

informatica di base per le discipline umanistiche

vito pirrelli

Istituto di Linguistica
Computazionale CNR Pisa

Dipartimento di linguistica
Università di Pavia

lezione 8

(per iniziare...) 

decima lezione

testo e struttura: un'introduzione all' XML

vito pirrelli

Istituto di Linguistica
Computazionale CNR Pisa

Dipartimento di linguistica
Università di Pavia

lezione 8

(continua...) 

dati e struttura

- dal punto di vista del calcolatore il testo è una *sequenza di caratteri alfanumerici* intervallati da *separatori*, come ad esempio lo spazio bianco o un segno di interpunzione, e *caratteri di scorrimento (o controllo) del testo*, come ad esempio il "ritorno carrello" (*carriage return*, per tornare accapo)
- per noi, invece, il testo è *molto di più* di una semplice sequenza di caratteri alfanumerici, separatori e caratteri di controllo (dati)

lezione 8

(continua...)



dati e struttura (II)

- il testo è innanzitutto *struttura* (informazione)
 - ciascun *elemento costitutivo* dei dati testuali entra a far parte di un elemento più grande: i caratteri formano sillabe, le sillabe parole, le parole frasi, le frasi paragrafi, i paragrafi capitoli ecc.
 - i capitoli si compongono a loro volta di numero, titolo e corpo del testo
 - le frasi si compongono di sequenze di parole semanticamente coese dette *sintagmi*
 - le stesse parole presentano al loro interno costituenti più piccoli solitamente chiamati *morfemi*

lezione 8

(continua...)



dati e struttura (III)

- nel testo (come nel linguaggio) questa struttura è quasi sempre *implicita*
- ad esempio, ogni parola del testo non “*mostra*” la sua struttura morfemica interna; analogamente, una frase del testo non presenta esplicitamente la sua *segmentazione* in costituenti sintagmatici
- il calcolatore è in grado di cogliere questa struttura implicita del testo solo se:
 - *conosce* le regole che la governano (o il codice sottostante), oppure
 - questa struttura è rappresentata *esplicitamente* nel testo

lezione 8

(continua...)



un esempio

Scadono i termini per la registrazione delle liste nelle 14 regioni interessate al voto del 3 e 4 aprile. In 42 milioni alle urne — occhiello

Regionali, parte la sfida tra i poli

titolo

Prodi: “Una in più è vittoria”. I Radicali non si presentano — sommario

ROMA — Romano Prodi sistema l'estetica delle regionali per l'Unione: se prendiamo una regione in più, annuncia il leader, è già una vittoria. Il Professore va ad Ancona, a presentare il candidato del centrosinistra Gian Mario Spacca, proprio nel giorno in cui scadono i termini per la presentazione delle liste per il voto del 3-4 aprile prossimo. Parte dunque la competizione nelle 14 regioni italiane (42 milioni di elettori alle urne) fra il centrosinistra e la Cdl, con i radicali che dopo il fallimento delle trattative non saranno presenti con proprie liste. E

Eda oggi la macchina elettorale per le regionali si mette in moto. Riflettori puntati in particolare su Lazio, Piemonte e Puglia, in mano al centrodestra ma dove l'Unione coltiva il sogno del sorpasso. Nel Lazio nella sfida fra il governatore uscente Francesco Storace e il candidato dell'opposizione Piero Marrazzo potrebbero risultare decisivi i voti del terzo incomodo, Alessandra Mussolini. La nipote del duce, con la sua lista di Alternativa sociale, può infatti sottrarre al candidato della destra quel poco che basta a fare il gioco di Marrazzo,

che le previsioni danno a ridosso di Storace. Dodici le liste a sostegno del presidente uscente il partito della Cdl, la lista Storace, Pensionati, Ecologisti Verdi, Costituyente Democratica, Movimento Idea Sociale con Emu, Lista Consumatori, Pri-Pli, Nuovo Psi e Trifoglio. Dieci le liste per Marrazzo (Uniti nell'Ulivo, Verdi, Prc, Pdci, lista Marrazzo, Udcour, Forza Roma, Avanti Lazio, Consumatori Uniti, Italia dei Valori). In Piemonte la lista di disturbo per il centrodestra potrebbe rivelarsi invece quella della Democrazia cristiana, che candida Gianfran-

co Botondi, ago della bilancia fra il governatore uscente Enzo Ciglio e la sfidante dell'Ulivo Mercedes Bresso. In Puglia invece c'è tempo fino al 9 marzo per la presentazione delle liste, ma anche con i “tempi supplementari” i radicali non hanno intenzione di scendere in campo con un proprio simbolo. Unica eccezione resterà così l'Abruzzo, dove alcuni esponenti radicali (come l'immediato antiproibizionista Ignazio Marozzi Rozzi) figurano nella lista dello Sdi di Ottaviano del Turco. (a.r.)

