



Validació de documents amb esquemes XML

UF1: Programació amb XML



Objectius de la presentació

Durant aquesta presentació estudiarem la sintaxi utilitzada per validar la informació estructurada en un document XML mitjançant esquemes XML



Introducció a la validació d'XML

- Un document XML amb la sintaxi correcta és un document **ben estructurat**
- Un document XML ben estructurat **no és un document XML vàlid**
- Hi ha 2 formes de definir tipus de documents XML:
 - **DTD**: forma original de definició de tipus de document
 - **Esquemes XML (XML Schema)**: alternativa als DTD basada en XML



Introducció a XML Schema

- Un esquema XML s'emmagatzema amb extensió **.xsd** (que significa definició d'esquema XML)
- Un esquema XML descriu **l'estructura d'un document XML**
- És a dir, descriu quins tipus d'elements pot contenir un element, quants subelements pot contenir, els valors que es poden col·locar en qualsevol element o atribut, etc.





Introducció a XML Schema

- XML Schema és actualment l'**estàndard de facto** per descriure documents XML
- Hi ha dues versions (1.0 i 1.1) que són, en general, les mateixes
- Un esquema XML és un **document XML**; existeix, fins i tot, un esquema XML per descriure l'estàndard XSD
- L'estàndard XSD està controlat pel **W3C**
- Ha evolucionat durant diversos anys, és extremadament **complet** i, com a resultat, s'ha tornat **força complex**





Exemple d'XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="libro">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="isbn" type="xs:string"/>
        <xs:element name="titulo" type="xs:string"/>
        <xs:element name="precio" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



Avantatges dels esquemes XML

- El propòsit d'un esquema XML és definir els blocs de construcció correctes d'un document XML:
 - Els elements i atributs que poden aparèixer en un document
 - La quantitat, i l'ordre, d'elements secundaris
 - Tipus de dades per als elements i atributs
 - Valors predeterminats i fixos pels elements i atributs



Avantatges dels esquemes XML

- Un dels punts forts dels esquemes XML és el suport als tipus de dades:
 - És més fàcil descriure el contingut permès del document
 - És més fàcil definir la validesa de les dades
 - És més fàcil definir restriccions a les dades
 - És més fàcil definir formats de dades
 - És més fàcil convertir la informació a altres tipus de dades



Avantatges dels esquemes XML

- Els esquemes XML són extensibles (són documents XML)
- Amb una definició d'esquema extensible, podem:
 - Reutilitzar l'esquema per altres esquemes
 - Crear els nostres propis tipus de dades, derivats dels tipus estàndard
 - Escriure diversos esquemes en el mateix document



Avantatges dels esquemes XML

- Al enviar dades en format XML, és essencial que l'emissor i el receptor tinguin les **mateixes "expectatives"** sobre el contingut
- Amb els esquemes XML, l'emissor pot descriure les dades d'una manera que el receptor comprendrà
- Per exemple, **"03-11-2004"** s'interpretarà en alguns països com a *3 de Novembre*, però en altres com a *11 de Març*



Avantatges dels esquemes XML

- Un element XML, validat amb un esquema que defineixi el tipus **date** per aquesta informació, garanteix una comprensió mútua del contingut
- El tipus de dades **date** requereix que el format de la data sigui **AAAA-MM-DD**
- Aleshores, seria impossible que el document XML estigués validat sense fer servir aquest format de data



Estructura del fitxer XSD

- Tots els fitxers XSD haurien de tenir una estructura similar a aquesta:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
...
```

```
</xs:schema>
```





Estructura del fitxer XSD

- La primera línia indica que un fitxer XSD té una estructura de fitxer XML
- L'element arrel **xs:schema** ens indica que és un esquema XSD
- El fragment **xmlns:xs=...** indica que els elements i els tipus de dades utilitzats en aquest esquema provenen de l'espai de noms **<http://www.w3.org/2001/XMLSchema>**
- També especifica que els elements i els tipus de dades d'aquest espai de noms han d'estar prefixats amb **xs:**



Referència a un esquema en un document XML

