Università Ca' Foscari di Venezia

Linguistica Informatica Mod. 1

Anno Accademico 2010 - 2011

XML

Rocco Tripodi
rocco@unive.it

Linguaggi dichiarativi: XML

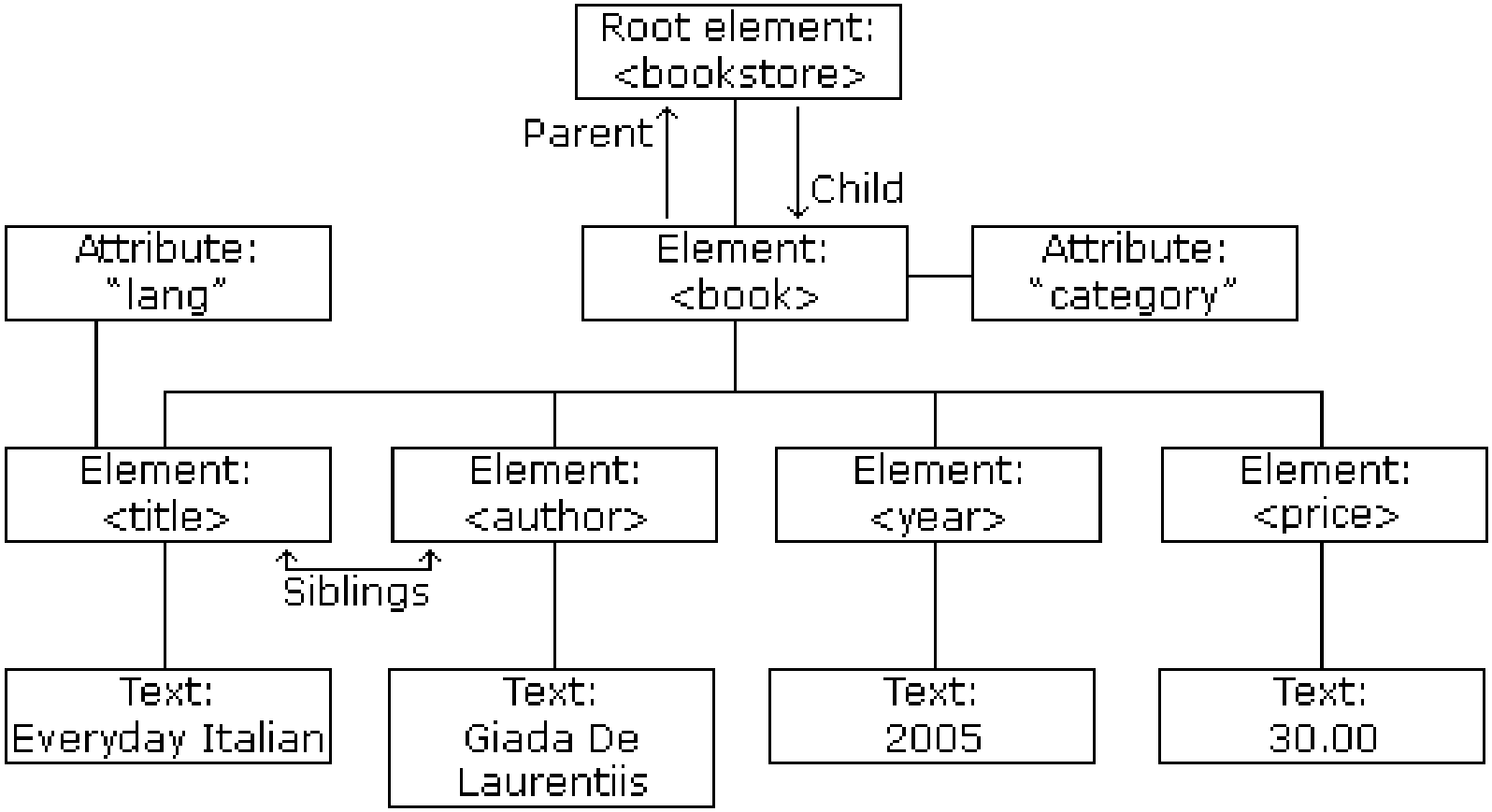
eXtensible Markup Language

meta linguaggio che consente di creare altri linguaggi di marcatura
Il progetto XML ha avuto inizio alla fine del 1996 e nel 1998 le specifiche sono divenute una raccomandazione ufficiale (Link)

