XML
eXtensible Markup Language

Schemi – parte b
XMLSchema

versione 2.0

G. Mecca – Università della Basilicata – mecca@unibas.it

Sommario

- XML Schema

XML Schema

- Standard più recente del consorzio
  - ⇨raccomandazione del Maggio 2001
- Livello base:
  - ⇨“DTD con tipi di dato”
  - ⇨in sostanza, un sistema ricchissimo di tipi
- Livello avanzato:
  - ⇨meccanismi per il riuso di schemi
  - ⇨vincoli di integrità avanzati

XML Schema

- Attenzione
  - ⇨si tratta di uno standard molto complesso
- Due parti
  - ⇨XML Schema Part 1: Structures (normativo)
  - ⇨XML Schema Part 2: Datatypes (normativo)
- Lo standard è intricato
  - ⇨molto verboso e sintattico
  - ⇨XML Schema Part 0: Primer (non normativo)

XML Schema

- L'opinione diffusa
  - ⇨è che la complessità dello standard sia sfuggita di mano al comitato che lo ha definito
- Infatti, nella rete
  - ⇨circola una certa ironia sulla complessità dello standard

XML Schema

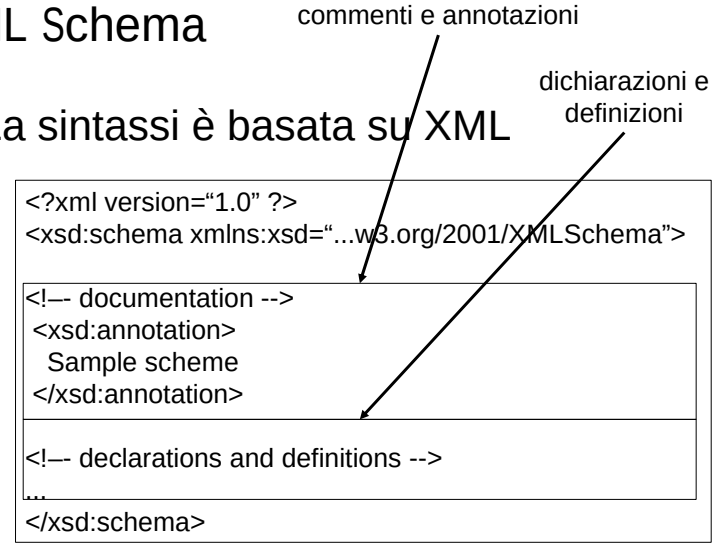
- In occasione della "Second Edition"
  - ⇨revisione correttiva del 2004
  - ⇨Rick Jelliffe scrive su www.oreillynet.com

Should you go rushing out to read it all? Well, if you enjoyed reading it the first time around, all your favourite passages will still be there! I have heard that the film rights have being negotiated for an epic trilogy, but the producers are not sure whether to make it in the style of the Matrix (programmer wakes up in a nightmare world, utterly impenetrable, sponsored by Oracle), Lord of The Ring (enormous scale, but only the innocent can handle something so powerful without being corrupted) or Harry Potter (problems solving with a magic wand, but full of goblins).

Fonte: http://www.oreillynet.com/pub/wlg/4624

XML Schema

- La sintassi è basata su XML



XML Schema

- Il documento
  - ⇨normalmente ha estensione .xsd
- L'elemento principale
  - ⇨schema
  - ⇨definisce anche il namespace standard di XMLSchema
  - ⇨xmlns:xsd=http://www.w3.org/2001/XMLSchema
  - ⇨tutti gli elementi sono annotati con namespace standard

XML Schema

- Definizioni e dichiarazioni
  - ⇨per capire la sintassi e la semantica è utile un parallelo con i linguaggi di programmazione
- Nei linguaggi
  - ⇨si possono definire tipi
  - ⇨e poi dichiarare variabili dei tipi definiti
- In XMLSchema
  - ⇨i tipi servono a descrivere gli elementi

XML Schema

- Alcuni esempi in C++
  - ⇨struct punto {float x, y, z}; // definizione di tipo
  - ⇨punto p1, p2; // dichiarazione
  - ⇨typedef int arrayDiInteri[10]; // definizione
  - ⇨arrayDiInteri vettore; // dichiarazione
- Nota
  - ⇨posso anche definire e dichiarare assieme
  - ⇨es: int vettore[10];

