XML DTD
A brief introduction

Glossary

XML E**X**tensible **M**arkup **L**anguage.

DTD **D**ocument **T**ype **D**efinition (**D**escription).

XML element A unit of markup used in XML. With each element, an identifier is associated. Elements are distinguished by putting their identifier between angle brackets (opening tags). Elements may have content and attributes. The end of the content specified by an element is represented by the element's identifier with a "/" character in front, enclosed in angle brackets (closing tags).

XML element attribute A *(name,value)* pair associated with a particular element. Attributes and their values are specified within the XML element.

XML element content Part of XML contained between the *opening* and *closing* tags of an XML element.

XML tag See XML element.

XML DTD

XML DTD allows one to describe the structure of an XML document. XML DTDs contain the means for a user to specify XML elements forming the document, their content/structure and their attributes.

If an XML DTD is associated with a particular XML document, than, an XML parser may perform a number of checks to establish whether the structure of the document matches that described by the DTD. If the checks are successful, the document is declared to be valid (w.r.t. given DTD).

<!DOCTYPE [ ]>

!DOCTYPE declaration is used to specify the root of the XML document and as the "wrapper" around the rest of the DTD. The format of this declaration is

<!DOCTYPE rootElement [Elements] >

Here, rootElement is the id of the root element of the XML document and Elements refers to a sequence of XML DTD declarations describing the structure of the XML document (See below).

Example.

<!DOCTYPE bibliography [
...
] >

defines an XML document that has as its root the element <bibliography>.

<!ELEMENT >

!ELEMENT declaration is the key DTD declaration and is used to define XML elements and their structure. Its format is

<!ELEMENT ElementName (Structure) >

Here, ElementName is the identifier of for the element and Structure is a **regular expression** specifying the structure/content of the element.

Regular Expressions for !ELEMENT declaration

An XML element can be of a simple type (i.e., its content does not include other XML elements) or it can be complex, possibly including other XML elements in it.

XML DTD defines one main simple datatype: #PCDATA (parsed character data). It can be used to specify the structure of simple elements as illustrated in the following example:

<!ELEMENT year (#PCDATA)>

The declaration above defines the XML element <year> which has simple content, such as

```
<year>1997</year>
<year> </year>
<year> Who is on first? </year>
```

We can now define the **regular expressions** for specification of the structure of XML elements:

Expression	Content of the element is
EMPTY	no content
#PCDATA	parsed character data
ElementName	element ElementName
R1 | R2	Either R1 or R2
R1 R2	R1 followed by R2
R1*	0 or more repetitions of R1
R1+	1 or more repetitions of R1
R1?	0 or 1 repetitions of R1
(EName1|Ename2|...|EnameN|#PCDATA)*	mixed content rule

Examples.

1. Different structure for the same element.

   DTD:

```
<!ELEMENT name ((first, last)|full)>
<!ELEMENT full (#PCDATA)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
```

   XML elements:

```
<name><full>Alex Dekhtyar</full></name>
<name>
    <first> John </first>
    <last>  Doe </last>
</name>
```

2. Repeated elements

   DTD:

```
<!ELEMENT roster (student*)>
<!ELEMENT student (sid name major?)>
<!ELEMENT sid (#PCDATA)>
<!ELEMENT major (#PCDATA)>
```

   (assuming <name> is borrowed from above)

   XML elements:

```
<roster/>

<roster>
  <student> <sid>00000001</sid> <name> John Brown</name> </student>
</roster>

<roster>
  <student>
     <sid> 000000002</sid>
     <name> <first> Paul </first> <last> White </last>
 </student>
 <student>
     <sid> 00000003</sid>
     <name> Steve Logan </name>
     <major> Computer Science</major>
 </student>
</roster>
```

3. Elements with mixed (tags embedded into text) content

   DTD

```
<!ELEMENT descr ( (b | i|\#PCDATA)* ) >
<!ELEMENT b (#PCDATA) >
<!ELEMENT i (#PCDATA) >
```

   XML elements:

```
<descr>
</descr>

<descr>
This item is very important
</descr>

<descr>
<i>This</i> item is very <b>important</b>
</descr>

<descr>
<b>This item is very</b> <i>important</i>
</descr>

<descr>
This <b>item</b> is <i>very</i> imor<b>t</b><i>a</i>nt
</descr>
```

4. Arbitrary order of subelements in the content

   DTD:

```
<!ELEMENT pair ((first second)| (second first))>
<!ELEMENT first (#PCDATA)>
<!ELEMENT second (#PCDATA)>
```

   XML Elements

```
<pair>
  <first> X </first>
  <second> Y </second>
</pair>

<pair>
  <second> Y </second>
  <first>  X </first>
</pair>
```

<!ATTLIST >

ATTLIST declaration allows the user to specify the list of attributes for each of the elements defined in the DTD. For each attribute, its type and whether it is a required attribute are specified.

