

Lecture 1: XML and DTD

1. XML Extensible Markup Language

XML is a method of putting structured data in a file format. XML aims to be easy to write, easy to read (by machine) and sufficiently rich to permit the kinds of data people are likely to need.

- An XML document includes text and mark-up; just like HTML.
- XML permits new markup tags to be defined
- XML is rather more strict than HTML

2. Well formed XML

From <http://www.w3.org/TR/2000/REC-xml-20001006>

Definition: A textual object is a well-formed XML document if:

1. Taken as a whole, it matches the production labelled document.
2. It meets all the well-formedness constraints given in this specification.
3. Each of the parsed entities which is referenced directly or indirectly within the document is well-formed.

2.1 Notes

- Tags must be closed.
- Elements may contain elements - nesting must "proper".
- Tags may contain attributes.
- Characters such as < and & must be "escaped" as < and &

3. Valid XML

If an XML document is well-formed we may ask if it is valid. A valid XML document is one that agrees with the additional rules set out in a DTD (document type definition). The DTD includes

- ELEMENT: These dictate what kind of child nodes each element is permitted.
- ATTLIST: This indicates what kind of attributes are permitted.

3.1 A simple example of valid XML

The supermarket system records the stock in XML format.

3.1.1 *supermarket.dtd*

```
<!ELEMENT stock (item*)>
<!ELEMENT item EMPTY>
<!ATTLIST item      BarCode      ID          #REQUIRED
                  legend        CDATA        #REQUIRED
                  price          CDATA        #REQUIRED>
```

3.1.2 *stock.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE stock SYSTEM "supermarket.dtd">
<stock>
  <item price="50" legend="Pr-Burger" BarCode="E1"/>
  <item price="15" legend="Crisp S+V" BarCode="E5"/>
  <item price="15" legend="Crisp C+O" BarCode="E6"/>
  <item price="50" legend="Flat Cola" BarCode="E7"/>
</stock>
```

3.2 Validating parsers

A validating parser will check that a given XML document agrees with its associated DTD. Validating parsers may be found at <http://www.ltg.ed.ac.uk/~richard/xml-check.html>

The data files used here are <http://www.dcs.napier.ac.uk/~andrew/xml/supermarket/>

3.3 DTD Syntax

3.3.1 *!ELEMENT*

The elements tags describe the children that a node may have

EMPTY	The node may have no children
ANY	Anything goes
#PCDATA	Parsed character data – normal characters – but no mark up elements
A?	matches A or nothing; optional A.
A, B	matches A followed by B. This operator has higher precedence than alternation; thus A, B C, D is identical to (A, B) (C, D).
A B	matches A or B but not both
A+	matches one or more occurrences of A – the same as A, A*
A*	matches zero or more occurrences of A

3.3.2 *!ATTLIST*

The attlist tag describes the attributes permitted.

NMTOKEN	Rather like a variable name, no spaces, must start with an alpha or _
NMTOKENS	A space separated list of NMTOKEN
ID	A unique token – maybe used to uniquely identify a node
IDREF	A value that shows up as an ID within this file
IDREFS	A space separated list of IDREF values
CDATA	Any reasonable string (no special characters)

3.3.3 *!ENTITY*

The entity tags allow us to define entities that may be referred to later. & Entities can show up in the text of the XML, % entities are used in the DTD.

3.4 A further example

3.4.1 *today.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE day SYSTEM "supermarket.dtd">
<day>
  <basket>
    <beep BarCode="E1"/>
    <beep BarCode="E5"/>
    <beep BarCode="E5"/>
    <beep BarCode="E5"/>
    <payment>
      <cash/>
    </payment>
  </basket>
  &stk;
</day>
```

3.4.2 *supermarket.dtd - extended*

```
<!ELEMENT stock (item*)>
<!ELEMENT item EMPTY>
<!ATTLIST item BarCode ID #REQUIRED
              legend CDATA #REQUIRED
              price CDATA #REQUIRED>
<!ELEMENT day (basket*, stock)>
<!ELEMENT basket (beep*, payment)>
<!ELEMENT beep EMPTY>
<!ELEMENT payment (cash|card)>
<!ELEMENT cash EMPTY>
<!ELEMENT card EMPTY>
<!ATTLIST beep BarCode IDREF #REQUIRED>
<!ENTITY stk SYSTEM "../supermarket/stock.xml">
```

Notice that:

- beep items must have valid barcodes
- we include the element stock from another file
- we permit payment to be either cash or card

3.5 Questions

1. Which of the following are legal elements:

a	<code><day></day></code>
b	<code><day><basket></basket><basket/><stock/></day></code>
c	<code><day><basket><beep BarCode="E1"/></basket><stock/></day></code>
d	<code><day><basket> <beep BarCode="E1"/> <payment><cash/></payment> </basket> <stock><item BarCode="E1" price="12" description="X"/></stock> </day></code>

2. Suggest some attributes for the elements cash and card.
3. Payment may consist of a combination of cash and card. However there must be at least one and there must be no more than one cash element. Identify the appropriate changes to the DTD.
4. Suggest some appropriate attributes for the day element
5. Suggest how system could be updated to deal with
- more than one till
 - historical data

3.6 Multiple Choice Version One

The following XML document has been used to store multiple choice questions and answers.

```
<?xml version="1.0"?>
<questions>
  <quest id="000001">
    <title>What tag is used with DataBind()</title>
    <q1>&lt;%=</q1>
    <q2>&lt;%=</q2>
    <q3>&lt;%=</q3>
    <q4>&lt;%=</q4>
    <q5>&lt;%=</q5>
    <correct>q2</correct>
    <level>3</level>
  </quest>
  <quest id="000002">
    <title>Which language is C# based on:</title>
    <q1>English</q1>
    <q2>Basic</q2>
    <q3>C++</q3>
    <q4>Visual Basic</q4>
    <correct>q3</correct>
    <level>1</level>
  </quest>
</questions>
```

How might we design a DTD for this document? It may be that we have to change the format slightly.

In particular

- How might the q tags be restricted in a DTD? We may assume:
 - there are at least two tags
 - there are never more than 6 tags in the sequence <q1>, <q2>, ...
 - They must occur in order.
- Why might we want to change the format of the quest id attributes?
- The correct tag indicates the right answer. How might we use the DTD to enforce this reference?