Il Professore vuole più di un confronto tv col premier, ma “chiaro è con arbitri imparziali” — catenaccio

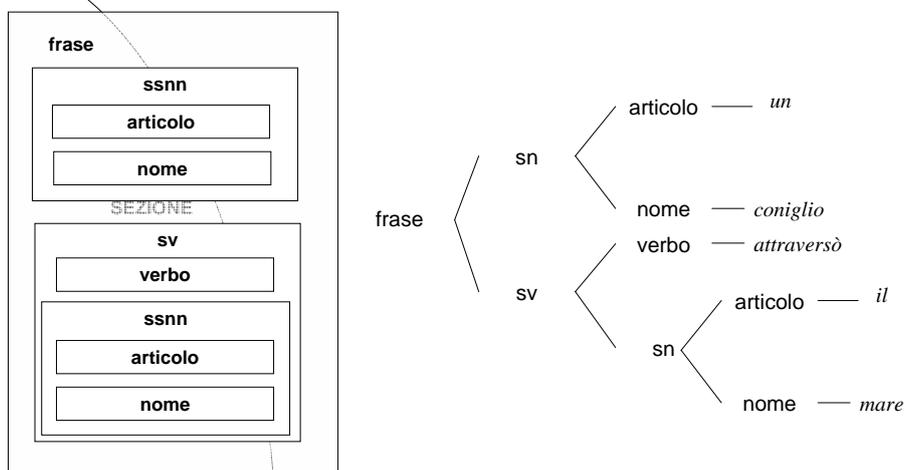
Oggi il rapporto è 8 a 6 per la Cdl. In Lazio, Piemonte e Puglia si profila il testa a testa

lezione 8

(continua...)



la frase come albero



lezione 8

(continua...)



dati e grammatiche

- è possibile, ad esempio, fornire al calcolatore *una serie di regole formali* che definiscono classi possibili di parole, di sintagmi, di frasi, di capitoli ecc.
- *espressioni regolari* formate da sequenze di categorie grammaticali (del tipo *art agg n*) possono essere usate per definire i *sintagmi grammaticalmente corretti* della lingua italiana
 - ◆ $sn \rightarrow \text{predet? art? avv* agg* nome}$
- l'insieme di queste regole definisce quella che viene chiamata *grammatica del testo*
- ogni livello di *analisi linguistica* del testo ha la sua grammatica: esiste una *grammatica morfologica* per la struttura interna delle parole, una *grammatica sintattica* per la struttura delle frasi in costituenti, e così via

lezione 8

(continua...)



dati e "glosse"

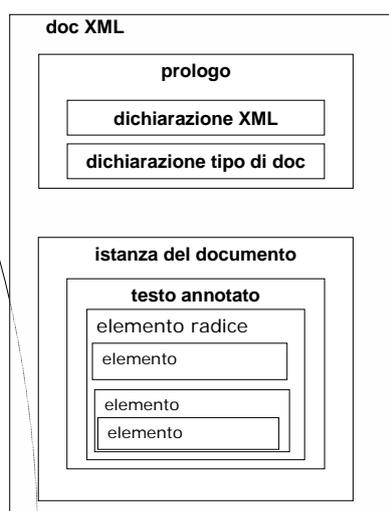
- è possibile *codificare* l'informazione relativa alla struttura dei dati testuali *direttamente nel testo*, mediante l'uso di un *linguaggio di annotazione* o *linguaggio di mark-up*
- l'*XML (eXtensible Markup Language)* è il più famoso tra questi linguaggi
- l'*XML* è nato per rispondere all'esigenza di accompagnare i dati testuali con una serie di "*glosse*" o "*commenti*" che servono a rendere esplicite le relazioni che intercorrono tra i dati stessi
- dal momento che le glosse hanno la funzione di convogliare *informazione riguardo ai dati* testuali in senso stretto, questa informazione viene anche definita come *meta-dati* ("dati sui dati")
- i meta-dati coprono un'ampia gamma di informazioni, anche estremamente eterogenee, che vanno dalla data di creazione del testo, al suo formato e al suo autore, ad una rappresentazione esplicita del suo *contenuto*

lezione 8

(continua...)



struttura di un documento XML



lezione 8

(continua...)