```
<?xml version="1.0" encoding="UTF-8"?>

<nota      xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"  xsi:noNamespaceSchemaLocation="nota.xsd">

<nota>

    <destino>María</destino>

    <origen>Jordi</origen>

    <titulo>Recordatorio</titulo>

    <cuerpo>No      olvides      traer      el
informe</cuerpo>

</nota>
```



Referència a un esquema en un document XML

- Aquest document XML té una referència a un esquema XML
- **xmlns:xsi** indica que aquest document s'ha de validar amb un XML Schema
- **xsi:noNamespaceSchemaLocation="nota.xsd"** especifica on podem trobar l'esquema (en aquest cas, a la mateixa carpeta que l'arxiu XML)



Elements simples

- Els elements són el bloc principal de qualsevol document XML; contenen les dades i determinen l'estructura del document
- Es pot definir un element dins d'un esquema XML de la següent manera:

```
<xs:element name="xxx" type="yyy"/>
```

- A on xxx és el nom de l'element i yyy és el tipus de dades de l'element



Elements simples

- XML Schema té molts tipus de dades incorporats, els més comuns són:
 - **xs:string**
 - **xs:decimal** → el punt (.) com a separador decimal
 - **xs:integer**
 - **xs:boolean** → només admet els valors *true* o *false*
 - **xs:date** → dates en format *AAAA-MM-DD*
 - **xs:time** → hores en format *HH:MM:SS*



Elements simples

- Si s'estableix la propietat de tipus d'un element a l'esquema, el valor corresponent al document XML ha d'estar en el **format correcte**
- Si no es fa això, es produirà un **error de validació**
- Es poden crear tipus definits per l'usuari mitjançant les etiquetes `<xs:simpleType>` o `<xs:complexType>`



Valors predeterminats i fixos

- Als elements simples es pot especificar un valor **predeterminat** o un valor **fix**
- El valor predeterminat s'assigna automàticament quan **no s'especifica cap** altre

```
<xs:element name="color" type="xs:string" default="red"/>
```

- Un valor fix s'assigna automàticament, però **no permet** especificar cap altre valor

```
<xs:element name="color" type="xs:string" fixed="red"/>
```



Cardinalitat dels elements

- Els atributs **minOccurs** i **maxOccurs** permeten especificar quantes vegades ha d'aparèixer, o pot aparèixer, un mateix tipus d'element
- A **minOccurs** podem assignar un valor **enter no negatiu**
- A **maxOccurs** podem assignar un valor **enter no negatiu** o la constant ***unbounded*** (no hi ha límit)
- Per defecte, **minOccurs** i **maxOccurs** valen **1**

```
<xs:element      name="numcomanda"      type="xs:integer"  
minOccurs="0" maxOccurs="unbounded"/>
```



Restriccions

- Per definir un interval de valors hem d'especificar els límits amb **minInclusive** i **maxInclusive**

```
<xs:element name="edat">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="18"/>
      <xs:maxInclusive value="65"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```



Restriccions

- Per restringir el contingut a un conjunt de valors acceptables, utilitzaríem la restricció d'**enumeració**

```
<xs:element name="transport">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="cotxe"/>
      <xs:enumeration value="autobus"/>
      <xs:enumeration value="tren"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```



Restriccions

- Podem restringir la **longitud mínima i màxima** d'un contingut de tipus cadena de text

```
<xs:element name="password">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:minLength value="5"/>  
      <xs:maxLength value="8"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```



Restriccions

- També podem restringir el contingut definit un **patró** de lletres i números
- Aquesta expressió estarà dividida en 2 parts:
 - Caràcters permesos (entre [])
 - Mínim i màxim de caràcters permesos (entre { })

```
<xs:pattern value="[abcd]{4,8}"/>
```

- Aquesta definició combinada, de caràcters permesos amb longitud mínima i màxima, es pot repetir varies vegades en un mateix patró

```
<xs:pattern value="[abcd]{4,8}[efg]{1,5}[hi]{2,6}"/>
```




