

XML eXtensible Markup Language

Schemi – parte a DTD

versione 3.0

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

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



XML: Schemi >> Sommario



Sommario

- Schemi
- DTD
- Validatori
- Linguaggi, Metalinguaggi, Documenti
- Dettagli sulla Sintassi
 - ⇒ Entita



Schemi

- Schema per un Documento XML
 - ⇒ “albero prototipo”
 - ⇒ descrive quali strutture sono ammissibili per i sottoalberi e quali non lo sono
 - ⇒ descrive alcuni vincoli di integrità sui dati
- Documento valido rispetto a uno schema
 - ⇒ l'albero è conforme allo schema (ovvero: è un'istanza dello schema)



Schemi

- Due categorie principali di schemi
- DTD (“Document Type Declaration”)
 - ⇒ orientato ai documenti elettronici
 - ⇒ parte dello standard XML 1.0
- XML Schema
 - ⇒ standard successivo
 - ⇒ orientato alla descrizione dei dati

XML: Schemi >> Schemi

Schemi

- Ma...
 - ⇒ nessuno dei due viene considerato completamente soddisfacente
- Infatti, ci sono moltissime altre proposte
 - ⇒ Relax
 - ⇒ Relax-NG
 - ⇒ Trex
 - ⇒ ...

G. Mecca - XML 5

XML: Schemi >> DTD

DTD

- Albero sintattico di una “grammatica”
 - ⇒ basato su “espressioni regolari”
- Un solo tipo di valori: la stringa
 - ⇒ #PCDATA
- Rispetto ai tipi dei linguaggi di progr.
 - ⇒ l'enfasi non è sulla natura dei valori (tipi)
 - ⇒ ma sulla posizione reciproca e sulle occorrenze degli elementi

G. Mecca - XML 6

XML: Schemi >> DTD

DTD

- Struttura del DTD
 - ⇒ contiene vari tipi di nodi
- Nodo documento
- Nodi per gli elementi
 - ⇒ con la descrizione del contenuto
- Nodi per gli attributi
 - ⇒ con i relativi vincoli
- Nodi per i valori
 - ⇒ nodi #PCDATA

G. Mecca - XML 7

XML: Schemi >> DTD

DTD

- Nodi speciali
 - ⇒ nodi che descrivono quali sottoalberi sono corretti per un certo elemento
- Nodo sequenza
 - ⇒ seq oppure ,
- Nodo alternativa
 - ⇒ U oppure I

G. Mecca - XML 8



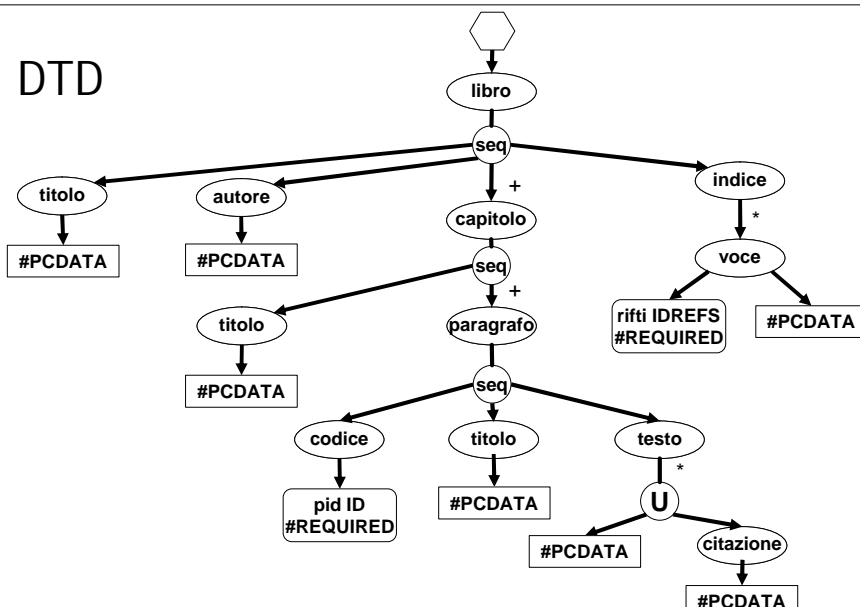
DTD

○ Indicatori di cardinalità

- ⇒ etichette degli archi dell'albero
- ⇒ indicano la numerosità degli elementi presenti in un sottoalbero
- ⇒ minimo e massimo numero di occorrenze del sottoalbero
- ⇒ una o più volte – 1..* oppure +
- ⇒ zero o una volta – 0..1 oppure ?
- ⇒ zero o più volte – 0..* oppure *