XML Schema

- In XMLSchema
  - ⇨lo schema è essenzialmente una collezione di dichiarazioni di elemento e di attributo
  - ⇨per ciascuna dichiarazione è necessario specificare un tipo
  - ⇨il tipo può essere definito separatamente oppure contestualmente

Esempio

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<xs:element name="domanda" type="xs:string"/>
```

dichiarazione di elemento di tipo semplice

```
<xs:complexType name="argomentoType">
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="nome" type="xs:string" use="required"/>
</xs:complexType>
```

definizione di tipo complesso

```
<xs:complexType name="listaArgomentiType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="argomento" type="argomentoType"/>
  </xs:sequence>
</xs:complexType>
```

dichiarazione di elemento di tipo complesso

dichiarazione di elemento con definizione contestuale del tipo

```
<xs:element name="questionario">
<xs:complexType>
  <xs:sequence>
    <xs:element name="listaArgomenti" type="listaArgomentiType"/>
    <xs:element name="listaDifficolta" type="listaDifficoltaType"/>
    <xs:element name="listaQuesiti" type="listaQuesitiType"/>
  </xs:sequence>
```

...

XML Schema

- Più di 40 tipi di base
- Tipi per elementi ed attributi
  - ⇨`xsd:string`
  - ⇨`xsd:integer, xsd:float`
  - ⇨`xsd:date, xsd:time`
  - ⇨`xsd:URIReference`
- Tipi per attributi
  - ⇨`xsd:id, xsd:idref`
  - ⇨`xsd:NMTOKEN`

XML Schema

- Oltre ai tipi di base esistenti
  - ⇨è possibile definire altri tipi semplici
- In particolare
  - ⇨tipi intervallo
  - ⇨tipi enumerati
  - ⇨tipi “pattern”

XML Schema

Tipo intervallo:

```
<xsd:simpleType name="tipoEta" base="xsd:integer">
  <xsd:minInclusive value="0"/>
  <xsd:maxInclusive value="120"/>
</xsd:simpleType>
```

Tipo enumerato:

```
<xsd:simpleType name="tipoSesso" base="xsd:string">
  <xsd:enumeration value="maschio"/>
  <xsd:enumeration value="femmina"/>
</xsd:simpleType>
```

Tipo pattern:

```
<xsd:simpleType name="tipoCodiceFiscale" base="xsd:string">
  <xsd:pattern value="[A-Z]{6}\d\d[A-Z]\d\d[A-Z]\d{3}[A-Z]"/>
</xsd:simpleType>
```


XML Schema

- Tipi complessi ("complexType")
  - ⇨costruiti a partire dai tipi di base utilizzando vari costrutti
- Sistema di costrutti più ricco del DTD
  - ⇨sequenza ("sequence")
  - ⇨alternativa ("choice")
  - ⇨cardinalità ("minOccurs", "maxOccurs")
  - ⇨insieme ("all")

XML Schema

- Esempio
  - ⇨il costruttore sequence; può essere omesso