For each XML element which has attributes, one !ATTLIST declaration is needed. The format of the declaration is

<!ATTLIS Element (AttName AttType AttReq)+ >

Here, Element is the name of the XML element for which the attributes are specified; AttName is the name of the attribute, AttType is its type and AttReq describes whether the attribute is required or optional.

AttType can take the following values:

Attribute Type	Meaning
CDATA	Non-parsed character data (string)
ID	Unique Object Identifier (oid)
IDREF	Reference to an object identifier (of another object)
IDREFS	List (separated by spaces) of IDREFs
Enumerated	One of the specified number of values

AttReq can take the following values:

Requirement Spec	Meaning
#REQUIRED	Required attribute (element cannot be w/o it)
#IMPLIED	Implied (optional) attribute)
"String"	Default value
#FIXED *val*	the value of the attribute is fixed to *val*

Examples.

1. Text attributes

   DTD:

   ```
   <!ELEMENT phrase (#PCDATA)>
   <!ATTLIST phrase  language CDATA #REQUIRED
                     keyword  CDATA #IMPLIED
   >
   ```

   XML elements:

   ```
   <phrase language=''English'' keyword=''Knowledge''>
   Knowledge is a deadly friend when noone sets the rules
   </phrase>

   <phrase language =''English''>
   Oh-oh
   </phrase>
   ```

2. IDs and IDREFs

   DTD:

   ```
   <!ELEMENT band (#PCDATA)>
   <!ELEMENT musician (#PCDATA)>
   <!ELEMENT instrument (#PCDATA)>
   <!ATTLIST band members IDREFS #REQUIRED>
   <!ATTLIST musician   mid   ID    #REQUIRED>
                       plays IDREF #REQUIRED>
   <!ATTLIST instrument iid  ID    #REQUIRED>
   ```

   XML elements:

   ```
   <band members=''john paul george ringo''>
   The Beatles
   </band>

   <musician mid=''john'' plays=''g''>
   John Lennon
   </musician>
   <musician mid =''paul'' plays=''b''>
   Paul McCartney
   </musician>
   <musician mid=''george'' plays=''g''>
   ```

```
George Harrison
</musician>
<musician mid=''ringo'' plays=''d''>
Ringo Starr
</musician>

<instrument iid=''g''>guitar</instrument>
<instrument iid=''b''>bass </instrument>
<instrument iid=''d''>drums</instrument>
```

3. Enumerated attributes

   DTD:

```
<!ELEMENT musician (#PCDATA)>
<!ATTLIST musician  plays (guitar|bass|keyboards|drums|vocals) #REQUIRED>
```

   XML:

```
<musician plays=``bass"> Paul McCartney</musician>
<musician plays=``guitar"> George Harrison</musician>
<musician plays=``drums"> Ringo Starr</musician>
<musician plays=``guitar"> John Lennon</musician>
```

Specifying DTDs.

There are two ways to specif a DTD for an XML file.

1. **Embedded DTDs.** DTD is specified in the body of the XML file. In this case, the complete DTD is inserted at the beginning of the XML document:

```
<?xml version=''1.0''?>
<!DOCTYPE salad [
    <!ELEMENT salad  (lettuce?|tomato?|onion?|dressing?)>
    <!ELEMENT lettuce EMPTY>
    <!ELEMENT tomato  EMPTY>
    <!ELEMENT onion   EMPTY>
    <!ELEMENT dressing (#PCDATA)>
]>

<salad>
  <tomato/>
  <onion/>
  <dressing> Italian</dressing>
</salad>
```

2. **Standalone DTDs.** DTD is specified in a separate (.dtd) file. XML document contains a special reference to the DTD file. The name of the DTD file should coincide with the name of the root element.

   menu.dtd

```
<!DOCTYPE menu [
    <!ELEMENT menu (dish+ drink+)>
    <!ELEMENT dish (name type price)>
    <!ELEMENT drink (name type price)>
    <!ELEMENT name (#PCDATA)>
```

```
    <!ELEMENT type (#PCDATA)>
    <!ELEMENT price (#PCDATA)>
    <!ATTLIST price currency CDATA #REQUIRED>
]>
```

menu.xml

```
<?xml version=''1.0''?>
<!DOCTYPE menu SYSTEM ``http://www.cs.uky.edu/~dekhtyar/menu.dtd''>

<menu>
<dish>
  <name>Chicken Kiev</name>
  <type>Entree</type>
  <price currency=''US Dollar''>10.95</price>
</dish>
<drink>
  <name> Ginger Ale</name>
  <type> Non-Alcoholic</type>
  <price currency =''US Dollar''>1.25</price>
</drink>

</menu>
```