DTD





DTD

○ Sintassi concreta del DTD

- ⇒ il DTD è a sua volta un documento di testo
- ⇒ tipicamente con estensione .dtd
- ⇒ ma non è un documento XML (sintassi diversa)

○ Elementi sintattici principali

- ⇒ dichiarazione di elemento `<!ELEMENT`
- ⇒ dichiarazione di attributi `<!ATTLIST`



DTD

```
<!ELEMENT libro      (titolo, autore, capitolo+, indice)>
<!ELEMENT titolo     (#PCDATA)>
<!ELEMENT autore     (#PCDATA)>
<!ELEMENT capitolo   (titolo, paragrafo+)>
<!ELEMENT paragrafo (codice, titolo, testo)>
<!ELEMENT testo      (#PCDATA | citazione)*>
<!ELEMENT citazione  (#PCDATA)>
<!ELEMENT codice     EMPTY>
  <!ATTLIST codice    pid ID #REQUIRED>
<!ELEMENT indice     (voce*)>
<!ELEMENT voce       (#PCDATA)>
  <!ATTLIST voce      rifti IDREFS #REQUIRED>
```

vincolo: una sola definizione per ogni elemento



Validità

>> libro.dtd, libro.xml

- Nel documento

- ⇒ il DTD può essere dichiarato con la sintassi `<!DOCTYPE`

- ⇒ es: `<!DOCTYPE libro SYSTEM "libro.dtd">`

- In questo caso

- ⇒ il documento si impegna ad essere valido rispetto al DTD associato



Validità

- Validità rispetto ad un DTD

- ⇒ l'InfoSet del documento deve rispettare i vincoli imposti dal DTD

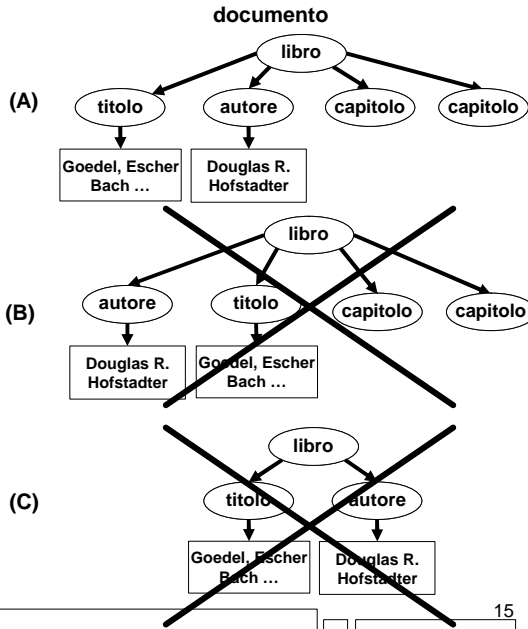
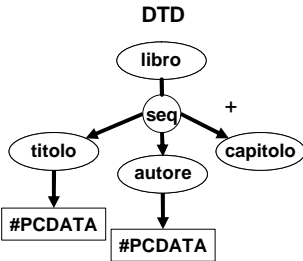
- Attenzione alla differenza

- ⇒ documento senza dichiarazione di DTD: deve essere ben formato

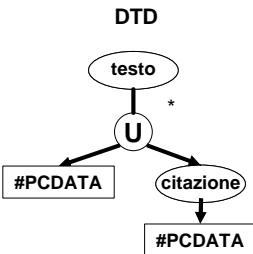
- ⇒ documento con dichiarazione di DTD: deve essere ben formato e valido rispetto al DTD



