XML
eXtensible Markup Language

Alberi e Documenti

versione 3.0

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

Sommario

- Modello Logico
  - ⇨Alberi XML (“InfoSet”)
- Sintassi XML
- Altri Esempi
- Nell’Applicazione
- Dettagli sulla Sintassi
- Namespace

Modello Logico

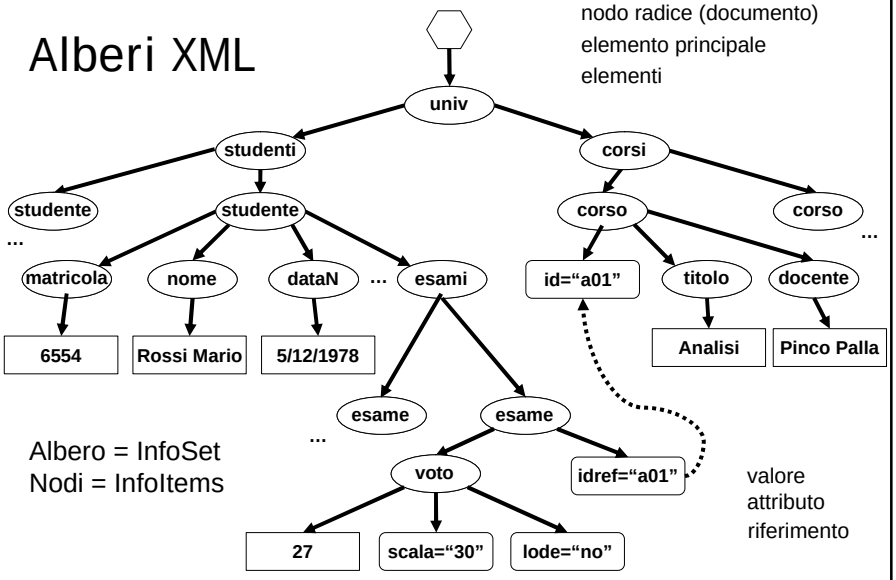
ATTENZIONE
alle particolarità
degli alberi XML

- Una sintassi per costruire documenti
  - ⇨basata su marcatori (“tag”)
- Modello logico sottostante
  - ⇨alberi di tipo particolare
  - ⇨contengono vari tipi di nodi (“nodi tipizzati”)
  - ⇨ciascun nodo può avere un identificatore unico che lo contraddistingue nell’albero
  - ⇨è possibile specificare riferimenti incrociati tra i nodi

Alberi XML

- Tipologie di nodi
  - ⇨nodo radice (o “nodo documento”)
  - ⇨nodi elementi
  - ⇨nodi attributo
  - ⇨nodi valore
- Un esempio
  - ⇨un albero XML che descrive una collezione di dati universitari (studtenti, corsi, esami)

Alberi XML



Alberi XML

- Riassumendo
  - ⇨un unico nodo radice (nodo documento)
  - ⇨un elemento principale (unico figlio del nodo radice)
  - ⇨i nodi interni sono tutti nodi elemento
  - ⇨i nodi foglia sono nodi valore oppure nodi attributo

Alberi XML

- Funzione degli attributi
  - ⇨simili ad una coppia elemento+valore
- Utilizzo degli attributi
  - ⇨gli attributi servono per valori speciali es: identificatori e riferimenti
  - ⇨gli attributi servono per i metadati (“dati che descrivono i dati”); es: scala dei voti
  - ⇨gli attributi servono a specificare valori che devono essere meno “visibili” (es: stampa)

Alberi XML

- Differenza con i modelli a oggetti
  - ⇨gli elementi dell’albero sono assimilabili ad oggetti, classificabili rispetto al loro nome
  - ⇨ma vengono utilizzati meccanismi molto diversi di organizzazione degli oggetti
  - ⇨contenimento, vincolato alla struttura di albero (un elemento non può essere figlio di due padri)
  - ⇨riferimenti espliciti con identificatori gestiti dal programmatore e non dal sistema (OID) >> correttezza dei riferimenti
  - ⇨non esiste esplicitamente la nozione di ereditarietà

