XML

Baltasar Fernández Manjón

Dpto. de Ingeniería del Software e Inteligencia Artificial,
Universidad Complutense de Madrid
Avda. Complutense s/n, 28040, Madrid, Spain,

Ejemplo de documento XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--COMENTARIO documento simple de un fax; archivo fax.xml -->
<!DOCTYPE fax SYSTEM "fax.dtd">
<fax>
    <para> Pilar Little </para>
    <de> Baltasar Fernández </de>
    <fecha>07/07/2000 </fecha>
    <numero> 619 10 19 00 </numero>
    <tema> Curso Interacción Persona-Computador </tema>
    <contenido>
      <parrafo> Querida Pilar, </parrafo>
      <parrafo> Te confirmo nuestra llegada el día 8 a las 8. </parrafo>
      <parrafo> Un saludo  </parrafo>
    </contenido>
</fax>
```

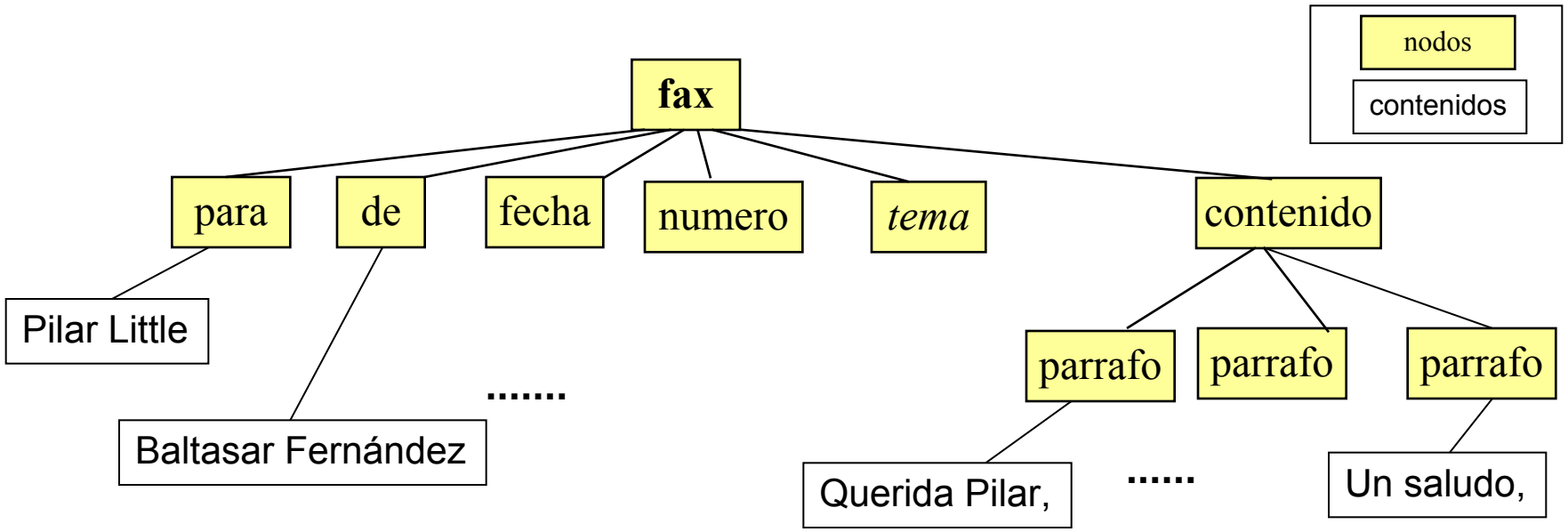
Prólogo

Declaración tipo de documento (opcional)

Cuerpo del Documento (fax es el elemento documento o nodo raíz)

Modelo de datos de un documento XML

- El documento se estructura de forma jerárquica como un **árbol de nodos**
  - Los nodos del árbol son los elementos que conforman el documento
    - Están organizados en niveles denotando composición
  - Las hojas del árbol son los contenidos de dichos elementos



Componentes de un documento XML

- **Prólogo**
  - Declaración XML
  - Instrucciones de procesamiento
- **Contenido**
  - Elementos
    - Etiquetas: etiquetas de inicio, etiquetas de fin
    - Etiquetas de elementos vacíos
    - Atributos
    - PCDATA
    - CDATA
    - Espacios en blanco
- **Comentarios**
- **Notaciones**
- **Entidades**

Elementos

- Componentes básicos de un documento XML:
  - `<identElemento>contenido</identElemento>`
  - `<identElemento atrib1="valor">contenido</identElemento>`
  - `<identElemento/>`

- Encapsulan los contenidos que pueden estar compuestos de:
  - Otros elementos
  - Datos formados por caracteres
  - Referencias a otras entidades
- Los elementos se delimitan mediante etiquetas formadas por el identificador.

Atributos

