

XML Syntax: DTDs

Andy Clark

Validation of XML Documents

- XML documents *must* be well-formed
- XML documents *may* be valid
 - Validation verifies that the structure and content of the document follows rules specified by grammar
- Types of grammars
 - Document Type Definition (DTD)
 - XML Schema (XSD)
 - Relax NG (RNG)

What is a DTD?

- Document Type Definition
 - Defined in the XML 1.0 specification
 - Allows user to create new document grammars
 - A subset borrowed from SGML
 - Uses non-XML syntax!
 - Document-centric
 - Focus on document structure
 - Lack of “normal” datatypes (e.g. int, float)

Document Structure

- Element declaration
 - Element name
 - Content model
- Attribute list declaration
 - Element name
 - Attribute name
 - Value type
 - Default value

Element Declaration

- Content models
 - ANY
 - EMPTY
 - Children
 - Nestable groups of sequences and/or choices
 - Occurrences for individual elements and groups
 - Mixed content
 - Intermixed elements and parsed character data

Children Content Model

- Sequences
 - Order required e.g. (foo,bar,baz)
- Choices
 - Any one from list e.g. (foo|bar|baz)
- Nested sequences and choices
 - e.g. (foo,bar,(baz|mumble))
 - e.g. (foo|(bar,baz))

Children Occurrences

- Specify occurrence count for...
 - Individual elements
 - Groups of sequences and choices
- Occurrences
 - Exactly one e.g. `foo (foo,bar)`
 - Zero or one e.g. `foo?(foo,bar)?`
 - Zero or more e.g. `foo* (foo|bar)*`
 - One or more e.g. `foo+(foo|bar)+`

Attribute List Declaration

- Value types
 - CDATA
 - ENTITY, ENTITIES
 - ID, IDREF, IDREFS
 - NMTOKEN, NMTOKENS
 - NOTATION
 - Enumeration of values e.g. (true|false)
- Default value
 - #IMPLIED, #REQUIRED, #FIXED
 - Default value if not specified in document

Example DTD (1 of 6)

- Text declaration

```
01  <?xml version='1.0' encoding='ISO-8859-1'?>
02  <!ELEMENT order (item)+>
03  <!ELEMENT item (name,price)>
04  <!ATTLIST item code NMTOKEN #REQUIRED>
05  <!ELEMENT name (#PCDATA)>
06  <!ELEMENT price (#PCDATA)>
07  <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (2 of 6)

- Element declarations

```
01 <?xml version='1.0' encoding='ISO-8859-1'?>
02 <!ELEMENT order (item)+>
03 <!ELEMENT item (name,price)>
04 <!ATTLIST item code NMTOKEN #REQUIRED>
05 <!ELEMENT name (#PCDATA)>
06 <!ELEMENT price (#PCDATA)>
07 <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (3 of 6)

- Element content models

```
01 <?xml version='1.0' encoding='ISO-8859-1'?>
02 <!ELEMENT order (item)+>
03 <!ELEMENT item (name,price)>
04 <!ATTLIST item code NMTOKEN #REQUIRED>
05 <!ELEMENT name (#PCDATA)>
06 <!ELEMENT price (#PCDATA)>
07 <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (4 of 6)

- Attribute list declarations

```
01 <?xml version='1.0' encoding='ISO-8859-1'?>
02 <!ELEMENT order (item)+>
03 <!ELEMENT item (name,price)>
04 <!!ATTLIST item code NMTOKEN #REQUIRED>
05 <!ELEMENT name (#PCDATA)>
06 <!ELEMENT price (#PCDATA)>
07 <!!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (5 of 6)

- Attribute value type

```
01 <?xml version='1.0' encoding='ISO-8859-1'?>
02 <!ELEMENT order (item)+>
03 <!ELEMENT item (name,price)>
04 <!ATTLIST item code NMTOKEN #REQUIRED>
05 <!ELEMENT name (#PCDATA)>
06 <!ELEMENT price (#PCDATA)>
07 <!ATTLIST price currency NMTOKEN 'USD'>
```

Example DTD (6 of 6)

- Attribute default value

```
01 <?xml version='1.0' encoding='ISO-8859-1'?>
02 <!ELEMENT order (item)+>
03 <!ELEMENT item (name,price)>
04 <!ATTLIST item code NMTOKEN #REQUIRED>
05 <!ELEMENT name (#PCDATA)>
06 <!ELEMENT price (#PCDATA)>
07 <!ATTLIST price currency NMTOKEN 'USD'>
```

Macro Substitution Using Entities

- What are entities?
 - Document pieces, or “storage units”
 - Simplify writing of documents and DTD grammars
 - Modularize documents and DTD grammars
- Types
 - General entities for use in document
 - Example of use: &entity;
 - Parameter entities for use in DTD
 - Example of use: %entity;

General Entities

- Declaration
 - `<!ENTITY name 'Andy Clark'>`
 - `<!ENTITY content SYSTEM 'pet-peeves.ent'>`
- Reference in document
 - `<name>&name;</name>`
 - `<pet-peeves>&content;</pet-peeves>`

Parameter Entities

- Declaration
 - `<!ENTITY % boolean '(true|false)'`
 - `<!ENTITY % html SYSTEM 'html.dtd'`
- Reference in DTD
 - `<!ATTLIST person cool %boolean; #IMPLIED>`
 - `%html;`

Specifying DTD in Document

- Doctype declaration
 - *Must* appear before the root element
 - *May* contain declarations *internal* to document
 - *May* reference declarations *external* to document
- Internal subset
 - Commonly used to declare general entities
 - Overrides declarations in external subset

Doctype Example (1 of 4)

- Only internal subset

```
01 <?xml version='1.0' encoding='UTF-16'?>
02 <!DOCTYPE root [
03     <!ELEMENT root (stem)>
04     <!ELEMENT stem EMPTY>
05 ]>
06 <root>
07     <stem/>
08 </root>
```

Doctype Example (2 of 4)

- Only external subset
 - Using system identifier

01 <?xml version='1.0' encoding='UTF-16'?>

02 <!DOCTYPE root SYSTEM 'tree.dtd'>

03 <root> <stem/> </root>

- Using public identifier

01 <?xml version='1.0' encoding='UTF-16'?>

02 <!DOCTYPE root PUBLIC '-//Tree 1.0//EN' 'tree.dtd'>

03 <root> <stem/> </root>

Doctype Example (3 of 4)

- Internal *and* external subset

```
01 <?xml version='1.0' encoding='UTF-16'?>
02 <!DOCTYPE root SYSTEM 'tree.dtd' [
03     <!ELEMENT root (stem)>
04     <!ELEMENT stem EMPTY>
05 ]>
06 <root>
07     <stem/>
08 </root>
```

Doctype Example (4 of 4)

- Syntactically legal but never used

01 `<?xml version='1.0' encoding='UTF-16'?>`

02 `<!DOCTYPE root >`

03 `<root>`

04 `<stem/>`

05 `</root>`

Beyond DTDs...

- DTD limitations
 - Simple document structures
 - Lack of “real” datatypes
- Advanced schema languages
 - XML Schema
 - Relax NG
 - ...

Useful Links

- XML 1.0 Specification
 - <http://www.w3.org/TR/REC-xml>
- Annotated XML 1.0 Specification
 - <http://www.xml.com/axml/testaxml.htm>
- Informational web sites
 - <http://www.xml.com/>
 - <http://www.xmlhack.com/>

XML Syntax: DTDs

Andy Clark
