

# *I Linguaggi di markup per l'elaborazione del linguaggio naturale*

Liliana Galdi  
Prof. Amedeo Cappelli

## Indice

- Cos'è il markup.
- Storia dei linguaggi di markup.
- XML e suoi usi.
- Livelli di annotazione + esempi
- Un esempio di annotazione XML

## PRIMA PARTE

### INTRODUZIONE E STORIA DEI LINGUAGGI DI MARKUP

## Cos'è il markup?

- Il *markup* è un insieme di convenzioni per decidere quali parti di documento vanno trattate in modo diverso dalle altre parti.
  - esempio: l'uso delle stili nei documenti scritti.
- L'idea è di inserire specifiche sequenze di caratteri o codici (detti markup tags o tags) nel testo per permettere ai programmi di distinguere le varie parti di un documento.
  - ad esempio, per decidere come visualizzarlo

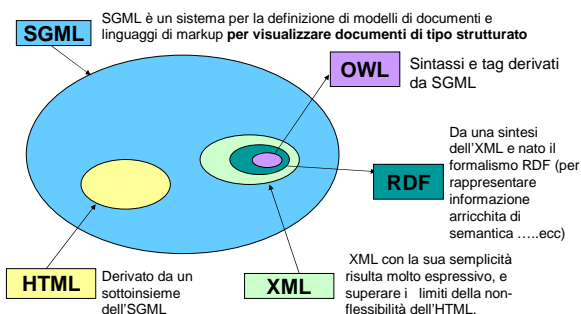
## Cosa sono i linguaggi di markup?

- Un linguaggio di markup è quindi un *sistema formale* (insieme di convenzioni) per scambiare e pubblicare informazioni (in formato testo) in modo strutturato.
- Esistono diversi linguaggi di markup, per rappresentare diversi tipi di documenti.

## Perché utilizzare LdM

- Uno dei motivi principali, per lo sviluppo di marcature leggibili da computer su testi, è la **necessità di sviluppare corpora testuali**, per studi linguistici, lessicografici, letterari e statistici.

## ( Schema )



## Un po' di storia (1)

- (1968-70):
  - Alcuni ricercatori dell'IBM iniziano a sviluppare l'idea di un markup generico, *Generalized Markup Language (GML)*, un linguaggio di markup per la gestione interna della società.

## Un po' di storia (2)

- (1986): **SGML** (Standard Generalized Markup Language) è standard ISO (International Standards Organization)
- E' un meta-linguaggio **per definire linguaggi**
  - (Un esempio di linguaggio derivato da SGML è HTML (HyperText Markup Language) utilizzato nel WWW)
- Non è un LdM, ma un linguaggio con cui definiamo LdM
- Non riesce a soddisfare tutte le richieste di markup per arricchire testi, ma **fornisce una sintassi per definire il linguaggio** adatto.
- SGML non sa cos'è un paragrafo, una lista, un titolo, ma fornisce una grammatica che ci permette di definirli

## Un po' di storia (3)

- (1991): **HTML** (Hyper Text Markup Language) è un linguaggio per la definizione di ipertesti.
  - i markup tag controllano principalmente la visualizzazione del contenuto