struttura di un documento XML

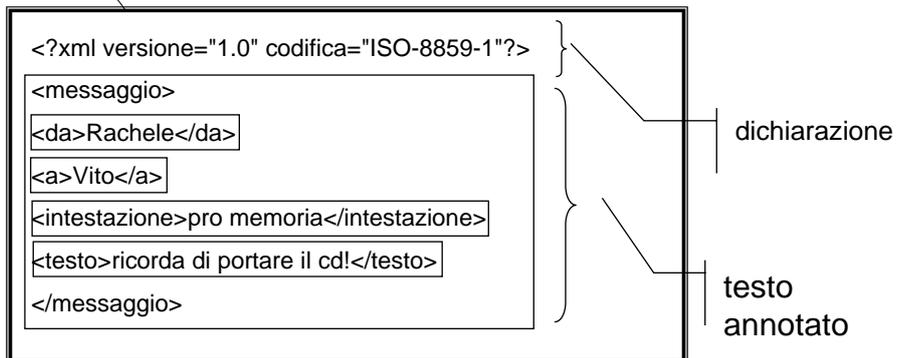
- Un documento XML si articola in due parti:
 - ◆ **prologo**
 - ★ contiene le informazioni che permettono di interpretare il documento come un documento XML, ricollegandolo alla corrispondente DTD (ove esista)
 - ◆ **istanza del documento**
 - ★ racchiude il dato testuale vero e proprio con la relativa marcatura
- **ATTENZIONE!**
 - ◆ il prologo è facoltativo ma è sempre consigliabile inserirlo nei propri documenti per favorire completezza di informazione e possibilità di validazione automatica

lezione 8

(continua...)



un esempio

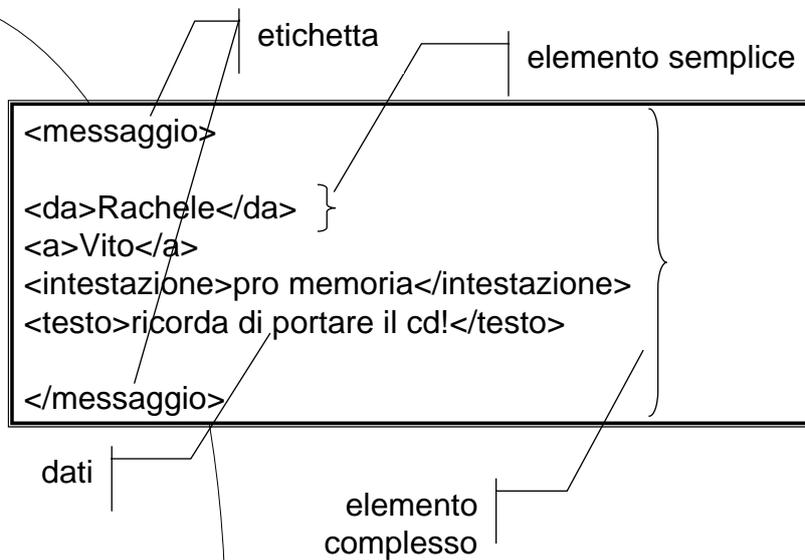


lezione 8

(continua...)



il testo annotato



lezione 8

(continua...)



XML è solo "informazione"

- ricorda, non c'è *niente di automatico* che XML possa fare al di là di strutturare l'informazione implicitamente presente nel testo
- è necessario scrivere dei *programmi* che utilizzino queste informazioni in modo coerente e conforme ad uno scopo
- la forza di XML sta proprio in questa sua *indipendenza* da una particolare applicazione o piattaforma software

lezione 8

(continua...)



XML è informazione "aperta"

- non esiste un repertorio pre-definito di etichette XML
- ciascuno è libero di definire il suo insieme di etichette in modo del tutto autonomo, così come può inventare la propria struttura di dati etichettati
- da questo punto di vista, XML è un *meta-linguaggio di annotazione*, nel senso che può essere usato per definire un linguaggio di annotazione vero e proprio
- inoltre dei documenti in XML sono *espandibili*, dal momento che uno può aggiungere nuovi metadati senza alterare né i dati originari, né i metadati pre-esistenti

lezione 8

(continua...)



XML è informazione "espandibile"

- supponiamo che il nostro messaggio codificato in XML possa essere *interpretato* da un'applicazione di interfaccia, che lo legge come segue:

MESSAGE

To: Vito
From: Rachele

heading: pro-memoria

ricorda di portare il cd!

lezione 8

(continua...)



XML è "espandibile" (II)

- supponiamo inoltre di aver arricchito il nostro messaggio codificato in XML, in modo tale che possa essere indicata anche la data:

```
<messaggio>
<data> 10.5.2003 </data>
<da>Rachele</da>
<a>Vito</a>
<intestazione>pro memoria</intestazione>
<testo>ricorda di portare il cd!</testo>
</messaggio>
```

lezione 8

(continua...)



XML è "espandibile" (III)

- l'applicazione originaria continuerà ad interpretare correttamente il nostro nuovo messaggio, *ignorando* tuttavia il dato annotato, di cui non conosce la traduzione attraverso l'interfaccia:

MESSAGE

To: Vito

From: Rachele

heading: pro-memoria

ricorda di portare il cd!

lezione 8

(continua...)



a cosa serve?

