Ch. 6: Creating a DTD

Seoul National University, Internet Database Laboratory

July, 2015
Contents

- Working with DTDs
- Defining an Element
- Defining Choices
- Defining Attributes
- Referencing Attributes
- Restricting Attributes

3 Missions
- XML’s underlying grammar which forms the rules of writing an XML document
- How to transform an XML document into another form (e.g. HTML)
- How to define a custom markup language in XML
Working with DTDs

- **DTD (Document Type Definition)**
  - A set of rules that defines a custom markup language in XML
  - Identifying elements and elements’ attributes

```xml
<?xml version="1.0"?>
<wonder>
  <name>Colossus of Rhodes</name>
  <location>Rhodes, Greece</location>
  <height>107</height>
</wonder>
```

Figure 6.1

```xml
<!DOCTYPE wonder [ 
<!ELEMENT wonder (name, location, height)> 
<!ELEMENT name (#PCDATA)> 
<!ELEMENT location (#PCDATA)> 
<!ELEMENT height (#PCDATA)> ]>
```

#PCDATA = parsed character data (text)

Figure 6.2
Defining a “text” Element

- **PCDATA**
  - Parsed character data
  - Referring text value of an element
    - ex) “Hello”, “4 Privet Dr.”, or “99811”
  - Character data parsed by an XML processor

```xml
<?xml version="1.0"?>
<ancient_wonders>
  <wonder>
    <name language="English">Colossus of Rhodes</name>
    <name language="Greek">Κολοσσός της Ρόδου</name>
    <location>Rhodes, Greece</location>
    <height units="feet">107</height>
    <history>
      <year_built era="BC">282</year_built>
      <year_destroyed era="BC">226</year_destroyed>
      <how_destroyed>earthquake</how_destroyed>
      <story>In 294 BC, the people of the island of Rhodes ...</story>
    </history>
  </wonder>
</ancient_wonders>
```

```xml
<!DOCTYPE wonder [ 
  <!ELEMENT name (#PCDATA)> 
  <!ELEMENT location (#PCDATA)> 
  <!ELEMENT height (#PCDATA)> 
  <!ELEMENT year_built (#PCDATA)> 
  <!ELEMENT year_destroyed (#PCDATA)> 
  <!ELEMENT how_destroyed (#PCDATA)> 
  <!ELEMENT story (#PCDATA)> ]>
```

Figure 6.3

Figure 6.4
Defining an Empty Element

When writing `EMPTY`, don’t use parentheses as is required when writing `(#PCDATA)`.

```xml
<main_image file="lighthouse.jpg" w="528" h="349"/>
<source sectionid="112" newspaperid="53"/>
```

* When writing `EMPTY`, don’t use parentheses as is required when writing `(#PCDATA)`.
Defining an Element having a single child

- To define an element that contains a single child ‘wonder’

**XML**

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

<ancient_wonders>
  <wonder>
    <name language="English">Colossus of Rhodes</name>
    <name language="Greek">Κολοσσός της Ρόδου</name>
    <location>Rhodes, Greece</location>
    <height units="feet">107</height>
    <history>
      <year_built era="BC">282</year_built>
      <year_destroyed era="BC">226</year_destroyed>
      <how_destroyed>earthquake</how_destroyed>
      <story>In 294 BC, the people of the island of Rhodes ...</story>
    </history>
    <main_image file="lighthouse.jpg" w="528" h="349"/>
    <source sectionid="112" newspaperid="53"/>
  </wonder>
</ancient_wonders>
```

**DTD**

```xml
<!ELEMENT ancient_wonders (wonder)>  
```

Figure 6.7

Figure 6.8
Defining an Element having many children elements

To define an element that contains children

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

<ancient_wonders>
  <wonder>
    <name language="English">Colossus of Rhodes</name>
    <location>Rhodes, Greece</location>
    <height units="feet">107</height>
    <history>
      <year_built era="BC">282</year_built>
      <year_destroyed era="BC">226</year_destroyed>
      <how_destroyed>earthquake</how_destroyed>
      <story>In 294 BC, the people of the island of Rhodes ...</story>
    </history>
    <main_image file="lighthouse.jpg" w="528" h="349"/>
    <source sectionid="112" newspaperid="53"/>
  </wonder>
</ancient_wonders>
```

Figure 6.9

```xml
<!ELEMENT wonder (name, location, height, history, main_image, source)>
```

Figure 6.10
Defining Cardinality of an Element

- To define how many occurrences
  - *: as many times as necessary, or not at all (zero or more times)
  - +: at least once, and as many times as desired (one or more times)
  - ?: at most once, if at all (zero or one time)
  - An element without a quantifier must appear exactly once

```xml
<!ELEMENT ancient_wonders (wonder+)>  
<!ELEMENT wonder (name+, location, height, history, main_image, source*)>
```

Figure 6.11
Defining an Element with optional children elements

- `year_destroyed`, `how_destroyed` element may be omitted

**Figure 6.12**

```xml
<!ELEMENT history (year_built, year_destroyed?, how_destroyed?, story)>
```

**Figure 6.15**

```xml
<!ELEMENT history ((year_built, year_destroyed, how_destroyed, story)| (year_built, story))>
```
Defining Choices