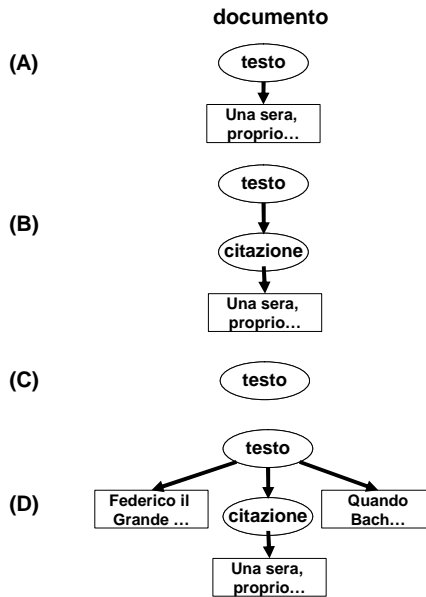
Validità



Validità



struttura tipica
per gli elementi di
contenuto misto
>> libro.xml, libro.dtd





DTD

- Vantaggi dei DTD
 - ⇒ sono la tecnologia più diffusa per associare schemi ad un documento XML
- Svantaggi dei DTD
 - ⇒ esiste un unico tipo di dati: #PCDATA
 - ⇒ definizione imprecisa di contenuto misto
 - ⇒ scarsa modularità: non è possibile riutilizzare o estendere le strutture già definite
- La soluzione: XML Schema



Validatori

- Esistono vari validatori
 - ⇒ un elenco completo su <http://w3.org/XML>,
<http://www.w3.org/XML/Schema>
- Browser
 - ⇒ tutti i principali browser consentono di visualizzare documenti XML e verificano che siano ben formati
 - ⇒ normalmente non effettuano la convalida rispetto al DTD

Validatori

- Due strumenti interessanti
 - ⇒ xmlvalid – Elcel Technologies
<http://www.elcel.com/products/xmlvalid.html>
 - ⇒ utilizzabile solo dalla linea di comando
 - ⇒ XMLSpy 2005 – Home Edition
 - ⇒ strumento con interfaccia grafica sofisticata
 - ⇒ entrambi distribuiti gratuitamente

Validatori

- XMLValid
 - ⇒ molto semplice da usare
 - ⇒ verifica solo la validità rispetto ad un DTD
- Sintassi
 - ⇒ `xmlvalid <file.xml>`

>> univ.xml, univ.dtd
>> xmlvalid univ.xml



Validatori

○ XMLSpy

- ⇒ strumento grafico che supporta sia DTD che XMLSchema (>>)
- ⇒ sia molte altre tecnologie collegate (es: XSL, XQuery ecc.)
- ⇒ la versione gratuita ha funzionalità ridotte (es: manca dell'editor di espressioni XPath)
- ⇒ ne esistono versioni commerciali più evolute

>> univ-s.xml, univ.xsd



Linguaggi, Metalinguaggi, Documenti

○ L'utilizzo tipico degli schemi

- ⇒ definire "linguaggi"

○ Linguaggio

- ⇒ collezione di stringhe che obbediscono ad una grammatica comune
- ⇒ normalmente utilizzati in un contesto applicativo particolare
- ⇒ es: linguaggio di programmazione



Linguaggi, Metalinguaggi, Documenti

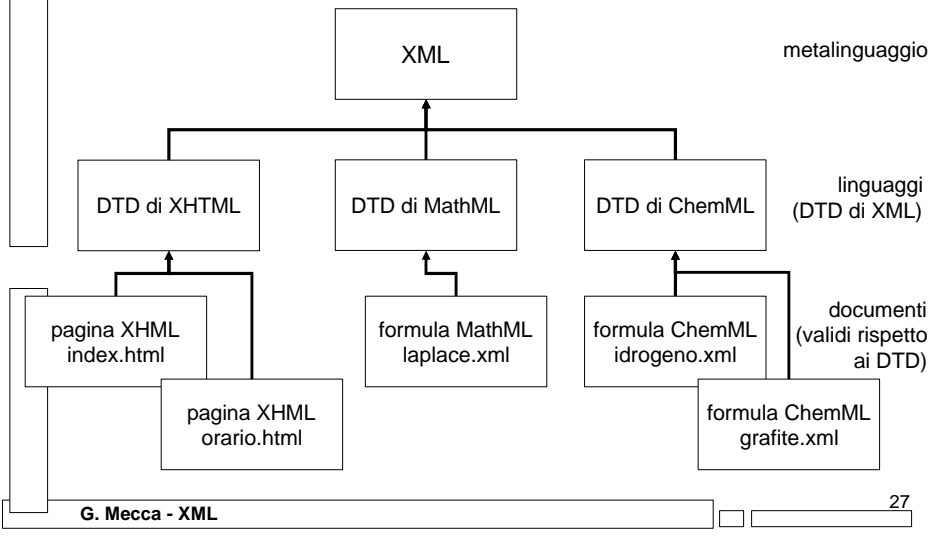
- Nella terminologia di XML
 - ⇒ stringa = documento
 - ⇒ linguaggio = collezione di documenti
 - ⇒ grammatica = schema (tipicamente DTD)
 - ⇒ lo schema definisce la grammatica che i documenti del linguaggio devono rispettare
- In quest'ottica
 - ⇒ XML è un "metalinguaggio", ovvero un linguaggio per definire altri linguaggi



