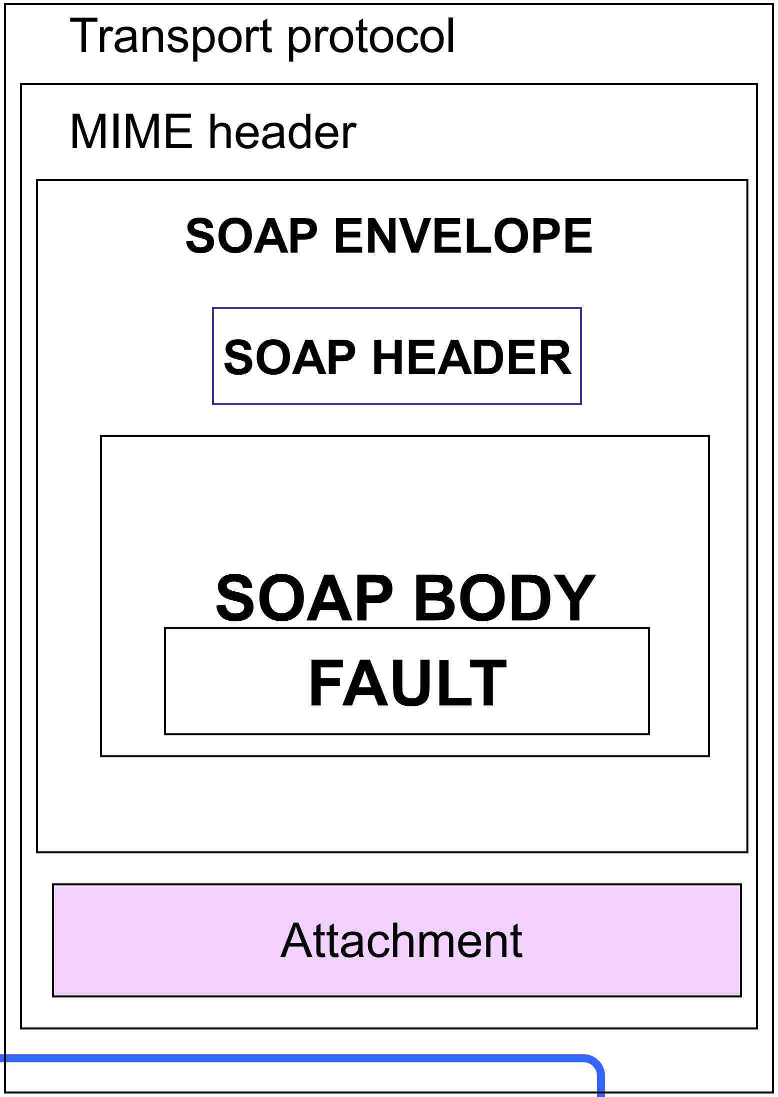
```
<?xml version=‘1.0’ encoding=‘UTF-8’?>
<SOAP-ENV:Envelope
   xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
   xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
   xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
   <SOAP_ENV:Body>
        – <SOAP-ENV:Fault>
                <faultcode>SOAP-ENV:Server</faultcode>
                <faultstring>Test fault</faultstring>
                <faultactor>/soap/servlet/rpcrouter</faultactor>
                <detail>
                          ..
                </detail>
          </SOAP-ENV:Fault>




   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Attachment

- **Large quantities or binary data may not fit well into a XML SOAP message.**
- **In which case it can be sent 'out of band' by attaching it to a SOAP message**
- ***Analogy : email attachments.***



Attaching a file to a SOAP message

- **To add a file to a SOAP message a tag is added within the body of the message.**

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>
<SOAP-ENV:Envelope
xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
            xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
            xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
            <SOAP_ENV:Body>


                        <attachment href=“{URL}”/>


            </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```