```
<identElemento atrib1="valor">contenido</identElemento>

<identElemento atrib1="valor"/>
```

- Un elemento puede contener **atributos** que proporcionen **información adicional** sobre dicho elemento
- La especificación de los atributos debe aparecer **sólo dentro de las etiquetas de inicio** o de las etiquetas de elementos vacíos.

Documentos bien formados: Reglas básicas

- El documento **debe tener exactamente un elemento de nivel superior** (elemento documento o **elemento raíz**)

```
<concesionario>
      <coche id="7005APG"/>
      <moto id="8943UJG/>
</concesionario>
```

```
<concesionario>
      <coche id="7005APG"/>
      <moto id="8943UJG/>
</concesionario>
<concesionario>
      <moto id="6312ASL/>
</concesionario>
```

Documentos bien formados: Reglas básicas

- Cada elemento debe tener una **marca de inicio y una marca de fin**
  - Los elementos vacíos pueden indicarse con una marca especial
- El nombre del tipo de elemento de una marca de inicio debe corresponder exactamente con su marca de fin correspondiente
  - En los nombres de los tipos de elementos se distingue entre mayúsculas y minúsculas

```
<concesionario>contenido</concesionario>
```

```
<concesionario/>
```

```
<concesionario></Concesionario>
```

```
<concesionario>
<coche/>
```

```
</concesionario>
```

Documentos bien formados: Reglas básicas

- No puede aparecer un atributo más de una vez, en un mismo elemento
- El valor de los atributos debe ir entre comillas dobles

```
<concesionario marca="seat" ciudad="Madrid"/>
```

```
<concesionario marca="seat" ciudad="Madrid" marc="peugeot">
```

```
<concesionario marca="seat" ciudad="Madrid" marca='peugeot'>
```

Documentos XML válidos

- Un documento XML bien formado que además sigue las reglas especificadas en una Definición de Tipo de Documento (DTD) o por un esquema de documento
  - La DTD define la estructura del documento y los elementos que pueden componer dicho documento

DTD XML simple para un fax

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
   DTD simple para un fax.
-->
<!ELEMENT fax (para, de, fecha, numero, tema?, contenido)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT fecha (#PCDATA)>
<!ELEMENT numero (#PCDATA)>
<!ELEMENT tema (#PCDATA)>
<!ELEMENT contenido (parrafo+)>
<!ELEMENT parrafo (#PCDATA)>
```

Archivo fax.dtd

Documentos XML Válidos

- La DTD sirve para describir los datos. XML está diseñado para que un documento conjuntamente con su DTD sea autodescriptivo y se pueda *validar* automáticamente su corrección
- Una DTD puede declararse dentro de un documento o mediante una referencia externa.
  - Declaración interna al documento:
    - <!DOCTYPE elemento-raíz [declaraciones de elementos]>
  - Es mejor realizar una declaración externa

Declaraciones básicas en una DTD

- Las declaraciones en una DTD tienen la siguiente forma
  - **<!keyword parámetro1 parámetro2 ... parámetroN>**
- Hay 4 palabras reservadas básicas
  - **ELEMENT**
    - Declara un nombre de tipo de elemento y sus posibles subelementos
  - **ATTLIST**
    - Declara los nombres de los atributos de un elemento, así como sus posibles valores y/o valor por defecto
  - **ENTITY**
    - Declara referencias a caracteres especiales o a bloques de texto (similar a un #define en C) o,
    - Declara referencias a contenido repetitivo que puede estar contenido en un recurso externo (similar a un #include en C).