Sintassi XML

- Memorizzazione concreta degli alberi
  - ⇨attraverso “documenti”
  - ⇨normalmente contenuti in file di testo
  - ⇨con estensione .xml
- Sintassi del file di testo
  - ⇨sintassi concreta definita nello standard basata su marcatori (“tag”)
  - ⇨prevede una serie di regole precise

Sintassi XML

- Marcatore (“Tag”)
  - ⇨stringa corrispondente ad un nome di elemento racchiuso tra parentesi acute
- Ne esistono di tre tipi
  - ⇨tag di apertura: <*elemento*>
  - ⇨tag di apertura: </*elemento*>
  - ⇨tag di apertura e chiusura: <*elemento/*>

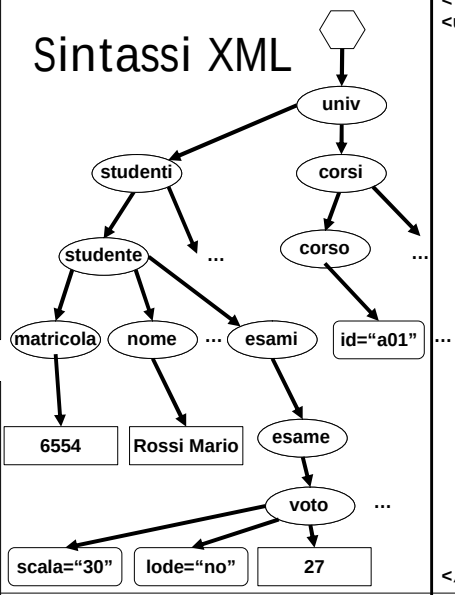
Sintassi XML

- Come ottenere il documento dall'InfoSet ?
  - ⇨è possibile definire un algoritmo preciso
  - ⇨consiste in una visita in preordine dell'albero; per ciascun elemento visitato vengono aggiunte stringhe al documento
  - ⇨per l'elemento radice viene aggiunto un prologo fisso: <? xml version="1.0" ?>

Sintassi XML

- Come ottenere il documento dall'InfoSet ?
  - ⇨per ciascun elemento visitato, viene prodotto il tag di apertura corrispondente; es: <corso>
  - ⇨vengono visitati ricorsivamente i sottoalberi
  - ⇨al termine della visita dei sottoalberi viene prodotto il tag di chiusura dell'elemento; es: </corso>
  - ⇨i valori sono riportati identicamente sotto forma di stringhe
  - ⇨gli attributi sono riportati sotto forma di coppie nome=valore all'interno dei tag di apertura dei corrispondenti elementi

Sintassi XML




```
<?xml version="1.0" encoding="UTF-8"?>
<univ>
  <studenti>
    <studente>
      <matricola>
        6554
      </matricola>
      <nome>Rossi Mario</nome>
      ...
      <esami>
        <esame>
          <voto scala="30"
                lode="no">
            27
          </voto>
          ...
        </esame>
      </esami>
    </studente>
    ...
  </studenti>
  <corsi>
    <corso id="a01">
      ...
    </corso>
  </corsi>
</univ>
```

file di testo
es: univ.xml

Sintassi XML

- In sostanza
  - ⇨alla nidificazione dei nodi nell'Infoset (contenimento) corrisponde la nidificazione (contenimento) dei tag nel documento
  - ⇨tutti i valori sono codificati come caratteri
- Regola sintattica
  - ⇨i documenti XML devono essere "ben formati"
  - ⇨si tratta di una serie di caratteristiche sintattiche che conseguono dall'algoritmo di codifica visto

Sintassi XML

- Documento Ben Formato (“well formed”)
  - ⇨deve contenere un prologo <? xml ... ?>
  - ⇨per ogni tag aperto deve esserci un tag chiuso corrispondente
  - ⇨i tag devono essere nidificati correttamente
    es: <a>...
    e non <a>...
  - ⇨gli attributi devono essere codificati correttamente tra virgolette

Sintassi XML

- Difetto della sintassi
  - ⇨eccessiva verbosità
  - ⇨i tag di chiusura sono espliciti mentre potrebbero essere omessi
- Esempio: il linguaggio delle parentesi
  - ⇨le parentesi chiuse sono implicitamente associate alle parentesi aperte
    $(_a\ 3+4\ *\ (_b\ 2+1\ )_b\ +\ 4\ )_a$ >> $(_a\ 3+4*(_b\ 2+1)\ +\ 4)$

