

XML Schema

1 XML Schema

Il linguaggio **XML Schema** (chiamato anche *XSchema*) è più espressivo rispetto a DTD:

- utilizza i namespace;
- ha un grande potere espressivo riguardo ai tipi di dati;
- comprende dei concetti object-oriented (ad esempio l'ereditarietà).

Inoltre, a differenza di DTD, uno schema definito con XML Schema è serializzato (scritto) in XML.

Tuttavia, questo linguaggio è molto complesso. Perciò, in pratica, esso viene solitamente utilizzato per circa il 20 % delle sue possibilità.

2 Componenti principali di uno schema

I principali tipi di componenti usati per definire gli schemi con XML Schema sono:

Definizione di tipo semplice: un tipo semplice definisce un insieme di valori (che sono stringhe di caratteri). XML Schema prevede dei tipi semplici *primitivi*, che sono predefiniti, mentre negli schemi si possono creare tipi semplici *derivati* (restrizioni sui valori dei tipi primitivi: ad esempio, i numeri interi compresi in un determinato intervallo).

Definizione di tipo complesso: un tipo complesso definisce la struttura degli elementi di tale tipo, mediante la specifica di una collezione di requisiti su attributi, sotto-elementi e dati carattere.

Dichiarazione di elemento: associa il nome di un elemento a un tipo semplice o complesso. Affinché siano valide, le istanze di questo elemento nel documento dovranno soddisfare tutti i requisiti definiti dal tipo dell'elemento; se tutti gli elementi nel documento sono validi allora lo è l'intero documento.

Dichiarazione di attributo: associa un nome di attributo a un tipo semplice (siccome i valori degli attributi contengono solo testo, non è possibile usare tipi complessi).

3 Esempio

Il seguente schema (che sarà contenuto in un file con estensione `.xsd`) descrive un elemento di tipo complesso:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:s="[URI del namespace]"
            targetNamespace="[URI del namespace]">

  <xsd:simpleType name="Voto">
    <xsd:restriction base="xsd:integer">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="100"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="Studente">
    <xsd:attribute ref="s:id" use="required"/>
    <xsd:attribute ref="s:voto" use="required"/>
  </xsd:complexType>

  <xsd:element name="studente" type="s:Studente"/>
  <xsd:attribute name="id" type="xsd:string"/>
  <xsd:attribute name="voto" type="s:Voto"/>

</xsd:schema>
```

- L'attributo `targetNamespace` specifica l'URI del namespace da associare allo schema che si sta definendo. Separatamente, si dichiara l'uso di questo stesso namespace (assegnando a esso, in questo esempio, il prefisso `s`), in modo da potervi fare riferimento all'interno dello schema. È necessario dichiarare anche l'uso del namespace di XML Schema (qui si è scelto il prefisso `xsd`).
- `Voto` è un tipo semplice derivato, definito come la restrizione del tipo predefinito `xsd:integer` (corrispondente ai numeri interi) ai valori compresi tra 0 e 100 (estremi inclusi).
- `Studente` è un tipo complesso: gli elementi ai quali esso verrà assegnato avranno due attributi, `id` e `voto`, entrambi obbligatori (`required`).
- Si dichiarano l'elemento `studente`, del tipo complesso `Studente`, e gli attributi `id` e `voto`, rispettivamente di tipo `xsd:string` (il tipo semplice predefinito corrispondente alle stringhe) e `Voto`.

Un esempio di documento XML valido per lo schema precedente è:

```
<?xml version="1.0" encoding="UTF-8"?>  
<s:studente xmlns:s="[URI del namespace]"  
  s:id="ABC1747"  
  s:voto="97"/>
```