Desarrollo de Aplicaciones en Red

José Rafael Rojano Cáceres
http://www.uv.mx/rrojano

Working with XML

Introduction

- **In many of our lectures we have seen XML. So by now it will not be strange, but let me start to define it…**
- **XML stand for Extensible Markup Language. But in fact it is not a markup language.**
- **XML is a meta language, it means that you can define new markup languages.**

Characteristics

- ***It's an open standard, it does not belong to a company nor married to any particular software***
- ***XML offers semantic to data***
- ***It's easy to parser by humans and programs***
- ***XML can store and organize any kind of information***

Component of XML

- **XML document.** It contains data and can be thought of as a data store or a mini-database.
- **XML Parser**: it's a program that understand XML producing a data structure that can be manipulated.
- **Document Type Definition (DTD):** It describe the tags allowed to exist and their relationship. (obsolete by XML schema).
- **XML schema:** It's an improved version of DTD.
- **Namespaces:** It defines unique names that let us avoid conflict between tag names.

Some rules to remember

- ***XML is case sensitive***
- ***Every open tag must be closed***
- ***All the tags must be nested correctly***
- ***Attribute values must be in quotation marks***
- ***Some characters must be represented by entity references.***
- ***The document XML must have a prologue***
- ***If a document follow this rule it's considered "well formed"***

A quick example

```
SNL Diner
Lunch Hours: 11:00AM - 2:00PM

Lunch Items
==============================
Cheeseburger* ......... $2.50
Chips ................. $1.25
Pepsi ................. $.75

Special Combo (Cheeseburger,
Chips, & Pepsi) ........ $4.50

* Soy burger available on
request.
```

```
<?xml version="1.0" encoding="utf-8" ?>
<restaurant>
  <name>SNL Diner</name>
  <menu type="Lunch" start-time="11:00AM"
  end-time="2:00PM">
    <items>
      <item id="L1">
        <name>Cheeseburger</name>
        <price>$2.50</price>
        <note>Soy burger available on
      request.</note>
      </item>
      <item id="L2">
        <name>Chips</name>
        <price>$1.25</price>
      </item>
      <item id="L3">
        <name>Pepsi</name>
        <price>$.75</price>
      </item>
      <combo id="C1">
        <name>Special Combo</name>
        <item ref="L1"/>
        <item ref="L2"/>
        <item ref="L3"/>
      </combo>
    </items>
  </menu>
</restaurant>
```


Starting a XML document (1)

- ***Generally, two files are needed by a XML compliant application***
  - **The XML document**
  - **The Document Type Definition (DTD)**
  - **Optionally a css file can be included**
- ***Every single XML document must start with the declaration about it's a XML document, this is done by the tag:***
  - **<?xml version="1.0" encoding="utf-8" ?>**
- ***Another valid declarion could be***
  - **<? xml version="1.0" ?>**
  - **<? xml version='1.0' encoding='US-ASCII' standalone='yes' ?>**
  - **<? xml version = '1.0' encoding= 'iso-8859-1' standalone ="no" ?>**

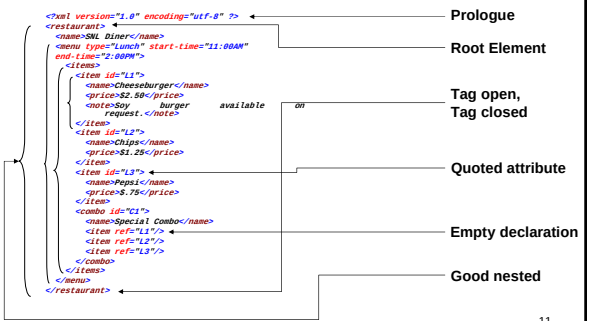
Defining tags on XML

- ***Each tag is defined with and starting and ending definition tag***
  - **<tagname> </tagname>**
  - **<name>SNL Diner</name>**
- ***Every tag must be closed, if there is not element inside it, can be used the empty declartion***
  - **<tagname />**