Sintassi XML

- Pregio della sintassi
  - ⇨semplice da interpretare (per umani e non)
- Il documento è auto-descrittivo
  - ⇨il significato (“semantica”) dei valori è codificato nel nome degli elementi e degli attributi
- L’analisi sintattica (“parsing”) è semplice
  - ⇨dal documento all’InfoSet (algoritmo inverso rispetto a quello visto)

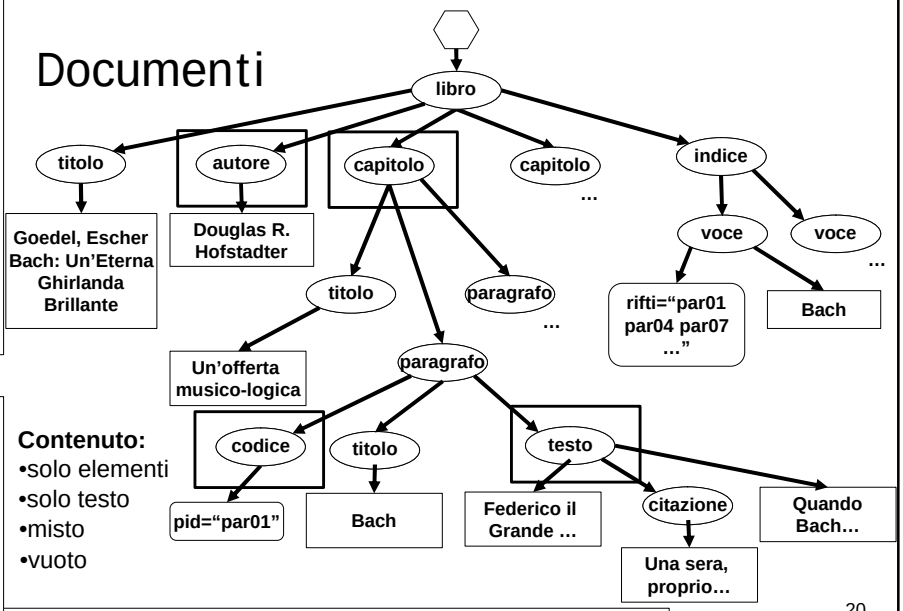
Altri Esempi

- Motivi di successo
  - ⇨gli alberi sono strutture flessibili
  - ⇨la sintassi è standard e relativam. semplice
- E’ applicato in moltissimi settori
  - ⇨siti Web
  - ⇨basi di dati
  - ⇨gestione di documenti elettronici
  - ⇨cooperazione applicativa

Altri Esempi

- Esempio
  - ⇨gestione di documenti elettronici
  - ⇨sistemi informativi in cui vengono memorizzati in formato elettronico documenti di vario genere
  - ⇨es: libri di una biblioteca
  - ⇨es: documenti ufficiali di una pubblica amministrazione (sistema di protocollo)

Documenti



Contenuto:
- solo elementi
- solo testo
- misto
- vuoto

Altri Esempi

- Contenuto di un elemento
  - ⇨varie possibilità, a seconda di come sono fatti i sottoalberi (trascurando gli attributi)
- Solo elementi (“element only”)
  - ⇨es: <capitolo id=“c01”>
    <titolo> ... </titolo>
    <paragrafo> ... </paragrafo>
    </capitolo>
- Solo valori (“text”)
  - ⇨es: <autore>D. Hofstadter</autore>

Altri Esempi

- Misto (“mixed”)
  - ⇨elementi e valori
  - ⇨es: <testo>
    Federico il Grande …
    <citazione>Una sera …</citazione>
    Quando Bach...
    </testo>
- Vuoto (“empty”)
  - ⇨solo attributi; viene codificato con tag di apertura e chiusura
  - ⇨es: <codice id=“par01” /> oppure ,

