Processing of structured documents

Spring 2003, Part 6
Helena Ahonen-Myka

XML for data interchange: Web Services

- Web Services Description Language (WSDL)
- Universal Description, Discovery and Integration (UDDI)
- Simple Object Access Protocol (SOAP)

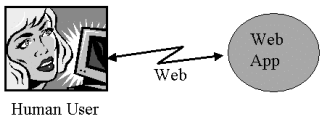
2

Traditional web application

- a traditional web application is designed
  - to take input from a human user (HTML form)
  - to display output to a human user (HTML)
- the user's browser and the web server that hosts the web application act as intermediaries

3

Traditional Web Application

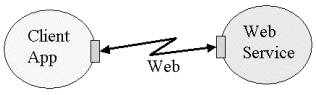


Web service

- a web service is an application component that is
  - accessible on the web
  - is intended to be used by another application (= the client application)
- input and output in some kind of XML format

5

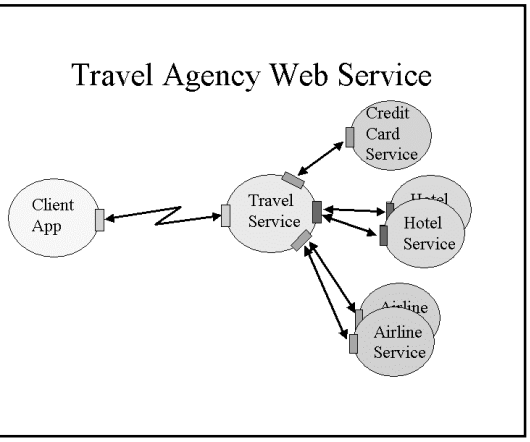
Web Service (WS)



Example: a travel service

- a travel service sells travel packages
- the travel service itself may make use of other web services when implementing its own web services
  - e.g. a credit card service, hotel services, airline services

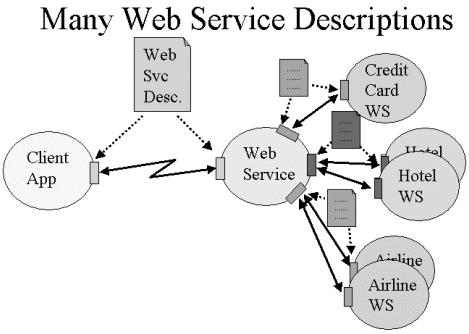
7



Web service descriptions

- if a web service and a client are going to interact meaningfully with each other, they will need to agree on
  - the purpose (what is the meaning of a message?) and
  - the mechanics of the interaction (message formats, datatypes, protocols)
- A **web service description** is a machine-processable document that describes how the client is expected to interact with the service

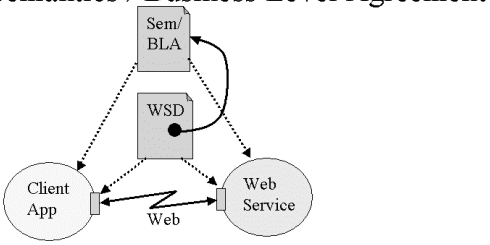
9



Semantics of interaction

- the purpose of the messages and interaction has to be available somewhere
  - e.g. in a separate document that is referenced from the web service description

11



Web Services Description Language (WSDL)

- a WSDL document describes the data that a web service
  - provides to a client and
  - gets from a client
- and how this data will be exchanged based on the underlying transportation protocols

13

WSDL

- general structure of a web service description:
  - Definitions
    - Type definitions, element declarations (schema)
    - Messages
    - Port types
    - Bindings
    - Services

14

Messages

- message is an abstract definition of the data being transmitted
- a message consists of logical parts
  - each part is associated with a definition within some type system (e.g. XML Schema)

15

Example: hotel reservation

```
<message name="inReserveRoom">
  <part name="customerName" type="xs:string" />
  <part name="checkInDate" type="xs:date" />
  <part name="checkOutDate" type="xs:date" />
  <part name="roomType" type="xs:string" />
  <part name="comments" type="xs:string" />
  <part name="agentID" type="xs:string" />
</message>
```

16

Example: hotel reservation

```
<message name="outReserveRoom">
  <part name="roomRate" type="xs:double" />
  <part name="reservationID" type="xs:string" />
</message>
```

17

Port types

- a port type groups related messages into operations (0-n)
- each operation includes a message exchange pattern, either
  - input element, optionally followed by ouput element, or
  - output element, optionally followed by input element
- input and output elements refer to messages

18

Example: hotel reservation

```
<portType name="ptHotelReservation">
   <operation name="oReserveRoom">
      <input message="inReserveRoom"/>
      <output message="outReserveRoom"/>
   </operation>
</portType>
```

19

Bindings

- a binding specifies concrete protocol and data format specifications for the operations and messages defined by a particular portType
- protocols e.g. SOAP, HTTP, MIME

20

```
<binding name="bndHotelReservationSOAP"
         type="tns:ptHotelReservation">
   <soap:binding
      transport="http://www.w3.org/2002/12/soap/bindings/HTTP/" />
   <operation name="oReserveRoom">
      <soap:operation soapAction="http://example.com/oReserveRoom"/>
      <input name="in ReserveRoom">
         <soap:header>...</soap:header>
         <soap:body>...</soap:body>
      </input>
      <output name="outReserveRoom>
         <soap:body>...</soap:body>
      </output>
   </operation>
</binding>
```