È stato creato per strutturare, immagazzinare e scambiare informazioni

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nota SYSTEM “nota.dtd">
<nota>                                                    Nodo radice, apertura
   <da>Rocco</da>
   <a>Studenti</a>
   <titolo>Esempio</titolo>
   <testo>Questa è una nota in formato XML</testo>
</nota>                                                   Nodo radice, chiusura
```

XML: albero gerarchico



XML: documenti ben formati

Ogni documento deve avere un elemento radice (regola 1)

Ogni tag deve avere il rispettivo tag di chiusura (regola 2)

XML è *case sensitive* (regola 3)

Ogni elemento deve essere annidato correttamente (regola 4)

XML non definisce i tag che devono essere usati

I nomi dei tag:

- possono contenere lettere, numeri e altri caratteri
- non possono cominciare con un numero o un elemento di punteggiatura
- non possono cominciare con le lettere xml
- non possono contenere spazi

Attributi

- Inserimento di informazioni riguardo ai dati
  <book category="CHILDREN">
- Il valore dell'attributo deve essere inserito tra apici (regola 5)

XML: punti di forza

Codifica di tipo descrittivo anziché procedurale
- Separa il contenuto dalla presentazione (CSS, XSLT)

Indipendenza hardware e software
- Stessa codifica dei caratteri: Unicode

Concetto di tipo di documento
- Si può ricondurre più documenti a un unico tipo astratto
- Elaborazione automatica mediante *parser*
- DTD e XML Schema
- Validazione

<poesia>
  <titolo> Titolo </titolo>
  <verso> Verso </verso>
  <verso> Verso </verso>
  …
</poesia

Questo modello riuscirà a codificare solo una parte di dati. Tralascia i dati dell'autore, della metrica, dell'edizione, ecc.

XML: DTD

Oltre ad essere conforme alle regole di buona formazione un documento XML deve essere valido in base alle regole definite nella DTD (estensione .dtd)

Document Type Definition

Definizione delle struttura e degli elementi di un documento
Può essere contenuta in un file esterno al file XML che la usa o inclusa in esso stesso e contiene:
Gli elementi in cui si articola il documento
Gli attributi associati a ciascun elemento
Le entità richiamabili attraverso riferimenti all'interno del testo
Può essere restrittiva o permissiva

Tutte le dichiarazioni sono inserite tra parentesi angolari. L'apertura è seguita da un punto esclamativo al quale segue:
ELEMENT: per la dichiarazione di elementi
ATTLIST: per la dichiarazione di attributi
ENTITY: per le entità

DTD: modello di contentuo (CM)

<!ELEMENT tag (modello di contenuto)>

Il tag è un identificatore generico dell'elemento

CM indica l'articolazione interna dell'elemento o il tipo di dato che contiene