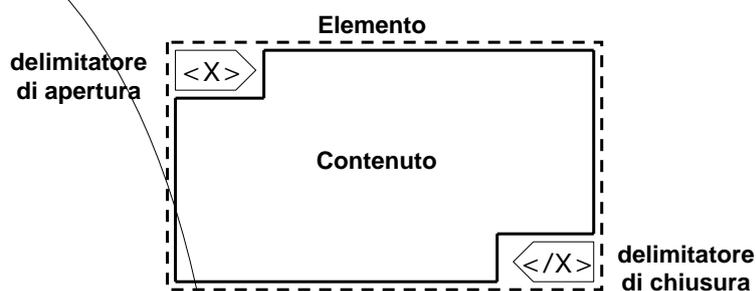
- l'obiettivo primario di XML è quello di *convogliare informazione* in un modo che sia *indipendente da una particolare applicazione o piattaforma software*
- il suo uso principale è come *formato di interscambio dei dati*, o per la *creazione di dati condivisi*, o per la loro memorizzazione in una base di dati

lezione 8

(continua...)



un elemento XML



- gli elementi rappresentano i blocchi costitutivi in cui si articola un testo
- ogni elemento viene marcato in modo esplicito nel testo inserendo un delimitatore di apertura all'inizio dell'elemento e uno di chiusura alla fine
 - ◆ es: `<autore>Carlo Collodi</autore>`

lezione 8

(continua...)



caratteristiche di base

- un elemento di XML incomincia con un *delimitatore di apertura* e finisce con il *delimitatore di chiusura corrispondente*:
 - `<X> dati </X>`
dove X è il nome (o etichetta) dell'elemento
- un nome di elemento non può essere una qualsiasi sequenza di caratteri, ma è soggetto a particolari restrizioni sintattiche
- in un nome XML un carattere minuscolo e il suo corrispondente maiuscolo *sono due lettere diverse*:
 - `<errore> elemento sbagliato </Errore>`
non rappresenta un elemento XML corretto, in quanto le etichette riportate nei delimitatori non sono identiche

lezione 8

(continua...)



il nome di un elemento XML

(fine excursus)



- **norme per la sintassi dei *nomi* degli elementi**
 - ◆ **possono contenere solo lettere, cifre, ., -, _**
 - ◆ **possono iniziare solo con una lettera o con _**
 - ★ nomi consentiti: `<autore.libro>`, `<_autore>`, `<AUTORE-LIBRO>`, `<autore_1>`
 - ★ nomi proibiti: `<1autore>`, `<autore libro>`, `<autore;@?libro>`
 - ◆ non esiste un limite di lunghezza per il nome di un elemento

[a-zA-Z_][a-zA-Z0-9.-_]*

- ◆ **nota: a queste stesse restrizioni soggiace anche il nome di un attributo**

lezione 8

(continua...)



relazione tra elementi XML

- due elementi XML *possono essere annidati* l'uno nell'altro:
 - `<messaggio>`
 - `<testo>`
questo è il testo del messaggio ...
 - `</testo>`
 - `</messaggio>`
- l'elemento `testo` è contenuto nell'elemento `messaggio` per indicare che il testo "questo è il testo del messaggio ..." è proprio il testo di quel messaggio
- l'elemento più esterno è detto *elemento genitore*, quello interno *elemento figlio*
- nota: tutto questo può sembrare *ovvio* e un po' barocco, ma ci consente, tra le altre cose, di *scrivere nello stesso documento più messaggi senza confonderli tra loro*

lezione 8

(continua...)



relazione tra elementi XML (II)

- genitori con più figli:
 - `<messaggio>`
 - `<destinatario> vito </destinatario>`
 - `<testo>`
questo è il testo del messaggio ...
 - `</testo>`
 - `</messaggio>`
- diremo che l'elemento `<testo>` e l'elemento `<destinatario>` sono *fratelli*, in quanto figli dello stesso genitore

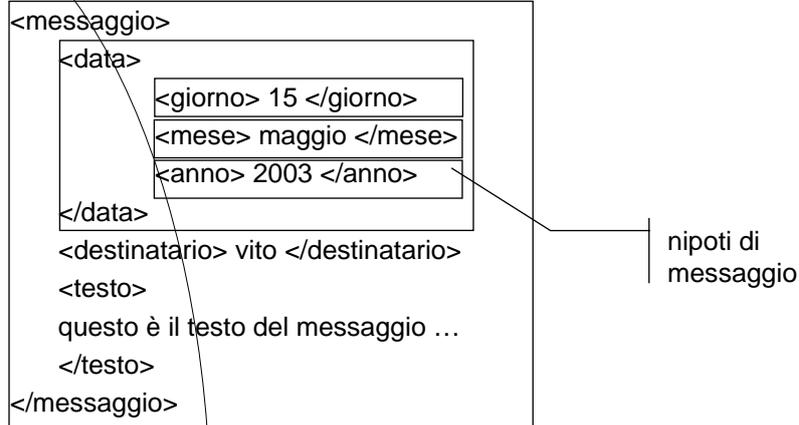
lezione 8

(continua...)