Declaración de elementos en una DTD

- La declaración de un elemento debe tener alguna de las dos siguientes formas
  - <**!ELEMENT** nombre_elemento categoría_contenido>
  - <**!ELEMENT** nombre_elemento (modelo_contenido) cardinalidad>
- Categorías de contenido para elementos
  - **ANY**
    - Puede contener cualquier XML bien formado
  - **EMPTY**
    - No puede contener nada (salvo atributos)
  - **Sólo texto**: PCDATA
    - No puede contener a otros subelementos
  - **Sólo elementos**
    - Contiene únicamente elementos hijos (o subelementos)
  - **Contenido mixto**
    - Puede contener tanto texto como otros elementos

Ejemplo – elemento vacío

Declaración:

```
<!ELEMENT br EMPTY>
```

VÁLIDO:

```
<br/>
<br></br>
<br
></br>
```

NO VÁLIDO:

```
<br>texto</br>
<br><item id="x"/></br>
<br>_</br>
<br>
</br>
```

Modelos de contenido

- Excepto en las categorías ANY o EMPTY es necesario proporcionar un modelo de contenido
- Normalmente un modelo de contenido es una lista de nombres de elementos o el identificador PCDATA encerrados entre paréntesis
  - Pueden aparecer mas paréntesis con propósito de agrupar otros elementos
- Dentro de los modelos de contenido pueden aparecer dos tipos de listas
  - Listas de secuencia
    - Los elementos hijos aparecen en orden separados por comas
  - Listas de elección
    - Sólo puede aparecer uno de los elementos especificados.
    - Los elementos se separan mediante barras verticales

Contenido textual - PCDATA

- Sólo puede incorporar contenido textual y referencia a entidades
  - No puede incluir a otros subelementos

Declaración:

```
<!ELEMENT parrafo (#PCDATA)>
```

Fragmento de documento válido:

```
<parrafo> Querida Pilar, </parrafo>
<parrafo> Te confirmo nuestra llegada el día 8 a las 8.
</parrafo>
<parrafo> Un saludo  </parrafo>
<parrafo> puede incluir referencias a entidades como &amp;amp;  </parrafo>
```

Contenido de subelementos

- Sólo puede contener a otros elementos
  - No puede contener texto fuera de los elementos hijos o subelementos

Declaración:

```
<!ELEMENT fax (para, de, fecha)>
```

Fragmento de documento válido:

```
<fax>
    <para> Pilar Little </para>
    <de> Baltasar Fernández </de>
    <fecha>07/07/2000 </fecha>
</fax>
```

Fragmento de documento no válido:

```
<fax>
    este contenido textual no es
válido
    <para> Pilar Little </para>
    <de> Baltasar Fernández
</de>
    <fecha>07/07/2000 </fecha>
</fax>
```

Contenido mixto

- Puede contener tanto texto como otros elementos
  - Siempre se especifica utilizando una lista de elección
- Siempre que aparezca la palabra clave PCDATA debe ser el primer item del modelo de contenido
- En el modelo de contenido mixto no se puede restringir el número de apariciones de los subelementos

```
<!ELEMENT foo (#PCDATA | bar |otro)*>
```

```
<!ELEMENT foo (bar |#PCDATA  |otro)*>
```