Attributes

- ***Each tag can contain attributes which describe information related to the element.***
- ***Attributes must be unique on an single element.***
- ***<menu type="Lunch" start-time="11:00AM" end-time="2:00PM">***

Anatomy



Namespace (1)

- ***We can find a problem if we try to use tag vocabularies that are identical, to fix this we can use namespaces.***
- ***This mean that each attribute or tag should start with a string that identify it uniquely.***
  - **<abc:Nombre> and <xyz:Nombre>**
- ***In order to make more readable an XML document and functional we can define a variable with the specific string:***

```
<?xml version="1.0" encoding="utf-8" ?>
<alias:restaurant xmlns:alias="www.comida.com/burgerking"
                  xmlns:dish="www.comida.com/burgerking/promotion">
  <name>SNL Diner</name>
  <menu type="Lunch" start-time="11:00AM"
  end-time="2:00PM">
    <dish:items>
```

DTD example

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE Edit_Mensaje SYSTEM
"Lista_datos_mensaje.dtd"
[
  <!ELEMENT Edit_Mensaje (Mensaje)*>
]>
<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail> Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades....
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades....
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Este es el DTD de Edit_Mensaje -->

<!ELEMENT Mensaje (Remitente, Destinatario, Asunto,
Texto)*>
        <!ELEMENT Remitente (Nombre, Mail)>
            <!ELEMENT Nombre (#PCDATA)>
            <!ELEMENT Mail  (#PCDATA)>

        <!ELEMENT Destinatario (Nombre, Mail)>
            <!ELEMENT Nombre (#PCDATA)>
            <!ELEMENT Mail  (#PCDATA)>

        <!ELEMENT Asunto (#PCDATA)>
        <!ELEMENT Texto  (Parrafo)>
            <!ELEMENT Parrafo (#PCDATA)>
```

13

XML schema (1)

- *XML schema is the way to define what is a legal XML document.*
- *XMLs is a document defined in XML an works like a DTD.*
- *XMLs let us define not only elements and attributes, but also the range of values of data types*

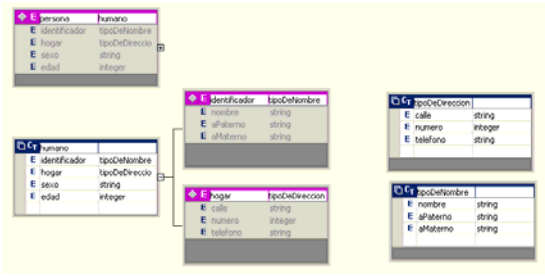
14

XML schema (3)

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:sujeto="http://148.226.81.254/~rocajo16/xml/"
            targetNamespace="http://148.226.81.254/~rocajo16/xml/">
  <xsd:annotation>
    <xsd:documentation xml:lang="es">
      Esto es un mecanismo de documentacion
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType name="tipoDeNombre">
    <xsd:sequence>
      <xsd:element name="nombre" type="xsd:string" />
      <xsd:element name="aPaterno" type="xsd:string" />
      <xsd:element name="aMaterno" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="tipoDeDireccion">
    <xsd:sequence>
      <xsd:element name="calle" type="xsd:string" />
      <xsd:element name="numero" type="xsd:integer" />
      <xsd:element name="telefono" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="humano">
    <xsd:sequence>
      <xsd:element name="identificador" type="sujeto:tipoDeNombre" />
      <xsd:element name="hogar" type="sujeto:tipoDeDireccion" />
      <xsd:element name="sexo" type="xsd:string" />
      <xsd:element name="edad" type="xsd:integer" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="persona" type="sujeto:humano">
  </xsd:element>
</xsd:schema>
```

15

XML schema (4)



16

References

- *Learning XML, Erik Ray, O'Reilly. 2001*
- *Strategic XML, Scott Means, Sams. 2002*
- *XML Pocket Reference 2nd Edition, Eckstein & Casabianca, O'Reilly. 2001*
- *XML Schema, Eric Van der Vlist , O'Reilly, 2002*
- *http://www.w3.org/XML/Schema*
- *http://es.wikipedia.org/wiki/XML*

17