relazione tra elementi XML (III)

- più "generazioni" di elementi:



lezione 8

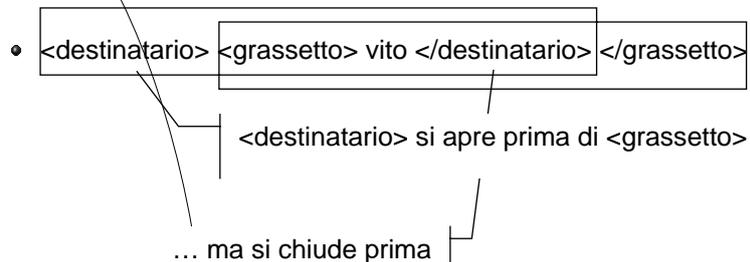
(continua...)



relazione tra elementi annidati

- la relazione di inclusione tra due elementi annidati deve essere sempre *propria*:

esempio di annidamento "improprio"



lezione 8

(continua...)



elemento "radice"

- ogni documento XML ben formato deve contenere un solo elemento che contiene tutti gli altri elementi (*elemento radice*):

esempio di documento senza *radice*

- `<destinatario> vito </destinatario>`
`<testo>`
questo è il testo del messaggio ...
`</testo>`

- in altre parole, un documento XML non può contenere *più di un orfano* (l'elemento radice è per definizione un orfano, ma è isolato)

lezione 8

(continua...)



nota a margine

- ogni documento XML può contenere una o più "*note a margine*" dette anche, in gergo tecnico, *commenti*
- questo tipo di note *non riguardano direttamente i dati* da annotare, e come tali sono ignorate da eventuali applicazioni
- possono fornire tuttavia informazioni preziose riguardo alle intenzioni dell'annotatore, a suoi eventuali dubbi, pro-memoria ecc. Il loro destinatario è tipicamente un lettore umano

esempio di nota a margine

- `<destinatario> vito </destinatario>`
`<testo>`
questo è il testo del messaggio ...
`</testo>`
`<!-- NB: questo è solo un esempio di annotazione, non un documento XML sintatticamente ben formato...-->`

lezione 8

(continua...)



anatomia di un elemento XML

- il delimitatore di apertura di un elemento XML può contenere una o più coppie "attributo/valore":



lezione 8

(continua...)



valori di un attributo

- ad ogni attributo deve essere assegnato *al massimo un valore* (in genere, un attributo non può ricevere una lista di valori)
- ogni valore *deve essere scritto tra virgolette* (doppie o semplici)
- nel caso in cui un valore contenga al suo interno delle virgolette, allora diventa obbligatorio differenziarle da quelle più esterne

esempio

- `<documento titolo="L'importanza di chiamarsi 'Ernesto'">`
`<autore> Oscar Wilde </autore>`
`<testo formato="corsivo">`
testo del documento ...
`</testo>`
`</documento>`

lezione 8

(continua...)



a cosa serve un attributo?

- un attributo può essere qualsiasi cosa
- in teoria, sarebbe possibile codificare qualsiasi elemento come un attributo o un insieme di attributi

esempio

```
<messaggio data="10.5.2003" da="Rachele" a="Vito"
intestazione="pro memoria" testo="ricorda di portare il
cd!">
</messaggio>
```

lezione 8

(continua...)



a cosa serve un attributo? (II)

- in pratica, un attributo non è un *contenitore (etichettato)* di dati (giacché questa funzione è già svolta da un elemento XML), ma una *glossa* dei dati
- più tecnicamente, l'uso indiscriminato di attributi presenta alcuni svantaggi:
 - un attributo non può contenere dati multipli
 - un attributo non può descrivere strutture complesse
 - è consigliabile che un attributo assuma un numero finito o sufficientemente piccolo di valori; pertanto non è espandibile *ad libitum*
- eccezione apparente: è buona pratica usare un attributo come *identificatore univoco di un elemento*; in questo caso l'attributo non è altro che un *contatore*

lezione 8

(continua...)



Il prologo: la dichiarazione XML

- Si articola a sua volta in due parti
 - ◆ dichiarazione XML (*XML Declaration*)
 - ★ fornisce informazioni sulla versione di XML utilizzata per scrivere il documento
 - attributo `version` (il valore attuale è "1.0")
 - ★ specifica il set di caratteri utilizzato per la codifica
 - attributo `encoding` (il valore è il nome di un seto di caratteri, es. "iso-8859-1")
 - se l'attributo è omesso, il documento è interpretato come codificato in UTF-8
 - ★ indica se al documento è associata o meno una DTD esterna
 - attributo `standalone` (vali valori sono "yes" o "no"); se omesso è interpretato come "no"

```
<?XML version="1.0" encoding="iso-8859-1" standalone="no"?>
```

lezione 8

(continua...)



