UF1 – Programació amb XML

**Nucli Formatiu 1 – Definició d'esquemes i vocabularis en XML - DTD**

**Que son els DTD?**

DTD significa “Document Type Definition”

És una mena de diccionari que especifica com ha de construir-se un XML.

Concreta dues coses fonamentals:

1. Quins elements han de ser acceptats.
2. En quines posicions han d'estar aquests elements dins el XML.

**Justificació**

Un document XML fixe pot anar viatjant entre emissor i receptor.

Però quan el XML no és fixe sinó que es pot modificar per part d'algú, és important saber el format del fitxer: quin elements i etiquetes són valides i quines regles sintàctiques ha de complir el XML. Aquesta informació li interessarà al receptor per tal de **validar** el document XML rebut.

Exemple d'un XML i el seu DTD

XML

<?xml version=”1.0” encoding=”UTF-8”?>

<!DOCTYPE BaseDadesWhatsapp SYSTEM “a.dtd”>

<BaseDadesWhatsapp>

<whatsapp>

<telefon>600 123 456</telefon>

<data>01/09/2015</data>

<hora>23:02</hora>

<missatge>Hola, ja he tornat a la ciutat</missatge>

</whatsapp>

<whatsapp>

<telefon>600 123 456</telefon>

<data>15/09/2015</data>

<hora>16:00</hora>

<missatge>comencem avui a l'institut!</missatge>

</whatsapp>

</BaseDadesWhatsapp>

DTD

<!ELEMENT BaseDadesWhatsapp (whatsapp\*)>

<!ELEMENT whatsapp (telefon,data,hora,missatge)>

<!ELEMENT telefon (#PCDATA)>

<!ELEMENT data (#PCDATA)>

<!ELEMENT hora (#PCDATA)>

<!ELEMENT missatge (#PCDATA)>

Fixar-se a la línia:

<!DOCTYPE BaseDadesWhatsapp SYSTEM “BaseDadesWhatsapp.dtd”>

especifica que el DTD es troba fora del document.

**Sintaxi per definir el DTD a dins el mateix document**: primer la DTD i després el XML:

<?xml version=”1.0”?>

<!DOCTYPE BaseDadesWhatsapp

[

<!ELEMENT BaseDadesWhatsapp (whatsapp\*)>

<!ELEMENT whatsapp (telefon,data,hora,missatge)>

<!ELEMENT telefon (#PCDATA)>

<!ELEMENT data (#PCDATA)>

<!ELEMENT hora (#PCDATA)>

<!ELEMENT missatge (#PCDATA)>

]>

<BaseDadesWhatsapp>

<whatsapp>

<telefon>600 123 456</telefon>

<data>01/09/2015</data>

<hora>23:02</hora>

<missatge>Hola, ja he tornat a la ciutat</missatge>

</whatsapp>

<whatsapp>

<telefon>600 123 456</telefon>

<data>15/09/2015</data>

<hora>16:00</hora>

<missatge>comencem avui a l'institut!</missatge>

</whatsapp>

</BaseDadesWhatsapp>

|  |
| --- |
| **Primer exercici**:  Fer el DTD pel següent xml: <?xml version=”1.0”?>  <persona>  <nom>A</nom>  </persona> |

**Blocs per construir una DTD i seqüències d'elements**

Elements i Atributs

Entitat: diguem que és un caràcter amb un significat especial.

Entitats personalitzables: primer definició i segon utilització:

<!ENTITY pi “3.14159265358979323846”> => &pi;

<!ENTITY fitxer SYSTEM “fitxerdetext.txt”> => &fitxer;

<!ENTITY web SYSTEM “http://www.google.es”> => &web;

PCData: Parsed Character Data. Indica que entre la etiqueta d'obertura i tancament d'aquell element, s'emmagatzemaran caràcters com a text i seran analitzats per un parser. Això implica que si al contingut hi ha entitats i elements es tractaran com a tals i no com a informació.

CDATA: Character Data. Igual a l'anterior però el seu contingut no serà analitzat pel parser. Això implica que si al contingut hi ha entitats i elements es tractaran com a informació.

**Definició d'elements**:

<!ELEMENT nom\_elem (nodefill1,nodefill2,...nodefillN)>

nom\_elem és el node fill i la llista entre parèntesi són els seus fills que han d'anar separats mitjançant comes.

Atenció a l'espai en blanc que és obligatori entre el nom de l'element i el parèntesi de començament!