- Un documento HTML è composto da tre parti

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE> Il mio primo documento HTML</TITLE>
  </HEAD>
  <BODY>
    <P>Hello world!
  </BODY>
</HTML>
```

Info. sulla versione HTML e sul DTD di riferimento

Un corpo, che contiene un documento vero e proprio (elemento BODY)

Una sezione dichiarativa di intestazione (elemento HEAD)

## Un po' di storia (4)

- **XML** (eXtensible Markup Language)
- Il suo sviluppo ha avuto inizio nel 1996.
- XML 1.0: standard W3C (*World Wide Web Consortium*) dal febbraio del 1998
- Assicura che i **dati strutturati** siano **uniformi e indipendenti dalle applicazioni**.
- E' un **meta linguaggio testuale per testi web**.
- E' un sottoinsieme dell' SGML, racchiude il meglio di SGML, per superare i limiti della non-flessibilità dell'HTML.

## Un po' di storia (5)

- **RDF** - Resource Description Framework
- **Standard per fare semplici descrizioni.**  
Quello che XML è per la sintassi, RDF lo è per la semantica: un insieme chiaro di regole, per fornire informazioni descrittive.
- **Sviluppato** principalmente per il **"Semantic web"**.
- Modello generale per fare delle "asserzioni semantiche" su oggetti Web.

## Un po' di storia (6)

- **OWL** (web ontology language)
- Nasce come l'evoluzione del DAML+OIL.
- E' un linguaggio per definire ontologie strutturate basate sul Web che permettano maggiore integrazione ed interoperabilità di dati tra applicazioni
- E' utilizzato nei casi in cui l'informazione contenuta in documenti deve essere elaborata da un'applicazione, infatti, **permette di esplicitare e rendere elaborabile automaticamente l'informazione** contenuta nei documenti;
- E' **basato sulla Logica Descrittiva**.

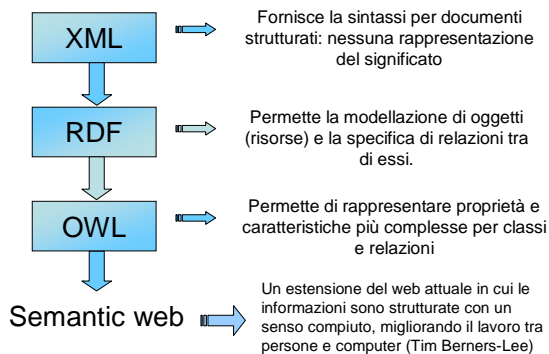
## ...Ancora sull'OWL...

- E' un linguaggio per esprimere espressioni ontologiche sul web.

**Ontologia** = Una *descrizione formale esplicita* di un *dominio* di descrizione



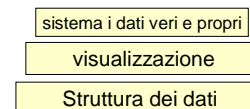
## ... OWL e il semantic web ...



## L'importanza dei LdM nel Semantic Web (1)

I **problemi ed i limiti dell'attuale Web** potranno forse essere superati grazie alla diffusione di linguaggi di markup.

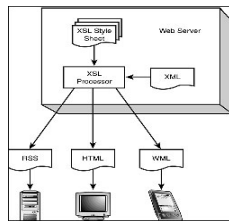
Uno di questi linguaggi sarà **XML**, che permette di gestire i dati in modo più agile, perché modula l'informazione a più livelli:



## L'importanza dei LdM nel Semantic Web (2)

Questo ci permette di:

- Avere gli stessi dati con due visualizzazioni differenti.
- Accedere unicamente ai dati individuati da una certa marcatura.
- Stabilire per diversi documenti la stessa struttura.



Secondo le intenzioni del W3C sulla tecnologia XML dovrà costruirsi nei prossimi anni un **Web più "intelligente"**, capace di gestire automaticamente le nostre richieste di informazione

## SECONDA PARTE

XML

## XML (1)

- Per XML si intende un **linguaggio di annotazione** che fornisce un formato per la **descrizione di dati strutturati**.
- Ciò consente di dichiarare con maggiore precisione il contenuto dei dati, e di ottenere risultati più significativi nelle ricerche, eseguite su diverse piattaforme.

## XML (2)

- *Definisce* in modo non ambiguo la *struttura di dati* contenuti in un documento
- L'*XML* risulta essere un linguaggio *facile da utilizzare ed elaborare*

### Obiettivi di XML (1)

- Deve essere utilizzato in modo **semplice** su internet
- Deve supportare un gran numero di applicazioni
- Deve essere **compatibile** con SGML
- Lo sviluppo di programmi che elaborino documenti XML deve essere **facile**

### Obiettivi di XML (2)

- I documenti XML dovrebbero essere **leggibili da un uomo** e ragionevolmente chiari
- La progettazione XML dovrebbe essere **rapida**
- La progettazione XML deve essere formale e **concisa**
- I documenti XML devono essere **facili da creare**

### XML: Uno standard per la rappresentazione dei dati (1)

- Grazie all' XML, e alle sue estensioni, è stato possibile definire uno **standard per la rappresentazione dei dati** che consentirà ad Internet di espandersi allo stesso modo in cui ciò è stato possibile alcuni anni fa grazie allo standard HTML per la visualizzazione.
- Ciò aprirà **nuove prospettive d'utilizzo**, tutte basate su una rappresentazione standard **per l'elaborazione di dati strutturati**.

### XML: Uno standard per la rappresentazione dei dati (2)

- il linguaggio XML renderà possibile una nuova generazione di applicazioni Web per la *visualizzazione e la manipolazione di dati*.
- Lo standard per i dati diventerà il veicolo di:
  - transazioni aziendali
  - pubblicazione dei profili personali
  - collaborazione automatizzata e condivisione dei DB
  - anamnesi cliniche e dati della ricerca farmaceutica
  - schede sui semiconduttori
  - ordini di acquisto

## Documenti XML

- Lo standard XML è un formato testuale, *ideato specificamente per memorizzare e trasmettere dati*.
- Un documento XML:
  - contiene del testo annotato con tag.
- Formato da:
  - tag iniziale(<title>), una tag finale(</title>)
  - informazioni,(*contenuto*), comprese tra i due tag

## Concetto di base

Isolare le parti che compongono un documento, in modo da riuscire ad accedere singolarmente ad esse per elaborarle.



dotare il documento di una **struttura**.

## Esempio (1)