Operadores de cardinalidad

- Los operadores de cardinalidad definen cuantos elementos hijo pueden aparecer en un modelo de contenido
- Operadores
  - Ninguno
    - La ausencia de operador indica que es necesaria y sólo se permite una y sólo una ocurrencia del elemento
  - ?
    - Cero o una instancia del elemento (denota opcionalidad)
  - *
    - Cero o más instancias (opcional y repetible)
  - +
    - Una o más instancias (obligatorio y repetible)

Ejemplo lista de clase

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT listaAlumnos (alumno)+>
<!ELEMENT alumno (nombre, apellidos, nota?, asignatura*)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT nota (#PCDATA)>
<!ELEMENT asignatura (#PCDATA |  curso)*>
<!ELEMENT curso (#PCDATA)>
```

listaClase.dtd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE listaAlumnos SYSTEM "listaClase.dtd">
<listaAlumnos>
    <alumno>
        <nombre>Balta</nombre>
        <apellidos>Fdez Manjón</apellidos>
        <asignatura>paw</asignatura>
        <asignatura>lp3</asignatura> </alumno>
    <alumno>
        <nombre>Antonio</nombre>
        <apellidos>Navarro</apellidos>
        <asignatura>IG1 <curso>2</curso></asignatura></alumno>
</listaAlumnos>
```

Declaración de atributos en una DTD

- La declaración comienza con ATTLIST seguida por el nombre del elemento al que pertenecen los atributos, seguida por la definición de los atributos individuales.
- El orden de los atributos es indiferente

```
<!ATTLIST nombreElemento
      nombreAtributo1   tipoAtributo1 caracteristica1 valorPorDefecto1
.................
      nombreAtributoN  tipoAtributoN caracteristicaN valorPorDefectoN>
```

```
<!ELEMENT elemConAtributos
(#PCDATA)>

<!ATTLIST elemConAtributos
        aaa CDATA #REQUIRED
        bbb CDATA #IMPLIED
        ccc CDATA #FIXED
        “valorPorDefecto” >
```

Características de los atributos

- La característica de los atributos indica si un atributo es necesario o no y como debe comportarse un analizador (parser) si no aparece el atributo en un documento.
- Posibles valores
  - #REQUIRED
    - El atributo debe estar siempre presente en la instancia del documento
  - #IMPLIED
    - El atributo es opcional
    - No se puede asociar un valor por defecto
  - #FIXED
    - El atributo es opcional. Si aparece debe coincidir con el valor por defecto. Si no aparece el parser puede proporcionarle el valor por defecto
  - Valor por defecto (sin palabra clave)
    - El atributo es opcional.
      - Si aparece debe ser un valor adecuado para su tipo de atributo.
      - Si no aparece el parser puede proporcionarle el valor por defecto

Tipos de atributos

- CDATA
  - Datos formados por caracteres (cadenas de texto)
  - Con < y & es necesario usar entidades como carácter de escape (`&` ó `<`)
- Valores enumerados
  - Se enumera el conjunto de los posibles valores permitidos
  - Se puede proporcionar un valor por defecto
- ID
  - Identificador único para cada instancia de elemento.
- IDREF
  - Una referencia a un elemento con un atributo de tipo ID
- IDREFS
  - Una lista de valores IDREF separados por blanco(s)

Tipos de atributos

- NMTOKEN
  - Cadena de texto que cumple las reglas de un identificador
- NMTOKENS
  - Lista de valores NMTOKEN separados por blancos
- ENTITY
  - El nombre de una entidad predefinida
- ENTITIES
  - Lista de nombres de entidades separados por blancos
- NOTATION
  - Un tipo de notación que se declara explícitamente en la DTD

Atributos CDATA

- Un atributo de tipo CDATA puede contener cualquier texto (secuencia de caracteres)
  - Se puede incluir referencias a entidades predefinidas (p. e. <) y a entidades internas
  - No se puede incluir referencias a entidades externas

```
<!ELEMENT elemConAtributos
(#PCDATA)>

<!ATTLIST elemConAtributos
        aaa CDATA #REQUIRED
        bbb CDATA #IMPLIED
        ccc CDATA #FIXED
        “valorPorDefecto” >
```