Il prologo: la DTD

- ◆ dichiarazione del tipo di documento (*DocType Declaration*)
 - ★ specifica qual è l'elemento radice dell'istanza del documento che segue
 - ★ associa al documento XML la relativa DTD che può essere interna o esterna al documento
 - ★ è opzionale; deve essere aggiunta solo quando bisogna validare il documento XML rispetto a una DTD

```
<!DOCTYPE elemento_radice DTD>
```

lezione 8

(continua...)



Document Type Declaration (DTD)

esempio

- <!DOCTYPE persona [
 <!ELEMENT persona (identità_anagrafica, professione*)>
 <!ELEMENT identità_anagrafica (nome, cognome)>
 <!ELEMENT nome (#PCDATA)>
 <!ELEMENT cognome (#PCDATA)>
 <!ELEMENT professione (#PCDATA)>
>

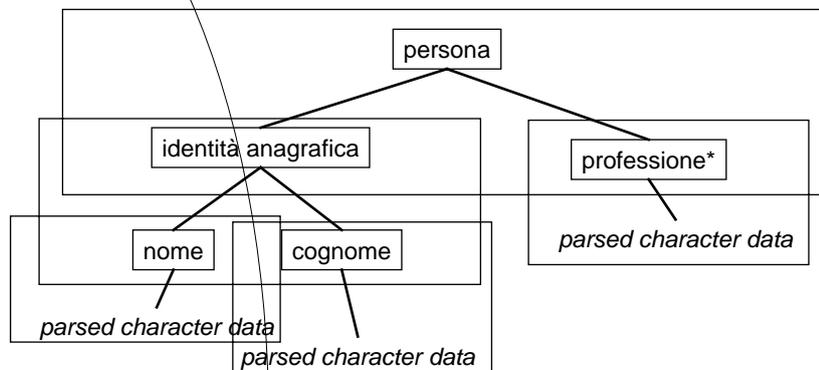
lezione 8

(continua...)



DTD: la struttura ad albero

- <!ELEMENT persona (identità_anagrafica, professione*)>
- <!ELEMENT identità_anagrafica (nome, cognome)>
- <!ELEMENT nome (#PCDATA)>
- <!ELEMENT cognome (#PCDATA)>
- <!ELEMENT professione (#PCDATA)>



lezione 8

(continua...)



DTD: la struttura ad albero

```
<!ELEMENT persona (identità_anagrafica, professione*)>  
<!ELEMENT identità_anagrafica (nome, cognome)>  
<!ELEMENT nome (#PCDATA)>  
<!ELEMENT cognome (#PCDATA)>  
<!ELEMENT professione (#PCDATA)>
```

RICORDA! come all'anagrafe, anche nella DTD ogni padre dichiara i propri figli, e **NON** anche i figli dei propri figli!

lezione 8



DTD ed espressioni regolari

- nella definizione di un elemento XML è consentito usare con certe restrizioni la sintassi delle espressioni regolari per definire strutture di annidamento anche complesse

esempio

```
<!ELEMENT persona (identità_anagrafica, professione*)>  
<!ELEMENT identità_anagrafica ((nome, cognome)|soprannome, data)>  
<!ELEMENT data (giorno, mese, anno)>  
<!ELEMENT giorno (#PCDATA)>  
<!ELEMENT mese (#PCDATA)>  
<!ELEMENT anno (#PCDATA)>  
<!ELEMENT nome (#PCDATA)>  
<!ELEMENT cognome (#PCDATA)>  
<!ELEMENT soprannome (#PCDATA)>  
<!ELEMENT professione (#PCDATA)>
```

lezione 8

(continua...)



elementi con contenuto misto

- all'interno di documenti di tipo narrativo un singolo elemento XML spesso contiene sia elementi figli sia dati di tipo carattere.

esempio

```
<testo> all'interno di documenti di tipo narrativo un singolo
<termine>elemento</termine> <sigla>XML</sigla> spesso contiene sia
<termine><composto>elementi figli</composto></termine> sia dati di
<composto>tipo carattere</composto></testo>
```

lezione 8

(continua...)



elementi con contenuto misto (II)

- è possibile definire il contenuto dell'elemento <testo> come *misto* nel seguente modo

esempio

```
<!ELEMENT testo (#PCDATA | termine | sigla | composto)*>
<!ELEMENT termine (#PCDATA | composto)>
<!ELEMENT composto (#PCDATA)>
<!ELEMENT sigla (#PCDATA)>
```

```
<testo>all'interno di documenti di tipo narrativo un singolo
<termine>elemento</termine> <sigla>XML</sigla> spesso contiene sia
<termine><composto>elementi figli</composto></termine> sia dati di
<composto>tipo carattere</composto></testo>
```

lezione 8

(continua...)



dove si trova la DTD?