```
<xsd:complexType name="tipoPersona">
   <xsd:sequence>
      <xsd:element name="nome"  type="xsd:string"/>
      <xsd:element name="cognome" type="xsd:string"/>
      <xsd:element name="eta"   type="xsd:integer"/>
   </xsd:sequence>
   <xsd: attribute name="sesso" type="xsd:string"/>
</xsd:complexType>
```


Sequence

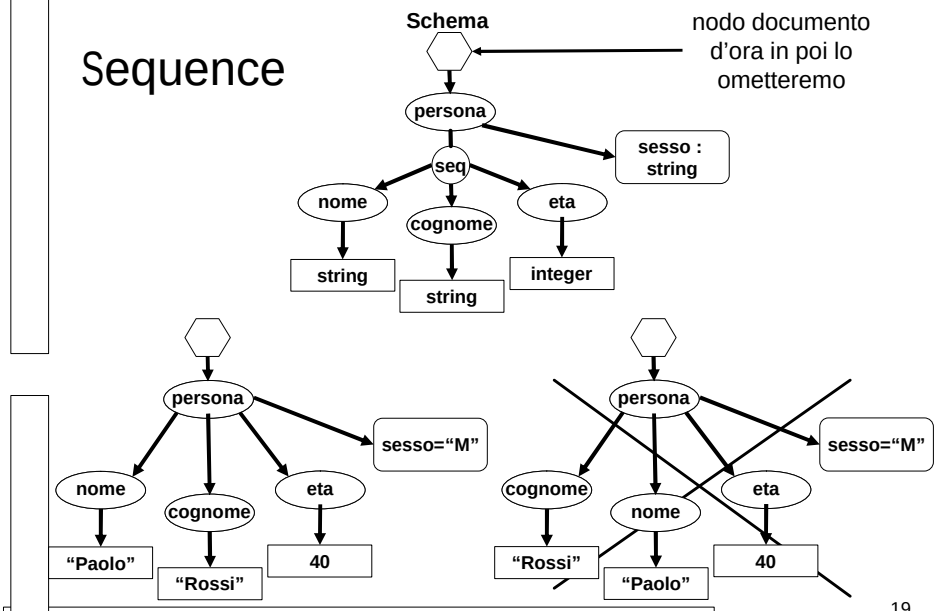
elenca i sottoalberi ed esprime un vincolo sull'ordine;
sequence è stato omesso

```
<xsd:complexType name="tipoPersona">
      <xsd:element name="nome"  type="xsd:string"/>
      <xsd:element name="cognome" type="xsd:string"/>
      <xsd:element name="eta"   type="xsd:integer"/>
   <xsd: attribute name="sesso" type="xsd:string"/>
</xsd:complexType>

<xsd:element name="persona" type="tipoPersona"/>
```

```
<?xml version="1.0" ?>
<persona sesso="M">
  <nome>Paolo</nome>
  <cognome>Rossi</cognome>
  <eta>40</eta>
</persona>
```

```
<?xml version="1.0" ?>
<persona sesso="M">
  <cognome>Rossi</cognome>
  <nome>Paolo</nome>
  <eta>40</eta>
</persona>
```

Sequence



All

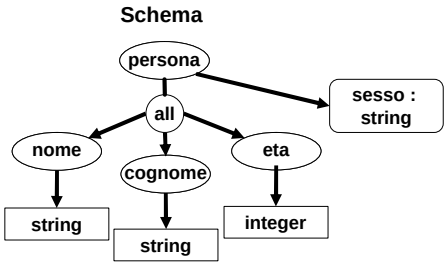
elenca i sottoalberi ma NON esprime un vincolo sull'ordine

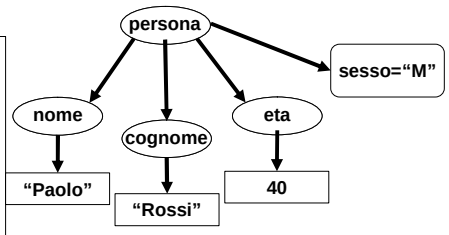
```
<xsd:complexType name="tipoPersona">
  <xsd:all>
     <xsd:element name="nome"  type="xsd:string"/>
     <xsd:element name="cognome" type="xsd:string"/>
     <xsd:element name="eta"   type="xsd:integer"/>
  </xsd:all>
  <xsd: attribute name="sesso" type="xsd:string"/>
</xsd:complexType>

<xsd:element name="persona" type="tipoPersona"/>
```

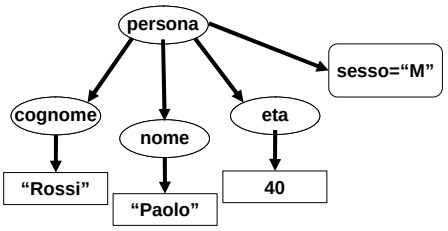
```
<?xml version="1.0" ?>
<persona sesso="M">
  <nome>Paolo</nome>
  <cognome>Rossi</cognome>
  <eta>40</eta>
</persona>
```