```
<organico>
  <persona>
    <nome>Mario</nome>
    <cognome>Rossi</cognome>
    <e-mail>mrossi@rossispa.it</e-mail>
    <telefono>062345653</telefono>
  </persona>
  <persona>
    <nome>Carlo</nome>
    <cognome>Bianchi</cognome>
    <e-mail>mbianchi@rossispa.it</e-mail>
    <telefono>063343253</telefono>
  </persona>
  ...
</organico>
```

## Esempio (2)

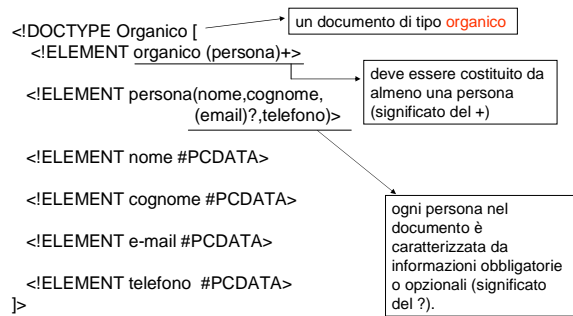
L'esempio precedente in XML è definito:

**istanza di documento**

Le "regole" che l'istanza di documento deve rispettare sono definite da un altro tipo di componente di XML:

**Document Type Definition (DTD)**

### Esempio (3)



### DTD

- Specifica quali sono le strutture ammesse nell'istanza.
- E' visto come una **grammatica** che, attraverso il procedimento di riscrittura, genera il documento stesso: il documento è del "tipo" definito dal DTD

### Vantaggi di un documento strutturato

La strutturazione dei documenti è un aspetto importante per l'elaborazione del linguaggio, ciò permette di:

- **Elaborare i documenti in modo automatico**
  - archiviazione dei documenti
  - estrazione dati da pagine XML
- **Elaborazione automatica del documento**
  - (ricerca di contenuti)
- **Rendere il documento facilmente recuperabile ed interpretabile.**

### Archiviazione dei documenti (1)

Portare un documento XML in un database risulta essere una cosa molto semplice, infatti un documento XML risulta essere una sorta di base di dati.

Quindi la struttura di un documento XML permette una facile archiviazione.



## Archiviazione dei documenti (2)



L'archiviazione risulterà tanto dettagliata quanto risulta fine la granularità dei documenti, superando così gli inconvenienti della archiviazione dei documenti come file interi.

## Estrarre informazioni (1)

Nei documenti XML si può identificare, elaborare ed estrarre qualsiasi parte di documento isolata tra due tag:

*più fine è il dettaglio della struttura del documento  
(tecnicamente si parla di **granularità**)*



*maggiori saranno le possibilità di isolare  
informazioni utili nelle pagine*

## Estrarre informazioni (2)

- riprendendo l'esempio precedente dell'organico se strutturo l'informazione telefono in:

```
<telefono> 06245671</telefono>
```

posso **estrarre** il numero intero.

- Se lo strutturo come:

