Egyptian E-Learning University (EELU)
Faculty of Computers and Information Technology

EELU
الجامعة المصرية للتعلم الإلكتروني
Egyptian E-Learning University

The Web Engineering 3

Introduction to Semantic Web

Lecture 7

Structured web documents in XML (Cont.)

Presented by

Prof. Khaled Wassif

Course Topics

- Introduction to the semantic web.
- Semantic web technologies and layered approach.
- Structured web documents in XML.
- Describing web resources in basic elements of Resource Description Framework (RDF).
- Web Ontology Language: OWL.
- Ontologies Applications.

Course References

1. Grigoris Antoniou, Paul Groth, Frank van Harmelen, Rinke Hoekstra, "A Semantic Web Primer", 2012.
2. John Domingue, Dieter Fensel, James A. Hendler, "Introduction to the Semantic Web Technologies", 2011.

Lecture Outline

- Introduction
- Detailed Description of XML
- **Structuring**
    a) DTDs
    b) XML Schema
- Namespaces
- Accessing, querying XML documents: Xpath
- Transformations: XSLT

Structuring of XML Documents

Properties of valid XML document:

- Define the names of all the elements and attributes that may be used.
- Define the structure (what is the meaning of define structure)
  - What are the values that an attribute may takes?
  - which elements may or must occur within other elements, etc. (sequencing operation)
- It is well-formed.
- Respects the structuring information in its uses.

Structuring of XML Documents

- There are two ways of defining the structure of XML documents:
  - DTDs (Document Type Definition)
    - The older and more restricted way.
  - XML Schema (offers extended possibilities)

DTD: Element Type Definition

```
<lecturer>
    <name>Yasser Ibrahim</name>
    <phone> +2–010–3875 507 </phone>
</lecturer>
```

The Standard Way discussed before

DTD for the above element (and all lecturer elements):

```
<!ELEMENT lecturer (name, phone)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

DTDs Way

The Meaning of the DTD

- The element types **lecturer**, **name**, and **phone** may be used in the document.
- A **lecturer** element contains a **name** element and a **phone** element, in that order (***sequence constraint***)
- A **name** element and a **phone** element may have any content.
- In DTDs, **#PCDATA** is the default type for elements.

DTD: Disjunction in Element Type Definitions

1) Case 1: expressing that a **lecturer** element contains *either* a **name** element *or* a **phone** element as follows:

   <!ELEMENT lecturer (name | phone)>

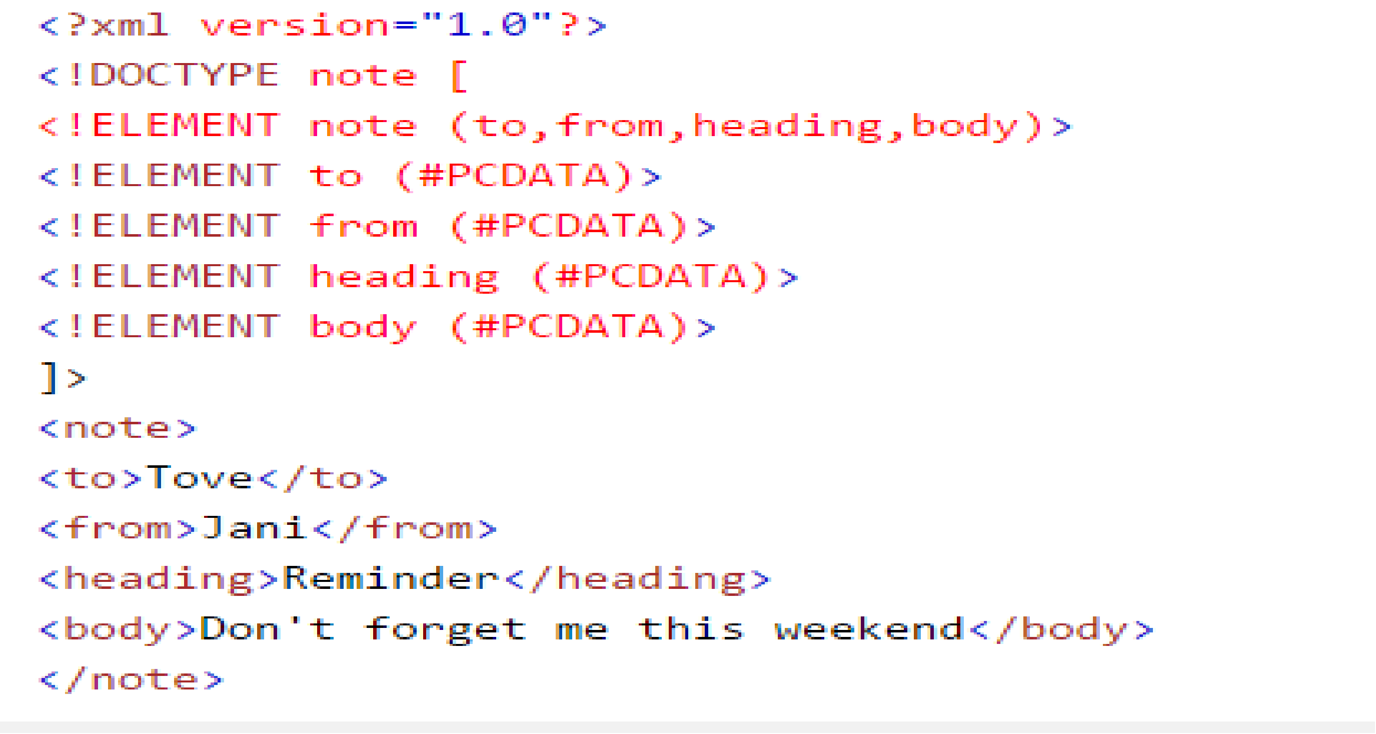
2) Case 2: A **lecturer** element contains a **name** element and a **phone** element in *any order*.

   <!ELEMENT lecturer((name, phone) | (phone, name))>

3) Case 3: A **lecturer** element contains a **name** element and a **phone** element in *any order*, the **name** element may exist alone, and the **phone** element may exist alone.

   <!ELEMENT lecturer((name | phone) | (phone | name))>

Example DTD and XML Elements