```
<?xml version="1.0" ?>
<persona sesso="M">
  <cognome>Rossi</cognome>
  <nome>Paolo</nome>
  <eta>40</eta>
</persona>
```

All







Choice

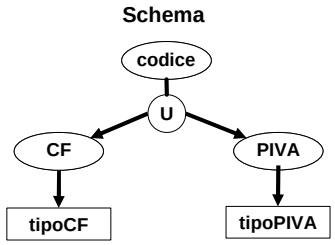
elenca i possibili sottoalberi in alternativa tra di loro

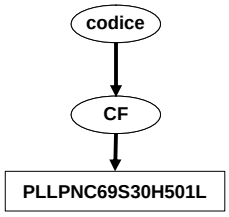
```
<xsd:complexType name="tipoCodice">
    <xsd:choice>
        <xsd:element name="CF"   type="tipoCodiceFiscale"/>
        <xsd:element name="PIVA" type="tipoPartitaIva"/>
    </xsd:choice>
</xsd:complexType>

<xsd:element name="codice" type="tipoCodice"/>
```

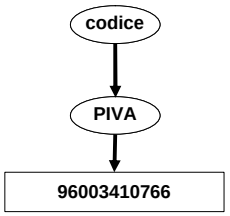
```
<codice>
  <CF>
    PLLPNC69S30H501L
  </CF>
</codice>
```

```
<codice>
  <PIVA>
    96003410766
  </PIVA>
</codice>
```

Choice







XML Schema

- Nidificazione
  - ⇨i costruttori di tipo possono essere liberamente nidificati
  - ⇨sequenza dentro sequenza, choice dentro sequenza ecc.
- Limiti
  - ⇨il costrutto all non può essere nidificato in altri costrutti

Nidificazione

esempio di definizione anonima di tipo

```
<xsd:element name="persona“>
 <xsd:complexType>
    <xsd:element name=“nome"  type="xsd:string"/>
    <xsd:element name=“cognome“ type="xsd:string"/>
    <xsd:element name=“eta"   type="xsd:integer"/>
    <xsd:element name=“indirizzo”>
    <xsd:complexType>
       <xsd:element name=“via“ type="xsd:string"/>
       <xsd:element name=“citta“   type="xsd:string"/>
   </xsd:complexType>
   </xsd:element>
</xsd:complexType>
</xsd:element>
```


Nidificazione

```
<xsd:complexType name=“tipoIndirizzo”>
   <xsd:element name=“via“ type="xsd:string"/>
   <xsd:element name=“citta“   type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name=“tipoPersona“>
  <xsd:element name=“nome"  type="xsd:string"/>
  <xsd:element name=“cognome“ type="xsd:string"/>
  <xsd:element name=“eta"   type="xsd:integer"/>
  <xsd:element name=“indirizzo” type=“tipoIndirizzo”/>
</xsd:complexType>

<xsd:element name=“persona” type=“tipoPersona”/>
```

XML Schema

- Cardinalità degli elementi
  - ⇨sono dichiarate attraverso gli attributi minOccurs e maxOccurs
- Valori possibili
  - ⇨`maxOccurs="1"` (default)
  - ⇨`maxOccurs="unbounded"`
  - ⇨`maxOccurs="5"`
  - ⇨`minOccurs="1"` (default per gli elementi)
  - ⇨`minOccurs="0"` (default per gli attributi)
  - ⇨`minOccurs="5"`

Cardinalità

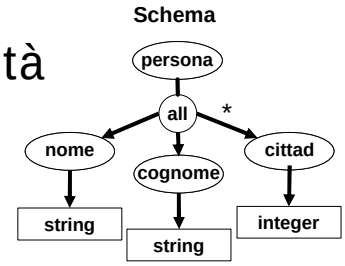
consente di definire sottoalberi ripetuti

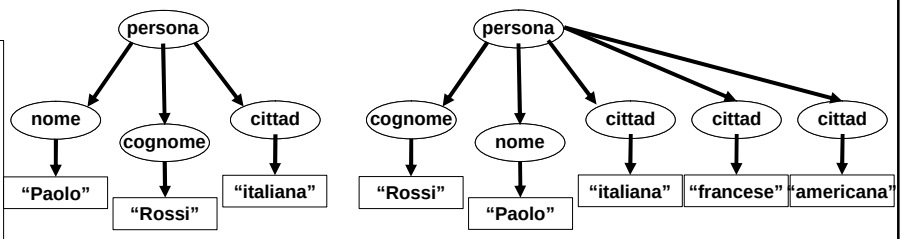
```
<xsd:complexType name="tipoPersona">
 <xsd:element name="nome"  type="xsd:string"/>
 <xsd:element name="cognome" type="xsd:string"/>
 <xsd:element name="cittadinanza"
                    type="xsd:string" maxOccurs="unbounded"/>
</xsd:complexType>