Restriccions

- És possible definir un rang de caràcters permesos (per exemple, **[a-z]**)
- Per indicar que es permet qualsevol caràcter introduïrem un punt (.) sense **[]**
- No és obligatori definir longitud màxima de caràcters (per exemple, **{6,}**)
- Podem definir una llista de valors acceptables amb un patró, separant-los amb el caràcter **|**



Exemples de restriccions amb patró

```
<xs:element name="matricula">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{4}[A-Z]{3}" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```



Exemples de restriccions amb patró

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9]{8,}" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```



Exemples de restriccions amb patró

```
<xs:element name="sexe">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:pattern value="home|dona" />  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```



Tipus complexes d'elements

- Un tipus complex és un **contenedor** per a altres definicions d'elements
- Podem especificar quins **elements secundaris** pot contenir un element
- D'aquesta manera definirem quina ha de ser l'**estructura jeràrquica** que han de tenir els documents XML





Exemple d'element complex

```
<xs:element name="facturalinia">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="data" type="xs:date"/>  
      <xs:element name="article" type="xs:string"/>  
      <xs:element name="preu" type="xs:integer"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



Exemple d'element complex

- En aquest exemple, *facturalinia* és un tipus complex que inclou una data, un article i un preu
- Tots 3 subelements són **obligatoris** pel fitxer XML
- Els subelements s'han d'especificar de forma seqüencial, en el **mateix ordre** que s'indica a l'esquema XML



Indicadors de tipus complexes

- Hi ha 3 tipus d'elements complexes:
 - **<xs:sequence>**: els elements secundaris del document XML han d'aparèixer en l'**ordre** que es declaren a l'esquema
 - **<xs:choice>**: **només un** dels elements secundaris descrits a l'esquema pot aparèixer al document XML
 - **<xs:all>**: els elements secundaris descrits a l'esquema poden aparèixer en el document XML, ocupant **qualsevol ordre**



Exemple d'indicadors de tipus complexes

```
<xs:element name="carrec">
  <xs:complexType>
    <xs:choice>
      <xs:element name="professor" type="xs:boolean"/>
      <xs:element name="coordinador" type="xs:boolean"/>
      <xs:element name="capestudis" type="xs:boolean"/>
      <xs:element name="director" type="xs:boolean"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```



Reutilitzar un element complex

- Es pot reutilitzar un element complex indicant en el tipus el nom d'aquell element complex

```
<xs:element name="empleat">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="nom" type="xs:string"/>  
      <xs:element name="ocupacio" type="carrec"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



Reutilitzar un element complex

```
<xs:complexType name="carrec">  
  <xs:choice>  
    <xs:element name="professor" type="xs:boolean"/>  
    <xs:element name="coordinador" type="xs:boolean"/>  
    <xs:element name="capestudis" type="xs:boolean"/>  
    <xs:element name="director" type="xs:boolean"/>  
  </xs:choice>  
</xs:complexType>
```



Atributs

- Els elements simples **no poden tenir atributs**
- Si un element té atributs, es considera que és un **tipus complex**
- L'atribut, però, sempre es declara com un tipus simple

```
<xs:attribute name="xxx" type="yyy"/>
```

- A on xxx és el nom de l'atribut i yyy especifica el tipus de dades de l'atribut



Atributs

- Els tipus de dades que podem fer servir són els mateixos que a l'hora de definir elements (**xs:string**, **xs:decimal**, **xs:integer**, **xs:boolean**, **xs:date**, **xs:time**, etc.)
- Podem indicar la obligatorietat de l'ús d'un atribut afegint **use="required"**

```
<xs:attribute name="id" type="xs:string" use="required"/>
```



Elements buits amb atributs

- Per afegir, al document XML, un element buit que contingui atributs, caldrà definir-lo a l'esquema com un element complex
- Lògicament, no es definirà cap altre element al seu interior

```
<xs:element name="producte">  
  <xs:complexType>  
    <xs:attribute name="codi" type="xs:integer"/>  
  </xs:complexType>  
</xs:element>
```



Elements buits amb atributs

- L'element definit a l'exemple anterior el farem servir, en un document XML, de la següent manera:

```
<producte codi="1345"/>
```