```
<!DOCTYPE nota
    [
    <!ELEMENT nota (to, da, titolo, testo)>
    <!ELEMENT a (#PCDATA)>
    <!ELEMENT da (#PCDATA)>
    <!ELEMENT titolo (#PCDATA)>
    <!ELEMENT testo (#PCDATA)>
    ]>
```

#PCDATA = qualsiasi sequenza di caratteri

L'articolazione interna viene indicata mediante l'elenco degli elementi figli, per i quali è possibile specificare la modalità di occorrenza

DTD: modalità di occorrenza

Gli elementi del modello del contenuto possono essere separati da un connettore

Virgola per indicare l'ordine in cui gli elementi devono comparire

Barra verticale per indicare l'opzionalità tra due elementi

Ogni elemento può essere seguito da un indicatore che specifica il numero di volte che l'elemento ricorre

?: occorrenza opzionale (zero o una volta)

+: occorrenza obbligatoria e ripetibile (>=1)

*: occorrenza opzionale e ripetibile (>=0)

Elementi misti: contenenti sia #PCDATA che sottoelementi

Elementi vuoti: indicati con #EMPTY

Elementi senza restrizioni: indicati con #ANY

DTD: attributi

<!ATTLIST tag_elemento nome_attributo tipo_valore modificatore>

Tipo di valore

CDATA: il valore dell'attributo è di tipo carattere
NMTOKEN(S): una o più stringhe di caratteri alfanumerici contenente soli i caratteri consentiti ad un identificatore generico
ID: sequenza di caratteri che definiscono l'identificatore univoco del tag
IDREF(S): punta al valore ID di un altro elemento (relazioni)
ENTITY: il valore è una entità definita della stessa DTD

Modificatori

#REQUIRED: attributo per il quale deve essere specificato un valore (ID)
#IMPLIED: opzionale
#FIXED: valore fisso

XML: riferimenti e entità

Riferimenti a carattere

Sono usate per inserire nel testo caratteri Unicode direttamente attraverso il rispettivo riferimento numerico (o alfaumerico)

& + **codice**

Entità

Sono sequenze arbitrarie di byte associate a nomi mnemonici

Le entità interne sono dichiarate nella DTD

Le entità esterne sono ricavabili da una sorgente esterna

Entità predefinite

Usate per indicare i caratteri di marcatura nel testo

< è uguale a <

> è uguale a >

& è uguale a

' è uguale a '

" è uguale a "

DTD: entità

Dichiarazione delle entità

<!ENTITY nome_entità "valore_entità">

Il nome dell'entità viene usato all'interno del testo per richiamare il suo valore &nome_entità;

Entità parametriche

Si usano quando si deve assegnare lo stesso gruppo di attributi a più elementi, in modo da rendere modulare la stesura della DTD

<!ENTITY % nome_entità_param "valore_entità_param">

Così da rendere più agevole la dichiarazione degli elementi nella DTD

<!ATTLIST elemento %nome_entità_parametrica;>

XML: struttura e validazione

Prologo

<?xml version="1.0" encoding="utf-8"?> (dichiarazione XML)

<!DOCTYPE elem_radice SYSTEM “nota.dtd"> (dichiarazione DTD esterna)

<!DOCTYPE elem_radice […dtd…]> (dichiarazione DTD interna)

XML valido: conforme alla DTD indicata (parser XML)

Commenti

inseriti tra i caratteri <-- e -->

XML DOM

Document Object Model

Raccomandazione W3C

Linguaggio che consente a programmi e script di accedere dinamicamente a un documento XML

Definisce gli oggetti e le proprietà di tutti gli elementi XML, inoltre definisce i metodi per accedere ad essi (funge da interfaccia)

È uno standard per recuperare, cambiare, aggiungere o eliminare elementi XML

Contiene funzioni per accedere e navigare la struttura ad albero di un documento XML

Un parser XML ricava la struttura del documento leggendolo e interpretandolo. Lo converte in un oggetto XML DOM al quale si può accedere tramite un linguaggio di scripting

XML DOM: funzioni

x.nodeName – il nome di x
x.nodeValue – il valore di x
x.parentNode – il nodo da cui discende x
x.childNodes – il nodo figlio di x
x.attributes – gli attributi di x

x.getElementsByTagName(name) recupera gli elementi con TagName = name
x.appendChild(node) – aggiunge un nodo figlio a x
x.removeChild(node) – elimina un nodo figlio a x

xmlDoc – l'oggetto XML DOM creato dal parser
getElementsByTagName("title")[0] – il primo elemento title
childNodes[0] – il primo elemento figlio

XML Schema 1

È un documento XML all'interno del quale vengono inserite le regole di un tipo di documento

definisce gli elementi che possono apparire in un documento
definisce gli attributi che possono apparire in un elemento
definisce gli elementi figli e il loro numero
definisce se un elemento è vuoto o può contenere testo
definisce il tipo degli attributi
definisce valori di default per elementi ed attributi

Può essere usato per validare i documenti XML
Può essere usato assieme a DTD
Non prevede la definizione delle entità
Supporta i *data types* e i *namespaces*

XML Schema 2

Il tag per la definizione degli elementi è <xs:element>

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

È equivalente a

<!ELEMENT xxx (yyy)>

Riferimento all'interno del file XML

<nota xmlns="nostro-indirizzo"

xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="nostro-indirizzo nota.xsd">

Elemento radice

<?xml version="1.0"?> (prologo XML)

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

targetNamespace="nostro-indirizzo" **dichiarazione del ns dello schema**

xmlns="nostro-indirizzo" **dichiarazione del ns di default**

Namespace

È un puntatore che indica dove sono definiti gli elementi che vengono usati in un documento

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

Indica che gli elementi e i *data types* usati nello Schema sono definiti all'interno del documento puntato

targetNamespace="nostro-indirizzo"

Viene definito un *namespace* per gli elementi che definiamo nello Schema

Oltre al *namespace* di default possono essere inseriti altri *namespace* per usare elementi definiti in altri schemi o DTD

<xmlns:gram=http://grammatica.it> Dichiarazione
<gram:avv>Sempre</gram:avv> Uso

Esempio

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

    <xs:element name="note">
     <xs:complexType>
      <xs:sequence>
       <xs:element name=“a" type="xs:string"/>
       <xs:element name=“da" type="xs:string"/>
       <xs:element name=“titolo" type="xs:string"/>
       <xs:element name=“testo" type="xs:string"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>

    </xs:schema>
```