<xsd:element name="persona" type="tipoPersona"/>
```

```
<persona>
  <nome>Paolo</nome>
  <cognome>Rossi</cognome>
  <cittad>Italiana</cittad>
</persona>
```

```
<persona>
  <nome>Paolo</nome>
  <cognome>Rossi</cognome>
  <cittad>Italiana</cittad>
  <cittad>Francese</cittad>
  <cittad>American</cittad>
</persona>
```

Cardinalità





Cardinalità

consente di definire sottoalberi opzionali

```
<xsd:complexType name="tipoPersona">
  <xsd:element name="nome"  type="xsd:string"/>
  <xsd:element name="cognome" type="xsd:string"/>
  <xsd:element name="moglie"  type="tipoPersona"
                              minOccurs="0" maxOccurs="unbounded"/>
</xsd:complexType>

<xsd:element name="persona" type="tipoPersona"/>
```

```
<persona>
  <nome>Paolo</nome>
  <cognome>Rossi</cognome>
  <moglie>...</moglie>
  <moglie>...</moglie>
</person>
```

```
<persona>
  <name>Paolo</name>
  <cognome>Rossi</cognome>
</persona>
```

XML Schema

- Modelli di contenuto ammessi
  - ⇨elementOnly (che è quello standard)
  - ⇨mixed (con una sintassi ed una semantica molto più precisa)
  - ⇨empty
  - ⇨anyType (qualsiasi contenuto)

XML Schema

```
<formulaMatrimoniale>
Oggi, <data>19/07/2000</data>, <sposo>Clark Kent</sposo> e
<sposa>Lois Lane</sposa> si sono uniti in matrimonio...
</weddingFormula>
```

DTD:

```
<!ELEMENT formulaMatrimoniale (#PCDATA | data | sposo | sposa )+ >
```

```
<xsd:element name="formulaMatrimoniale">
   <xsd:complexType mixed="true">
      <xsd:element name="data"  type="xsd:date" minoccurs="1"/>
      <xsd:element name="sposo" type="xsd:string" minoccurs="1"/>
      <xsd:element name="sposa" type="xsd:string" minoccurs="1"/>
   </xsd:complexType>
</xsd:element>
```

XML Schema

- Quello che abbiamo visto
  - ⇨sono solo le funzionalità di base di XMLSchema
- Funzionalità avanzate
  - ⇨meccanismi di estensione/riuso di schemi >> limitate funzionalità orientate agli oggetti
  - ⇨vincoli di integrità (di chiave e di riferimento) >> basati su XPath

XML Schema

- Differenze principali rispetto ai DTD
  - ⇨è molto più orientato alle applicazioni centrate sui dati e meno ai documenti
  - ⇨consente di descrivere con dettaglio molto maggiore la struttura dei documenti (vincoli maggiori sui documenti validi – meno errori)
  - ⇨consente di gestire più efficacemente la composizione ed il riuso degli schemi

XML Schema

- Come associare lo schema al documento
  - ⇨non si utilizza l'elemento DOCTYPE
  - ⇨ma una sintassi alternativa

```
<?xml version="1.0"?>
<questionario id="geo"
      disciplina="Geografia“
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="E:\varie\questionario.xsd">
 <listaArgomenti>
  ...
</questionario>
```


XML Schema

- Esempi di schemi XML
  - ⇨questionario.xsd
  - ⇨libro.xsd (>> contenuto misto)
  - ⇨univ.xsd (>> tipi di dato)
- Albero prototipo associato
  - ⇨molto simile al DTD
  - ⇨le foglie non sono solo #PCDATA
  - ⇨i nodi speciali sono diversi

Riassumendo

- XML Schema

Termini della Licenza