```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

- **!DOCTYPE note** defines that the root element of this document is note
- **!ELEMENT note** defines that the note element must contain four elements: "to,from,heading,body"
- **!ELEMENT to** defines the to element to be of type "#PCDATA"
- **!ELEMENT from** defines the from element to be of type "#PCDATA"
- **!ELEMENT heading** defines the heading element to be of type "#PCDATA"
- **!ELEMENT body** defines the body element to be of type "#PCDATA"

Example of an XML Element

```
<order>

      <orderNo="23456" customer="John Smith" date="October 15, 2002">

     <item itemNo="a528" quantity="1"/>

     <item itemNo="c817" quantity="3"/>

</order>
```

Usually, the element form discussed before contains attributes within elements

The DTD Shapes comparing with XML shape

The DTD structure for the cases:

- Element contain one attribute or more
- An empty element
- A single attribute

```
<!ELEMENT (order, item) >

<!ATTLIST order      orderNo      ID          #REQUIRED
                     customer     CDATA       #REQUIRED
                     date         CDATA       #REQUIRED>

<!ATTLIST item       itemNo       ID          #REQUIRED
                     quantity     CDATA       #REQUIRED
                     comments     CDATA       #IMPLIED>
```

The DTD Shapes comparing with XML shape

The DTD structure for the cases:

- Element contain one attribute or more
- An empty element
- A single attribute

```
<!ELEMENT item EMPTY>
```

The DTD Shapes comparing with XML shape

The DTD structure for the cases:

- Element contain one attribute or more
- An empty element
- A single attribute

```
<!ELEMENT item EMPTY>
<!ATTLIST item       itemNo      ID          #REQUIRED
                     quantity    CDATA       #REQUIRED
                     comments    CDATA       #IMPLIED>
```

Comments on the DTD (2)

- In addition to defining elements, we define attributes
- The **attribute list** containing:
  - Name of the element to which the list of attributes is applied.
  - The internal attributes list contains three descriptions (attribute name, attribute type, and value type).
- ***Attribute name***: A name must be used in an XML document using a DTD format.

DTD: Attribute Data Types

- Like predefined data types:
- The most important types are
  - **CDATA**, a string (sequence of characters)
  - **ID**, a name that is unique across the entire XML document
  - **IDREF**, a reference to another element with an ID attribute carrying the same value as the IDREF attribute
  - **IDREFS**, a series of IDREFs
  - **(v_1| . . . |v_n)**, a number of all possible values

DTD: Attribute Value Types (Constraints)

- **#REQUIRED**
  - Attribute must appear in every occurrence of the element type in the XML document
- **#IMPLIED**
  - The appearance of the attribute is optional
- **#FIXED "value"**
  - Every element must have this attribute value
- **"value"**
  - Specifies the default value for the attribute

DTD: Quantifier for elements and attributes

Quantifier is a single character that immediately follows a specified item (in element or attribute) to restrict the number of occurrences of these items at the specified position it may be either:

- **+** for specifying that there must be at least one or more occurrences of the item (the item is not optional)
- ***** for specifying that any number (zero or more) of occurrences is allowed (the item is optional, and the effective content of each occurrence may be different)
- **?** for specifying that there must not be more than one occurrence (the item is optional but for only one occurrence)

➢ If there is no quantifier, the specified item must occur exactly one time at the specified position in the content of the element.

Referencing with IDREF and IDREFS

```
<!ELEMENT family (person*)>

<!ELEMENT person (name)>

<!ELEMENT name (#PCDATA)>

<!ATTLIST person  id        ID        #REQUIRED
                  mother    IDREF     #IMPLIED
                  father    IDREF     #IMPLIED
                  children  IDREFS    #IMPLIED>
```

A DTD example for an Email Element

```
<!ELEMENT email (head,body)>
<!ELEMENT head (from,to+,cc*,subject)>
<!ELEMENT from EMPTY>

<!ATTLIST from   name      CDATA    #IMPLIED
                 address   CDATA    #REQUIRED>
<!ELEMENT to EMPTY>
<!ATTLIST  to    name      CDATA    #IMPLIED
                 address   CDATA    #REQUIRED>

<!ELEMENT cc EMPTY>

<!ATTLIST cc     name      CDATA    #IMPLIED
                 address   CDATA    #REQUIRED>
```

A DTD for an Email Element (Cont.)

```
<!ELEMENT subject (#PCDATA)>
<!ELEMENT body (text,attachment*)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT attachment EMPTY>
<!ATTLIST attachment    encoding  (mime|binhex)  "mime"
                          file       CDATA      #REQUIRED>
```

- mime : Multipurpose Internet Mail Extensions is an extension of the original Simple Mail Transport Protocol (SMTP) email protocol.
- binhex : referring to document location

Interesting Parts of the DTD

- A **head** element contains (in that order):
  - a **from** element
  - at least one **to** element
  - zero or more **cc** elements
  - a **subject** element
- In **from**, **to**, and **cc** elements
  - the **name** attribute is not required
  - the **address** attribute is always required

Interesting Parts of the DTD (2)

- A **body** element contains
  - a **text** element
  - possibly followed by a number of **attachment** elements
- The **encoding** attribute of an **attachment** element must have either the value “**mime**” or “**binhex**”
  - “**mime**” is the default value

Thank you

Prof. Khaled Wassif