- una DTD può essere *salvata in un documento indipendente* (con estensione dtd) e *richiamata* dall'interno della lista di persone attraverso un *identificatore univoco* (o *URI Uniform Resource Identifier*), rappresentato dal nome del documento .dtd e dal suo *path*:

esempio

```
<!DOCTYPE person SYSTEM "http://ibiblio.org/xml/dtds/person.dtd">
<persona>
  <identità_anagrafica>
    <nome> Pablo </nome>
    <cognome> Picasso </cognome>
  </identità_anagrafica>
  <professione> pittore </professione>
</persona>
```

lezione 8

(continua...)

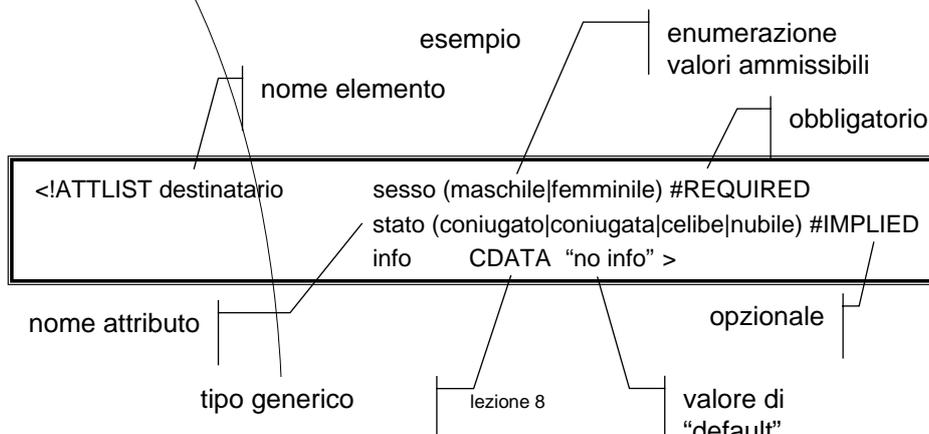


Dichiarazione di attributi

(continua...)



- oltre a definire la struttura di un elemento XML se ne possono dichiarare anche gli *attributi*, specificando il loro nome, il tipo di valori e il loro status (obbligatori o opzionali)



principali tipi di valori di un attributo

- CDATA dati di tipo carattere*
- (en1|en2|..)
- ID il valore è un elemento di una lista disgiuntiva
- IDREF il valore è un identificatore univoco (nome XML)
- IDREFS il valore è un riferimento a un ID
- NMTOKEN il valore è una lista di riferimenti a ID°
- NMTOKENS il valore è un nome XML valido**
- ENTITY il valore è una lista di NMTOKEN°
- ENTITIES il valore è l'abbreviazione di un testo
- il valore è una lista di ENTITY°

- * è il tipo di valori meno restrittivo (non può tuttavia includere i caratteri "<" e "&")
- ° "idrefs", "nmtokens" ed "entities" sono le sole liste possibili come valori di un attributo
- ** a differenza di un nome XML, tuttavia, NMTOKEN può iniziare con un punto o un trattino e può contenere anche '.'

lezione 8

(fine excursus)



cos'è un corpus annotato?

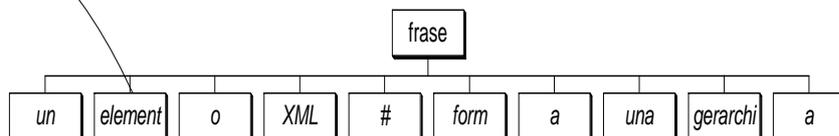
*un
elemento
XML
forma
una
gerarchia*

lezione 8

(continua...)



cos'è un corpus annotato?(II)

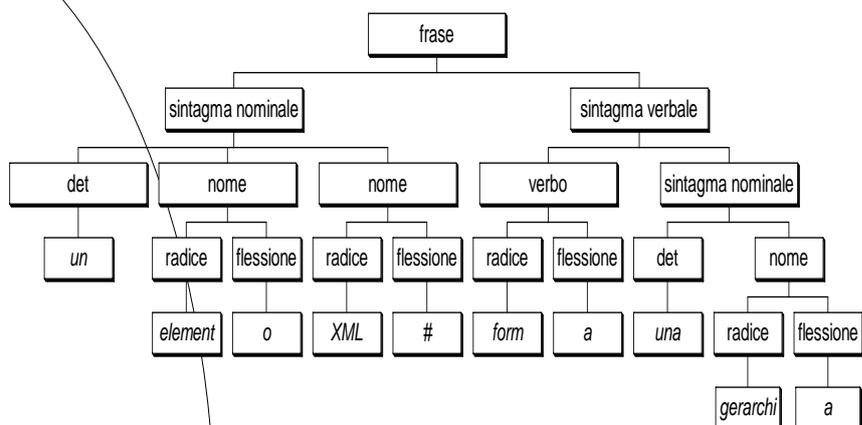


lezione 8

(continua...)



cos'è un corpus annotato? (III)



lezione 8

(continua...)



perché un corpus annotato?