Després es defineix un a un el tipus de cada element fill. Si algú d'aquests elements ha d'estar buit hi ha la paraula reservada EMPTY:

<!ELEMENT saltdelinia EMPTY> => <saltdelinia/>

**Definició d'elements simples**

<!ELEMENT mobil (#PCDATA)> utilitzarem només aquest tipus d'element simple

<!ELEMENT hora (CDATA)>

<!ELEMENT elem\_lliure (#ANY)> això és qualsevol combinació d'elements parseables o no

Atenció a l'espai en blanc que és obligatori entre el nom de l'element i el parèntesi de començament.

**Opcionalitat entre 2 elements**

<!ELEMENT nom\_elem (nodefill1, (nodefill2|nodefill3))>

declara els elements fills nodefill1 i o bé nodefill2 o bé nodefill3

exemple: persona (telefonfix|telefonmobil)

Com especifiquem a la definició d'un determinat element que aquest té una o més ocurrències?; o cero o més ocurrències?; o que únicament pot haver-hi 0 o 1? ocurrències?. Posant darrera el seu nom els següents caràcters:

* '' (res) : l'element apareix obligatòriament una vegada
* '+': pot haver 1 o més ocurrències de l'element.
* '\*': pot haver 0 o més ocurrències de l'element.
* '?': pot haver 0 o 1 ocurrència de l'element.

**Definició d'atributs**

<!ATTLIST nom\_elem nom\_atrib tipus\_atrib valor\_x\_defecte>

Atenció: el primer paràmetre que s'escriu darrera d'ATTLIST és el nom de l'element i no pas el nom de l'atribut.

exemple:

<!ATTLIST pagament tipus CDATA “targeta”>

es correspondria amb l'XML:

<pagament tipus=”targeta” />

Podem definir 1 o més atributs a dins d'un mateix “ATTLIST”. La sintaxis per definir-ne més d'un és:

<!ATTLIST pagament

tipus CDATA “targeta”

banc CDATA “”

data CDATA “”

hora CDATA “”

>

es correspondria amb l'XML:

<pagament tipus=”targeta” banc=”caixa terrassa” data=”01/10/2015” hora=”12:49” />

Els **tipus d'atribut** poden ser:

* **CDATA**: Text
* **ID**: es un identificador únic que ha de ser únic per aquell element en tot el XML. Es una translació directa de la clau principal a bases de dades.

Atenció al fet que el llenguatge DTD exigeix que el valor d'un atribut de tipus ID sigui un nom vàlid en XML, és a dir, no pot començar per un número.

* **IDREF**: es un identificador d'altre element del propi document XML. O sigui l'atribut ID utilitzat en altre element. Equivalent a una clau forana de bases de dades.
* **IDREFS**: es una llista d'identificadors d'altres elements del propi document XML.

**(ASIX)**

* **(tipus1 | tipus2 | tipus3 | ...)**: el valor es un dels indicats en aquesta llista de tipus d'atributs. Es l'equivalent en programació a un tipus enumerat.

Exemple:

<!ATTLIST pagament tipus (efectiu | targeta) “targeta”>

<pagament tipus=”efectiu” />

<pagament tipus=”targeta” />

* **NMTOKEN**: es un nom vàlid en XML, lo qual significava que només pot contenir : lletres, números, guió mig - , subratllat \_ , punt . i dos punts :

No pot contenir cap espai en blanc.

* **NMTOKENS**: es una llista de noms XML vàlids separats per un espai.
* ENTITY: el tipus de l'atribut és una entitat que s'ha declarat anteriorment.
* ENTITIES: es una llista d'entitats.

El valor\_x\_defecte pot ser qualsevol dels següents:

* **“un valor concret”**, que serà el valor de l'atribut
* **#REQUIRED**: obligatori, l'atribut ha de tenir un valor però no s'especifica quin. Exemple: DNI a persona.
* **#IMPLIED**: optatiu. Exemple: Passaport a persona.
* **#FIXED** : fixe, i qui fa el XML no ho podrà canviar.

És a dir, no és obligatori escriure en tots els casos un valor per defecte perquè, de fet, moltes vegades no té sentit.

Exemple:

<!ATTLIST institut nom CDATA #FIXED “Nicolau Copernic”>

<institut nom=”Nicolau Copernic” />

|  |
| --- |
| **Segon exercici**:  Crear una biblioteca de llibres en un XML. Crear-li un DTD per tal que emmagatzemi els següents camps d'un llibre:   * Codi de llibre * Títol * Editorial * Edició * ISBN * Número de pàgines * Autor |

Referència de Software d'edició/validació:

* XMLCopyEditor
* XMLNotepad,
* TotalEdit,
* SernaFree,
* xmlPro (Vervet Logic)
* <oXygen/>

Referència de Validadors online de XML:

* The W3C Markup Validation Service
* XML Validation
* Validome