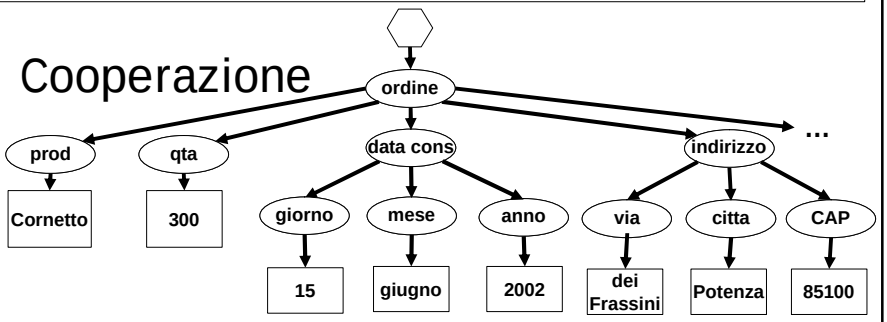
Altri Esempi

- Un altro esempio: cooperaz. applicativa
  - ⇨scenario in cui due sistemi informativi indipendenti devono scambiarsi dati e servizi attraverso la rete
  - ⇨es: il sistema informativo di un supermercato COOP ordina gelati al sistema informativo dell'Algida

Altri Esempi

- Se i due sistemi sono eterogenei
  - ⇨presumibilmente utilizzano strutture di dati e formati completamente diversi per la rappresentazione dei dati
  - ⇨tipicamente hanno basi di dati diverse ed indipendenti
- XML è un'ottima soluzione per lo scambio
  - ⇨fornisce un formato "neutro" e facilmente interpretabile dalle due applicazioni

Cooperazione




```
<?xml version="1.0" ?>
<ordine>
  <prod>Cornetto</prod>
  <qta>300</qta>
  <dataCons>
    <giorno>15</giorno>
    <mese>giugno</giugno>
    <anno>2002</anno>
  </dataCons>
  <indirizzo>
    <via>dei Frassini</via>
    <citta>Potenza</citta>
    <CAP>85100</CAP>
  </indirizzo>
  ...
</ordine>
```


Nell'Applicazione

- Questionari
  - ⇨documenti XML
- Principali elementi
  - ⇨listaArgomenti, argomento
  - ⇨listaDifficolta, difficolta
  - ⇨listaQuesiti, quesito
  - ⇨domanda
  - ⇨listaRisposte, risposte

```
<?xml version="1.0" ?>
<questionario id="geo" disciplina="Geografia">
   <listaArgomenti>
      <argomento id="cap" nome="Capitali"/>
      <argomento id="naz" nome="Nazioni Europee"/>
   </listaArgomenti>
   <listaDifficolta>
       <difficolta id="d0" descrizione="Difficolta' limitata"/>
       <difficolta id="d1" descrizione="Difficolta' media"/>
       <difficolta id="d2" descrizione="Difficolta' avanzata"/>
   </listaDifficolta>
   <listaQuesiti>
      <quesito id="cap01" argomento="cap" difficolta="d1">
          <domanda>Qual e' la capitale dell'Italia ?</domanda>
          <listaRisposte soluzione="b">
              <risposta> Parigi </risposta>
              <risposta> Roma </risposta>
              <risposta> Londra </risposta>
              <risposta> Potenza </risposta>
          </listaRisposte>
      </quesito>
      ...
   </listaQuesiti>
</questionario>
```


Nell'Applicazione

>> DOMInspector
>> questionario-test-q9.xml

- Come vedere l'InfoSet
  - ⇨è possibile utilizzare il DOMInspector di Mozilla
  - ⇨uno strumento che visualizza l'InfoSet di un documento XML consentendo di esplorarlo
- Attenzione
  - ⇨si scoprono alcuni nodi dell'albero non attesi
  - ⇨quelli corrispondenti agli spazi bianchi

Nell'Applicazione

- Di conseguenza
  - ⇨i frammenti seguenti corrispondono a sottoalberi diversi

```
<listaRisposte soluzione="a">
  <risposta>
    Parigi
  </risposta>
</listaRisposte>
```

```
<listaRisposte soluzione="a">
  <risposta>Parigi</risposta>
</listaRisposte>
```

```
<listaRisposte soluzione="a"><risposta>Parigi</risposta></listaRisposte>
```