- perché è un utile strumento di indagine linguistica:
 - ◆ rende esplicita l'informazione linguistica implicitamente contenuta nel testo
 - ◆ è interrogabile automaticamente
 - ◆ consente di verificare su base empirica il grado di copertura e la coerenza di uno schema di annotazione linguistica
 - ◆ lo rende condivisibile e verificabile da altri
 - ◆ fornisce uno strumento raffinato di indicizzazione del testo, la cui utilità prescinde dalla condivisione dello schema di annotazione in senso stretto
 - ◆ può essere usato come base di dati per lo sviluppo di grammatiche

lezione 8

(continua...)



desiderata per l'annotazione del testo

- replicabilità dell'annotazione
- separabilità dell'annotazione dal dato (il dato è permanente e stabile, l'annotazione no)
- leggibilità
- documentabilità
- inter-traducibilità teorica
- massima utilizzabilità vs minima manualità

lezione 8

(continua...)



perché XML?

- l'uso di XML per l'annotazione linguistica del testo consente di definire gerarchie di elementi testuali anche molto complesse
- questa possibilità porta a concepire una tipica annotazione XML come un'unica gerarchia che include un intero documento
- tuttavia:
 - ◆ il documento da annotare può essere anche molto grande
 - ◆ può non essere scrivibile
 - ◆ può richiedere più livelli di annotazione gerarchicamente non allineati

(per saperne di più clicca sulle parole evidenziate in giallo!)

lezione 8

(continua...)



XML e annotazione "stand-off"

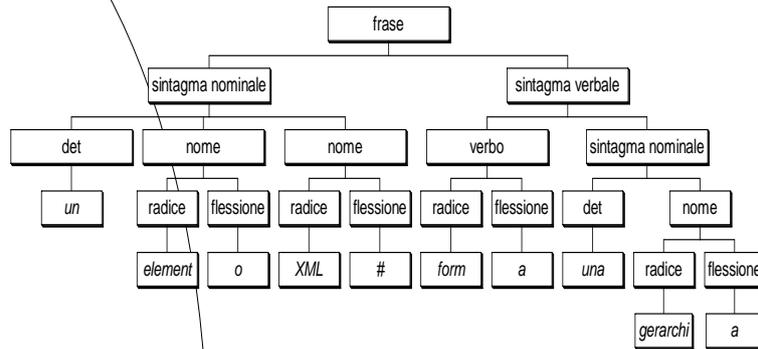
- l'uso di XML consente di aggirare questi problemi
- in XML l'annotazione dei dati è logicamente distinta dalla loro presentazione
- XML fornisce più di una possibile "vista" sugli stessi dati
- con XML è possibile mantenere il piano dell'organizzazione fisica del dato distinto dal livello della sua organizzazione logica
- lo strumento sintattico che consente a dati e metadati di risiedere fisicamente in due posti distinti ma di fare riferimento logico gli uni agli altri è il linking da un elemento di un file XML ad un elemento di un altro file XML
- potere espressivo di una base di dati

lezione 8

(continua...)



<!ELEMENT F (SN, SV)>
 <!ELEMENT SN (det?, aggettivo?, nome+, aggettivo?)>
 <!ELEMENT det (il|lo|la|i|gli|le|un|uno|una)>
 <!ELEMENT aggettivo (radice, flessione)>
 <!ELEMENT nome (radice, flessione)>
 <!ELEMENT radice (#PCDATA)> ...



lezione 8

(fine excursus)



XML su internet

- <http://http://www.w3schools.com/>

lezione 8

fonti e link utili

fine ottava e ultima lezione

ringraziamenti

<http://www.ateneonline.it/>

"Introduzione ai sistemi informatici, 2/ed" (D. Sciuto, G. Buonanno, W. Fornaciari, L. Mari)

"Informatica di base" (Curtin D.P., Foley K., Sen K., Morin C.)

"Basi di Dati" (Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Riccardo Torlone)

<http://www.math.unipi.it/~frossi/assembly.ppt>

http://www.dsi.unifi.it/~costa/lucidi_3_02/Lezione2.pdf

<http://www.w3schools.com/>

<http://foxdrake.ilc.cnr.it/webtools/>

<http://etext.lib.virginia.edu/helpsheets/regex.html>

<http://digilander.libero.it/nick47/index.htm#rife1>

<http://computer.howstuffworks.com/boolean1.htm>

<http://www.unicode.org/>

<http://www.liceofoscarini.it/studenti/probabilita/index.html>

<http://digilander.libero.it/shppotta/dice.htm>

http://apollo.vc.unipmn.it/~ling_gen/opening.htm

<http://www.routledge.com/linguistics/using-comp.htm> lezione 8

<http://www.textworld.net/>