- Information for the ancient wonders came from different styles

```xml
<ancient_wonders>
  <wonder>
    <name>Colossus of Rhodes</name>
    <location>Rhodes, Greece</location>
  </wonder>
  <wonder>
    Great Pyramid of Giza, Giza, Egypt
  </wonder>
  <wonder>
    Temple of Artemis at Ephesus
    <city>Ephesus</city>
    <country>Turkey</country>
  </wonder>
</ancient_wonders>
```

```xml
<!DOCTYPE wonder [ 
<!ELEMENT ancient_wonders (wonder+)>
<!ELEMENT wonder (#PCDATA | name | location | city | country)*)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT country (#PCDATA)>
]>
```

Figure 6.13

**DTD**

Figure 6.14
Defining an Element having anything

```xml
<ancient_wonders>
  <wonder>
    <name>Colossus of Rhodes</name>
    <location>Rhodes, Greece</location>
  </wonder>
  <wonder>
    Great Pyramid of Giza, Giza, Egypt
  </wonder>
  <wonder>
    Temple of Artemis at Ephesus
    <city>Ephesus</city>
    <country>Turkey
  </wonder>
  <wonder>
    Mausoleum at Halicarnassus
    <location>
      <city>Bodrum</city>
      <country>Turkey
    </location>
  </wonder>
</ancient_wonders>
```

```xml
<!ELEMENT ancinet_wonders (wonder+)>  
<!ELEMENT wonder (
   #PCDATA | name | location  | city
   | country)*>

<!ELEMENT name (#PCDATA)>  
<!ELEMENT location ANY>  
<!ELEMENT city (#PCDATA)>  
<!ELEMENT country (#PCDATA)>  

<how_destroyed year="426">
  fire</how_destroyed>

<year_destroyed>426</year_destroyed>
```

Figure 6.16

Figure 6.17

Figure 6.18
Defining Attributes

- An attribute definition consists of 4 parts
  - Element name, attribute name, attribute type, and an optional status

- Optional Status
  - #REQUIRED indicates the attribute must contain a value (attribute가 꼭 있어야함)
  - #IMPLIED indicates the attribute may be omitted (attribute가 optional)
  - “vvv” indicates a “default” value (attribute는 있으면 그 옆의 value를 인정, 없으면 attribute value는 “vvv”로 인정)
  - #FIXED “vvv” indicates a value (attribute가 optional, 있으면 값은 꼭 “vvv’)

```xml
<!ELEMENT height (#PCDATA)>  
<!ATTLIST height units CDATA #REQUIRED>
```

```
<height units="feet">39</height>
<height units="meters">feet</height>
<height>39</height>
```

Figure 6.19

Figure 6.21
Defining Attributes

<!ELEMENT height (#PCDATA)>
<!ATTLIST height
  units CDATA #IMPLIED>

Figure 6.23

<!ELEMENT height (#PCDATA)>
<!ATTLIST height
  units CDATA "feet">

Figure 6.24

<!ELEMENT height (#PCDATA)>
<!ATTLIST height
  units CDATA "feet" #FIXED "feet">

Figure 6.25

<!ELEMENT height (#PCDATA)>
<!ATTLIST height
  units CDATA "feet" #FIXED "feet">

Figure 6.26
Defining Attributes with choices

- To allow two possibilities for the value of the `units` attribute in the `height` element: *inches* or *feet*

```xml
<!ELEMENT height (#PCDATA)>
<!ATTLIST height units (inches|feet) #REQUIRED>
```

Figure 6.27

```xml
<height>39</height>
<height units="feet">39</height>
<height units="inches">39</height>
<height units="meters">39</height>
```

Figure 6.28
Defining Attributes with unique values

- **ID attributes** are defined to have a value that is unique (not repeatable)

```
<!ELEMENT wonder (name)>
<!ATTLIST wonder code ID #REQUIRED>
```

Figure 6.29

```xml
<wonder code="w_143">
  <name language="English">
    Hanging Gardens of Babylon
  </name>
</wonder>

<wonder code="w_284">
  <name language="English">
    Statue of Zeus at Olympia
  </name>
</wonder>
```

Figure 6.30
Referencing Attributes with unique values

- An attribute whose value is the same as any existing ID attribute in the XML document is called an IDREF attribute.

```xml
<!ELEMENT special_site (title, url)>
<!ATTLIST special_site wonder_focus IDREF #REQUIRED>
```

- The wonder_focus attribute must contain a value from an existing ID attribute in the document:

```xml
<special_site wonder_focus="w_143">
  <title>The Lost Gardens</title>
  <url>www.lost-gardens.com</url>
</special_site>

<special_site wonder_focus="w_143">
  <title>Herodotus in Babylon</title>
  <url>www.herodotus.com/babylon</url>
</special_site>

<special_site wonder_focus="w_284">
  <title>Zeus at Olympia</title>
  <url>www.olympiazeus.com</url>
</special_site>
```
Contents attribute can contain a list of the IDs of the wonders on which the general_site element focus

```
<!ELEMENT general_site (title, url)>  
<!ATTLIST general_site contents IDREFS #REQUIRED>
```

```
<general_site contents= "w_143 w_284" > 
<title>Wonders of the World</title> 
=url>www.wonders_of_the_world.com</url> 
</general_site>
```
Restricting Attributes

- To restrict attributes to valid XML names
  - The value of an attribute defined as the `NMTOKEN` type, must be a valid XML name
  - A value that begins with a letter or an underscore
  - A value that contains only letters, numbers, underscores, hyphens, and periods

```xml
<!ELEMENT w_visit EMPTY>
<!ATTLIST w_visit primary_keyword NMTOKEN #REQUIRED>
```

Figure 6.35

```xml
<wonder>
  <w_visit primary_keyword = "great pyramid"/>
</wonder>
```

Figure 6.36