XML Schema: data types

xs:string	Stringa di caratteri
xs:integer	Numero intero
xs:decimal	Numero decimale
xs:boolean	Valore booleano
xs:time	Ora
xs:uriReference	URI

<xs:element name="quantita" type="xs:integer" />

<quantita>123</quantita>

<quantita>uno</quantita> sbagliato

XML Schema: valori

Valore di default

è assegnato quando non viene specificato un altro valore

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

Valore fisso

è assegnato automaticamente e non può essere modificato

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

Restrizioni

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

XML Schema: attributi

Tutti gli attributi sono dichiari come elementi semplici

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

Valgono le stesse regole viste per gli elementi riguardo ai data types e ai valori

Attributi opzionali e obbligatori

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

XML Schema: mixed 1

Elementi complessi con contenuto misto

XML

```
<letter>
  Caro Sign.<name>Mario Rossi</name>.
  Il suo ordine <orderid>1032</orderid>
  sarà spedito il <shipdate>2001-07-13</shipdate>.
</letter>
```

Schema

```
<xs:element name="letter">
  <xs:complexType mixed="true"> Consente l'uso di testo libero tra i nodi figli
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="orderid" type="xs:positiveInteger"/>
      <xs:element name="shipdate" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XML Schema: ordine o occorrenze

xs:all - gli elementi devono ricorrere almeno una volta. Non imposta l'ordine

```
<xs:all>
elementi ..
</xs:all>
```

xs:choice - condizionale, o uno o l'altro elemento

xs:sequence - indica l'ordine di apparizione degli elementi

min e maxOccurs - indica il numero di occorrenze

```
<xs:element name="paragraph" type="xs:string"
  maxOccurs="10" minOccurs="0"/>
```

group - raggruppa una serie di elementi

es: person che ha un gruppo di elementi figli indicanti nome, cognome, ecc

```
<xs:group name="persongroup"> elementi </xs:group>
```