Atributos con valores enumerados

- Se enumera el conjunto de cadenas de caracteres que forman los posibles valores permitidos (esos valores deben ser tokens solo con caracteres NameChar)
  - Estas cadenas no pueden contener espacios en blanco
- Se puede proporcionar un valor por defecto

```
<!ELEMENT elemento ANY>
<!ATTLIST elemento
    tipo (dificil|fácil) #IMPLIED
    curso (1|2|3|4|5) "1">
.....
<elemento curso="2">contenido del elemento</elemento>
<elemento tipo="fácil"/>
```

Atributo ID – identificador de elemento

- Proporciona un identificador único para una instancia de un elemento
  - Debe ser un identificador XML válido (p.e. No puede empezar por un número y no debe tener espacios en blanco)
  - Debe ser único dentro de un documento
  - Un elemento no puede tener mas de un ID
  - Es
    #I

```
<!ELEMENT alumno (nombre, apellidos, nota?, asignatura*)>
<!ATTLIST alumno
        IDalumno ID #REQUIRED
        repetidor CDATA #REQUIRED
        sexo (h | m)  "h">
.....
<alumno IDalumno="Alumno1" sexo="h" repetidor="no">
            <nombre>Balta</nombre>
            <apellidos>Fdez Manjon</apellidos>
            <asignatura>paw</asignatura>
            <asignatura>Ip3</asignatura>
</alumno>
```

Atributo IDREF / IDREFS – Relaciones entre elementos

- El atributo IDREF se utiliza para establecer relaciones o enlaces desde un elemento a otro con un atributo ID
  - El valor de un IDREF debe ser un identificador XML y debe coincidir con un valor de ID
  - Es posible hacer varias referencias a un mismo ID

```
<!ELEMENT alumno (nombre, apellidos, nota?, asignatura*)>
<!ATTLIST alumno
        IDalumno ID #REQUIRED
        compañero IDREF #REQUIRED
        compañeros IDREFS #REQUIRED
        repetidor CDATA #REQUIRED
        sexo (h | m)  "h">
.....
<alumno IDalumno="Alumno1"
        compañero="Alumno2"
        compañeros="Alumno2 Alumno3 Alumno4"
        sexo="h" repetidor="no">
.....
</alumno>
```

Relación con su compañero de prácticas

Grupos de prácticas de mas de dos alumnos

Atributos NMTOKEN/NMTOKENS

- Similar a CDATA pero impone algunas restricciones sobre el tipo posible de valores
  - No se pueden incluir espacios en blanco ni algunos signos de puntuación (solo caracteres NameChar)

Ejemplo lista de clase con atributos

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT listaAlumnos (alumno)+>
<!ELEMENT alumno (nombre, apellidos, nota?, asignatura*)>
<!ATTLIST alumno
        repetidor CDATA #REQUIRED
        sexo (h | m)  "h">
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellidos (#PCDATA)>
<!ELEMENT nota (#PCDATA)>
<!ELEMENT asignatura (#PCDATA |  curso)*>
<!ELEMENT curso (#PCDATA)>
```

listaClase2.dtd

```
<?xml version="1.0" encoding="ISO-8859-1“ standalone=“no”?>
<!DOCTYPE listaAlumnos SYSTEM "listaClase2.dtd">
<listaAlumnos>
        <alumno sexo="h" repetidor="no">
                <nombre>Balta</nombre>
                <apellidos>Fdez Manjon</apellidos>
                <asignatura>paw</asignatura>
                <asignatura>Ip3</asignatura></alumno>
        <alumno repetidor="dos veces" >
                <nombre>Antonio</nombre>
                <apellidos>Navarro</apellidos>
                <asignatura>IG1 <curso>2</curso></asignatura></alumno>
</listaAlumnos>
```