Dettagli sulla Sintassi

- Regole generali
  - ⇨la sintassi è sensibile alle maiuscole
  - ⇨per convenzione i nomi degli elementi e degli attributi vengono specificati in minuscolo
- Commenti
  - ⇨stringhe delimitate da <!-- e -->
  - ⇨possono occupare più di una riga
  - ⇨es: <!-- quesito n. 2 -->

Dettagli sulla Sintassi

- Identificatori
  - ⇨nomi di elementi e attributi
  - ⇨sequenza di lettere, cifre, e underscore (_)
  - ⇨non possono contenere caratteri speciali
  - ⇨iniziano con una lettera oppure _ oppure :
- Esempi
  - ⇨questionario, risposta_1, :domanda

Dettagli sulla Sintassi

- Contenuto di un InfoSet XML
  - ⇨nodo documento
  - ⇨nodi elemento (tra cui l'elemento principale)
  - ⇨nodi attributo
  - ⇨nodi valore
  - ⇨commenti
  - ⇨istruzioni di elaborazione
  - ⇨namespace
  - ⇨nodo DOCTYPE (>>)

Dettagli sulla Sintassi

- Istruzioni di Elaborazione
  - ⇨“processing instructions”
  - ⇨servono a dare indicazioni alle applicazioni che manipolano il documento
  - ⇨delimitate da <? e ?>
- Esempi
  - ⇨prologo <? xml version=“1.0 ? encoding=“”>
  - ⇨dichiarazione del foglio di stile
    <?xml-stylesheet type="text/xsl" href="PdD-stile.xsl"?>

Namespace

- In un documento XML
  - ⇨è possibile che ci siano elementi che hanno lo stesso nome ma denotano concetti diversi
  - ⇨es: in un’applicazione della pubblica ammin. che offre servizi di rete
    protocollo (tecnologia o burocrazia ?)
  - ⇨è necessario uno strumento per distinguere elementi con lo stesso nome ma semantiche diverse

Namespace

- Spazio di Nomi (“Namespace”)
  - ⇨standard (W3C) per qualificare i nomi
  - ⇨gruppo di elementi con un nome
  - ⇨nello standard W3C il nome deve essere un URI - normalmente si tratta di un URN
- Esempio: l’elemento protocollo
  - ⇨urn:w3.org:protocollo
  - ⇨urn:funzionepubblica:protocollo

Namespace

- Attenzione
  - ⇨spesso l’URI utilizzato sembra un URL
  - ⇨http://www.w3c.org:protocollo, oppure http://www.funzionepubblica.it:protocollo
  - ⇨ma in realtà si tratta comunque di un URN
  - ⇨ovvero non corrisponde ad una risorsa fisicamente disponibile sulla rete
  - ⇨ma solo ad un nome che rispetta la sintassi degli URL

Namespace

- Quindi
  - ⇨un namespace non è un dizionario né una collezione di stringhe corrispondenti a nomi
  - ⇨un namespace non corrisponde necessariamente ad un'autorità che ha controllo sui nomi descritti dal namespace
  - ⇨cercando di visualizzare l'URI attraverso il browser normalmente non viene trovato nulla
  - ⇨funzione analoga ai namespace di .NET

Namespace

- Per definire il namespace
  - ⇨attributo speciale "xmlns"
  - ⇨consente di introdurre un namespace (URN)
  - ⇨ed inoltre di associargli un prefisso compatto per fare riferimento al namespace nel resto del documento
  - ⇨è possibile associare ad uno stesso documento vari namespace, ognuno con il proprio prefisso

Namespace

- Sintassi per i namespace
  - ⇨l'attributo xmlns può essere associato ad un qualsiasi elemento n (tipicamente la radice)
  - ⇨es: <documento xmlns:w3c="http://www.w3.org" xmlns:fp="http://www.funzionepubblica.it">
  - ⇨dal quel momento il namespace è visibile in tutto il sottoalbero la cui radice è n
  - ⇨nel sottoalbero i nomi degli elementi possono essere distinti attraverso il prefisso es: <w3c:protocollo>, <fp:protocollo>

Riassumendo

- Modello Logico
  - ⇨Alberi XML ("InfoSet")
- Sintassi XML
- Altri Esempi
- Nell'Applicazione
- Dettagli sulla Sintassi
- Namespace

Termini della Licenza