JavaScript and AJAX

XML, DTD, XSD, SOAP, WSDL and all that

Huw Davies
daviesah7@cardiff.ac.uk

JavaScript and AJAX

- XML
  - The parent of HTML (or XHTML, at least)
  - The child of SGML
  - More generalised than HTML
  - More restricted than SGML

JavaScript and AJAX

- XML basics
  - Case sensitive: <fred> != <Fred>
  - Elements conform to:
    /^<[A-Z][a-z]_][^<\"\&\']*\s*^/
  - Reserved terms:
    - ':' is valid, but reserved for namespaces
    - The string 'xml' is reserved everywhere
  - Never say 'tag'. Always say 'element'

JavaScript and AJAX

- Well-formed XML obeys the rules
  - Single root node
  - Properly nested tags
  - All tags properly closed
  - All attribute values quoted
- Well-formed does not mean valid or 'correct'

JavaScript and AJAX

- Valid XML
  - Well-formed
  - References either a DTD or a schema
  - Conforms to that DTD/schema

JavaScript and AJAX

DTD Example:

```
<!DOCTYPE NEWSPAPER [

<!ELEMENT NEWSPAPER (ARTICLE+)>
<!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT BYLINE (#PCDATA)>
<!ELEMENT LEAD (#PCDATA)>
<!ELEMENT BODY (#PCDATA)>
<!ELEMENT NOTES (#PCDATA)>

<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>

]>
```

JavaScript and AJAX

- DTD Element Declaration:
  - <!ELEMENT element-name category> or
  - <!ELEMENT element-name (element-content)>
- Category := EMPTY | (#PCDATA) | ANY
  - The type of content it can have

JavaScript and AJAX

- Declaring Attributes
  - <!ATTLIST element-name attribute-name attribute-type attribute-value>
  - Attribute-type
    - CDATA | enumerated list | id | …
  - Attribute-value
    - Default value | #REQUIRED | #IMPLIED | #FIXED

JavaScript and AJAX

- Exercise 1
  - Create a DTD for an email with possible from, to, cc, bcc, and body fields
    - The body may, or may not be HTML

JavaScript and AJAX

- Using a DTD
  - Normally referenced in the <!DOCTYPE>
    - Comes after the XML declaration
    - <!DOCTYPE note SYSTEM "note.dtd">
    - SYSTEM introduces a relative or absolute URL
    - Alternative is PUBLIC (see XHTML doctype)
  - DTD can be included directly in the XML file rather than referencing it

JavaScript and AJAX

- XML schemas
  - XML documents that describe other XML documents
  - Schemas are likely to take over from DTDs

JavaScript and AJAX

Example schema (from w3schools):

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

</xs:element>

</xs:schema>
```

JavaScript and AJAX

Example document conforming to the schema:

```
<?xml version="1.0"?>

<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

JavaScript and AJAX

- Schemas can be more complex than DTDs
- But have some advantages e.g.
  - Data typing of elements
    - xs:string
    - xs:decimal
    - xs:integer
    - xs:boolean
    - xs:date
    - xs:time

JavaScript and AJAX

- Attributes in schemas:
  - <xs:attribute name="xxx" type="yyy"/>
  - If an element can have attributes, it cannot be a simple type
    - The attribute definition comes within xs:complexType

JavaScript and AJAX

Example of defining an attribute in a schema

```
<xs:element name="product">
    <xs:complexType>
      <xs:attribute name="prodid"
type="xs:positiveInteger"/>
  </xs:complexType>
</xs:element>
```

Note: this element is empty – like <meta>

JavaScript and AJAX

Exercise 2

Write a valid XML document that conforms to the following schema:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/
        <xs:element name="body" type="xs:string">
          <xs:complexType>
            <xs:attribute name="html" type="xs:boolean"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

JavaScript and AJAX

- XML is structured just like HTML
- So the DOM applies:

```
<doc>
 <atag>
   <tag2>text</tag2>
   <tag3>xxx</tag3>
   <tag4></tag4>
 </atag>
</doc>
```

JavaScript and AJAX

If the variable x contains the document in the previous slide, then if we code:

```
var ans =
getElementsByTagName['tag2'][0].firstChild.nod
eValue;
```

What does ans contain?

JavaScript and AJAX

Exercise 3

A server program returns XML conforming to the schema set out in exercise3.xsd.

Write a program that obtains data from that server.

A starter program is given to start you off.

JavaScript and AJAX

- Web Services
  - A way of making remote procedure calls over HTTP
  - The interface is specified in a WSDL (Web Service Definition Language) document
  - The request and response objects are contained in SOAP (Simple Object Access Protocol) documents

JavaScript and AJAX

- SOAP
  - Simple Object Access Protocol
  - “A protocol specification for exchanging structured information in the implementation of web services in computer networks.”

JavaScript and AJAX

Using a SOAP object for the request in Ex3:

```
<soap:Envelope
xmlns:soap= "http://www.w3.org/2003/05/soap-envelope">

<soap:Body xmlns:m="http://learn.cf.ac.uk/ajax/2010/hr">

  <m:GetTelNumber>
  <m:FirstName>Fred</m:FirstName>
  <m:LastName>Bloggs</m:LastName>
  </m:GetTelNumber>

</soap:Body>

</soap:Envelope>
```

JavaScript and AJAX

The hr schema to which the SOAP request and response objects must conform is given at:

http://learn.cf.ac.uk/ajax/2010/hr/hr.xsd

JavaScript and AJAX

- Note that the schema defines the following procedures:
  - getTelNumber(ls:FirstName, ls:LastName)
  - getName(ls:TelNumber)
- Note the namespace definitions at the top of the file

JavaScript and AJAX

A SOAP response to a getTelNumber request:

```
<soap:Envelope
xmlns:soap=" http://www.w3.org/2003/05/soap-envelope “>
<soap:Body xmlns:m="http://learn.cf.ac.uk/ajax/2010/hr">
  <m:GetTelNumberResponse>
  <m:FirstName>Fred</m:FirstName>
  <m:LastName>Bloggs</m:LastName>
  <m:TelNumber>4321</m:TelNumber>
  </m:GetTelNumberResponse>
</soap:Body>
</soap:Envelope>
```

JavaScript and AJAX

- WSDL
  - Web Service Description Language
  - Describes the API of a Web Service
    - The service's location
    - The operations (methods) it offers

JavaScript and AJAX

Main structure of a WSDL document (w3schools):

```
<definitions>
<types>
   data type definitions.....
</types>
<message>
   definition of the data being communicated....
</message>
<portType>
   set of operations......
</portType>
<binding>
   protocol and data format specification....
</binding>
</definitions>
```

JavaScript and AJAX

Simplified WSDL fragment (from w3schools):

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

JavaScript and AJAX

- SOAP and WSDL-based web services are much more powerful than I've been able to show here:
  - Security and encryption
  - Capable of dealing with extremely complex requirements
- Criticisms
  - Over-complex
  - Very verbose (network issues)