Linguaggi, Metalinguaggi, Documenti

- Esempi di linguaggi basati su XML
 - ⇒ XHTML: linguaggio per la composizione di pagine Web
 - ⇒ WML: linguaggio per la composizione di pagine Wap
 - ⇒ MathML: linguaggio per la scrittura di espressioni matematiche
 - ⇒ ChemML: linguaggio per la descrizione di formule e strutture chimiche

Linguaggi, Metalinguaggi, Documenti



Linguaggi, Metalinguaggi, Documenti

○ Definire schemi per XML

- ⇒ corrisponde a definire i requisiti che i documenti XML utilizzati in un contesto applicativo specifico devono rispettare
- ⇒ la parola linguaggio va interpretata in quest'ottica
- ⇒ esempio di linguaggio: il DTD che stabilisce come due sistemi informativi distribuiti devono scambiarsi i dati



Dettagli sulla Sintassi

- Schema per il documento XML
 - ⇒ vincoli sulla struttura dell'albero
- Sintassi per i DTD
 - ⇒ non è basata su XML
- Attenzione
 - ⇒ la presenza dello schema è opzionale
 - ⇒ documenti validi e documenti ben formati



Dettagli sulla Sintassi

- Riferimento al DTD
 - ⇒ nodo DOCTYPE
- DTD Esterno
 - ⇒ il DTD è contenuto in un file separato
 - ⇒ viene specificato l'URI del DTD (relativo o assoluto)
 - ⇒ es: `<!DOCTYPE questionario SYSTEM "questionario.dtd">`



Dettagli sulla Sintassi

○ DTD interno

⇒ il DTD può essere contenuto nel file XML

⇒ `<?xml version="1.0">`

```
<!DOCTYPE archiviodomande [  
  <!ELEMENT archiviodomande  
    (elencoArgomenti,elencoQuesiti)>
```

```
  <!ATTLIST archiviodomande  
    disciplina CDATA #REQUIRED
```

```
  ... ]>
```

```
<archiviodomande>...</archiviodomande>
```



Dettagli sulla Sintassi

○ Identificatore del DTD

⇒ normalmente: identificatore di "sistema"
(SYSTEM)

○ Identificatori pubblici

⇒ alcuni DTD molto utilizzati hanno un "nome"
pubblico (PUBLIC); es: il DTD di XHTML

⇒ `<!DOCTYPE html
 PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`



Dettagli sulla Sintassi

- Definizione degli elementi

- ⇒ <!ELEMENT *nome* (*contenuto*)>

- ⇒ es: <!ELEMENT elencoQuesiti (quesito)* >

- Per ciascun elemento

- ⇒ è consentita una sola dichiarazione di contenuto



Dettagli sulla Sintassi

- Modello di contenuto (“content model”)

- ⇒ ELEMENT ONLY elementi con * + ?

- ⇒ TEXT (#PCDATA)

- ⇒ MIXED (#PCDATA | <el1> | <el2> | ...)*

- ⇒ EMPTY

- ⇒ ANY



Dettagli sulla Sintassi

○ Utilizzo di EMPTY

⇒ per quegli elementi che hanno attributi ma non hanno contenuto né sottoelementi

○ Esempio

<!ELEMENT soluzione EMPTY>

<soluzione lettera="b" />

<!ELEMENT difficolta EMPTY>

<difficolta valore="d0" />



Dettagli sulla Sintassi

○ Utilizzo di ANY

⇒ per quegli elementi di cui non si vuole vincolare il contenuto

○ Esempio

⇒ <!ELEMENT risposta ANY>

⇒ nelle risposte proposte per una domanda è opportuno poter specificare elementi diversi; es: immagini, o testo formattato



Dettagli sulla Sintassi

- Definizione degli attributi

⇒ `<!ATTLIST nomeEl listaDefAttributi>`

- Definizione dell'attributo

⇒ `nomeAttr tipo vincolo`

- Esempio

`<!ATTLIST soluzione
lettera CDATA #REQUIRED>`



Dettagli sulla Sintassi

- Tipo dell'attributo

⇒ CDATA: testo

⇒ ID: identificatore; non ammette duplicati

⇒ IDREF: riferimento ad un ID (integr. ref.)