Service

- a service component describes
  - the set of port types that a service provides and
  - the ports they are provided over
- a port specifies an address for a binding -> defines a single communication

22

Web service discovery

- web service discovery =
  - to dynamically find a web service that meets the needs of the application

23

Universal Description Discovery and Integration (UDDI)

- UDDI provides a platform-independent way of describing services, discovering businesses, and integrating business services using the Internet
- information stored in a UDDI registry
- two actions:
  - publishing a service
  - searching for a service

24

UDDI

- publishing
  - a service provider has to write WSDL descriptions and register them in a UDDI registry
- searching
  - customers can search for services
  - service interfaces are also displayed, and the services can be invoked

25

Simple Object Access Protocol (SOAP)

- A lightweight protocol for exchange of information in a decentralized, distributed environment
- XML-based protocol
- submitted to W3C in May 2000; latest version Dec 2002 (SOAP 1.2)

26

SOAP

- SOAP consists of three parts:
  - an envelope: defines a framework for describing what is in a message and how to process it
  - a set of encoding rules for expressing instances of application-defined datatypes
  - a convention for representing remote procedure calls and responses
- SOAP messages may be exhanged using a variety or "underlying" protocols
  - usually SOAP is used together with HTTP (Post)

27

Example: SOAP request

```
<env:Envelope
    xmlns:env="http://www.w3.org/2002..."
    env:encodingStyle="http://www.w3.....">
    <env:Body>
        <m:GetLastTradePrice xmlns:m="Some-URI">
            <tickerSymbol>DIS</tickerSymbol>
        </m:GetLastTradePrice>
    </env:Body>
</env:Envelope>
```

28

Example: SOAP response

```
<env:Envelope
    xmlns:env="http://www.w3.org..."
    env:encodingStyle="http://www.w3.....">
    <env:Body>
        <m:GetLastTradePriceResponse
            xmlns:m="Some-URI">
            <price>34.6</price>
        </m:GetLastTradePriceResponse>
    </env:Body>
</env:Envelope>
```

29

SOAP Message Exchange Model

- SOAP messages are fundamentally one-way transmissions between SOAP nodes, from a SOAP sender to a SOAP receiver
- SOAP messages are often combined (by applications) to implement patterns such as
  - request/response
  - multiple, back-and-forth "conversational" exchanges

30

SOAP messages

- a SOAP message contains two SOAP specific subelements within the Envelope
  - Header and Body
  - contents of these subelements are application defined
    - SOAP defines (partially) how these elements must be handled

31

SOAP messages

- a header is optional
  - there may be intermediate SOAP nodes along a message's path from a sender to an ultimate receiver
  - the intermediate nodes may provide value-added services using the header information
    - without understanding the body
- the body is the mandatory element within an Envelope
  - main information is carried in the body

32

SOAP message exchange

- If the exchanged messages conform to a well-defined signature for a remote call and its return, the exchange is modeled as Remote Procedure Calls (RPCs)
- otherwise the message exchanges can be modeled simply as documents exchanged
  - semantics are at the level of sending and receiving applications

33

Remote procedure call

- SOAP specification defines a uniform representation for RPC invocations and responses
- target (the procedure to be called) and the parameters have to be given
  - RPC is carried as a part of the Body element
    - modelled as struct which takes the name of the procedure or method
    - subelements represent the parameters

34

Remote procedure call

- encodingStyle attribute
  - tells how the contents (e.g. parameters) have been serialized
  - SOAP specifies particular encoding schemes for various datatypes, but other encoding schemes may be used for application-specific data

35

Remote procedure call

- A transport binding supporting request-response pattern can provide the correlation between a request and a response (e.g. HTTP)
- the application designer can also put a correlation ID relating a call and its return in a SOAP header
  - makes the RPC independent of any underlying transport

36

Application-specific content

- SOAP and WSDL define the mechanics of an interaction, not the purpose
- the problem remains, how to model the application-specific content?
  - applications/systems should understand each other and interoperate
- (partial) solutions:
  - semantics general to many application areas (e.g. E-commerce in general)
  - vertical schemas (e.g. for retail industry)

37

ebXML

- International initiative established by UN/CEFACT and OASIS
- vision: create a single global electronic marketplace where enterprises of any size and in any geographical location can meet and conduct business with each other through the exchange of XML based messages
- ebXML: a set of specifications

38

ebXML Architecture

- a way to define business processes and their associated messages and content
- a way to register and discover business process sequences with related message exchanges
- a way to define company profiles
- a way to define trading partner agreements
- a uniform message transport layer

39

Vertical specifications

- ontologies, vocabularies, schemas
- e.g. Global Commerce Internet Protocol
  - Internet trading in the consumer good industry
  - manufacturers (e.g. Coca-Cola, Kodak, Nestle, Philips) and retailers (e.g. Carrefour, Marks & Spencer, WalMart Stores) + trade organisations representing 850000 companies (year 2000)
  - ebXML for message transportation and routing

40

XML for data interchange

- messages (XML documents) are exchanged
- abstract descriptions of messages (format + exchange patterns): WSDL
- registry of descriptions: UDDI
- "implementation" of messages: SOAP
  - envelope + metadata
  - application-specific message in the envelope
- application-specific semantics: standard schemas etc. and/or negotiation between business partners

41