⇒ IDREFS: riferimenti ad una lista di ID
(integrità referenziale)

⇒ tipo enumerato: (val1 | val2 | ... | valn)

es: centesimi ("1" | "2" | "5" | "10" | "20" | "50")

⇒ NMTOKEN, NMTOKENS



Dettagli sulla Sintassi

○ Vincoli sull'attributo

⇒ #REQUIRED obbligatorio

es: <!ATTLIST soluzione lettera CDATA #REQUIRED>

⇒ #IMPLIED opzionale

es: <!ATTLIST elencoQuesiti disciplina CDATA #IMPLIED>

⇒ *valore* opzionale con v. standard

es: <!ATTLIST difficolta valore CDATA "1">

⇒ #FIXED *valore* valore fissato

es: <!ATTLIST difficolta valore "1" #FIXED>



Dettagli sulla Sintassi

○ Contenuto del DTD

⇒ dichiarazione di elementi

⇒ dichiarazioni di attributo

⇒ dichiarazione di entità

○ Entità

⇒ le vediamo nel seguito



Entità

○ Entità

- ⇒ costante simbolica corrispondente ad un pezzo di documento con un nome
- ⇒ viene definita nel DTD

○ Due categorie

- ⇒ entità generali (“general entity”) – dichiarate nel DTD e usate nel documento
- ⇒ entità parametriche (“parameter entity”) – dichiarata e usata nel DTD



Entità

○ General Entity

- ⇒ definite nel DTD, utilizzate nel documento

○ Esempio

- ⇒ `<!ENTITY pigreco “3.1412”>`
- ⇒ utilizzo della costante: `&<nome>`;
es: `&pigreco`;
- ⇒ possono essere definite in file esterni da includere
`<!ENTITY capitolo1 SYSTEM “cap1.xml”>`



Entità

○ Attenzione

⇒ alcuni caratteri sono riservati e non possono essere usati nel testo

○ Esempi: <, >, &

○ In questo caso

⇒ due possibili soluzioni

⇒ soluzione 1: utilizzare le entità predefinite

⇒ soluzione 2: utilizzare sezioni di tipo CDATA



Entità

○ Entità predefinite

⇒ corrispondono ai caratteri riservati della sintassi

⇒ <

<

~~Esempio scorretto:~~

~~<risposta>x > y</risposta>~~

⇒ >

>

~~<risposta>(x > y) && (y > 1)</risposta>~~

⇒ &

&

Esempio corretto:

<risposta>x > y</risposta>

⇒ '

'

<risposta>(x > y) &&

⇒ "

"

(y > 1)</risposta>



Entità

○ Più in generale

- ⇒ esiste un'entità predefinita per ogni carattere del set di caratteri Unicode
- ⇒ `&#<n>`, dove `<n>` è il codice esadecimale del carattere es: `>` equivale a `>`;
- ⇒ di conseguenza lo stesso carattere speciale può essere inserito in modi diversi nel documento XML



Entità

○ Sezioni CDATA

- ⇒ sezioni del documento i cui caratteri non vengono sottoposti ad analisi sintattica
- ⇒ possono contenere testo di ogni tipo
- ⇒ delimitate tra `<![CDATA[` e `]]>`

○ Esempio

```
<risposta>  
<![CDATA[  
    (x > y) && (y > 1)  
]]>  
</risposta>
```



Entita

○ Parameter Entity

- ⇒ entità che non vengono utilizzate nel documento ma nello stesso DTD
- ⇒ a differenza delle entità generali
- ⇒ si tratta sostanzialmente di pezzi di DTD da riutilizzare varie volte
- ⇒ tipicamente definizioni di elementi o sottoalberi frequenti (es: persona)



Entita

○ Utilizzo tipico

- ⇒ definisco la struttura di un albero in un DTD
- ⇒ riutilizzo la struttura in un altro DTD senza doverla ridefinire

○ Esempio: collezioneQuestionari.dtd:

- ⇒ collezione di questionari
- ⇒ `<!ENTITY % questionario SYSTEM "questionario.dtd">`
- ⇒ nel DTD: `%questionario;`



Riassumendo

- Schemi
- DTD
- Validatori
- Linguaggi, Metalinguaggi, Documenti
- Dettagli sulla Sintassi
- Entità



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.