```
<telefono>
```

```
<prefisso>06</prefisso>
```

```
<numero>245671</numero>
```

```
</telefono>
```

posso **accedere ai singoli componenti** del numero.

## Estrarre informazioni (3)

Esistono tantissimi strumenti di estrazione, i **parser**, la maggior parte di questi è di tipo assolutamente generico: dato il documento XML, se conoscono il DTD cui fanno riferimento, possono restituire non solo i singoli elementi del documento ma anche **tutta la struttura gerarchica**.

## Elaborazione automatica

**La struttura gerarchica** avvia una **catena di elaborazione automatica** del documento

**Esempio:** sono in fase di realizzazione sistemi di gestione automatica del workflow aziendale.

Ogni processo aziendale viene accompagnato dalla relativa documentazione che passa di elaborazione in elaborazione.

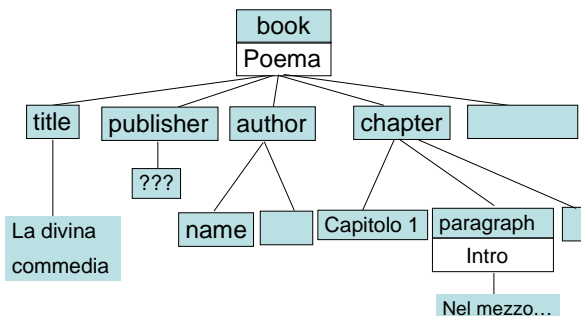
L'elaborazione automatica **elimina il data entry a mano (elimina quindi la maggior fonte di errori sui dati)**

## Una vista concettuale dell'XML (1)

Un documento XML è un insieme ordinato di nodi etichettati, organizzato gerarchicamente con una struttura ad albero.

```
<book category="Poema">
  <title>La divina commedia</title>
  <publisher>???
```

## Una vista concettuale dell'XML (2)



PERTE TERZA

LdM per l'Elaborazione del Linguaggio Naturale

## Ling. di markup per l'Elaborazione del Linguaggio Naturale

- Alcuni tra gli usi più diffusi ed interessanti dei linguaggi di markup sono certamente:
  - Annotazione linguistica di testi
  - Rappresentazione del contenuto per il recupero di informazione
- E' quindi possibile dare una struttura all'**informazione** e renderla **più** facilmente **recuperabile ed elaborabile**

## Annotazione linguistica

- Con il termine "annotazione linguistica" ci si riferisce ad ogni notazione descrittiva o analitica che possa essere applicata a dati linguistici raccolti sotto forma di dati testuali o di dati espressi come funzioni temporali (registrazioni audio, video).
- Le notazioni possono includere le trascrizioni di ogni tipo (dalle caratteristiche fonetiche alle strutture della frase, del discorso o del dialogo), "*part-of-speech*" e altre specifiche annotazioni, analisi sintattiche, annotazioni incrociate con multi-riferimenti, e così via.
- Il focus è sia sui tool che sono stati utilizzati per la costruzione di basi di dati annotate linguisticamente, sia sui *formati* comunemente adottati da tali tool e da tali database

## Come eseguire una annotazione (1)(specifiche per l'annotazione)

- Leech (1993): sette criteri per una annotazione adeguata.

- 1) Deve essere possibile eliminare la annotazione dal testo e tornare ad un corpus non annotato.

Es: Clara\_NP1raccoglie\_VVZfunghi\_NN2

<NP1>Clara</NP1><VVZ>raccoglie</VVZ> <NN2>funghi</NN2>

``Clara raccoglie funghi``

## Come eseguire una annotazione (2)

- 2) Viceversa, deve essere possibile separare l'annotazione dal testo e conservarla da qualche altra parte (p.es. su un altro file).

File 1  
<NP1>Clara</NP1>  
<VVZ>raccoglie</VVZ>  
<NN2>funghi</NN2>

File 2  
``Clara raccoglie funghi``

ricavo la regola sintattica

- 3) Le annotazioni devono essere basate su specifiche disponibili all'utente finale, che spieghino il significato delle etichette usate.

Ad esempio:

- VVZ ==> verbo transitivo di modo finito

### Come eseguire una annotazione (3)

- 4) Deve essere ben specificato *come, quando e da chi* sono state fatte le annotazioni.  
In particolare, deve essere chiaro se sono state fatte a macchina o a mano, se sono state corrette a mano e quante volte.
- 5) Deve essere reso chiaro all'utente finale che le annotazioni non sono infallibili, sia per la possibilità di errori umani, sia perché annotare un testo significa comunque interpretarlo

### Come eseguire una annotazione (4)

- 6) Gli schemi di annotazione devono essere basati su principi il più possibile condivisi e indipendenti dai dettagli di una teoria linguistica testuale

N.B. Questo criterio pone numerosi problemi appena si lascia il piano delle pure annotazioni lessicali e si decide p.es. di annotare la struttura sintattica degli enunciati o il loro ruolo in un dialogo.

- 7) Nessuno schema di annotazione dovrebbe essere considerato a priori "quello standard".

N.B. Uno dei principali problemi è riuscire a realizzare un testo annotato adatto a soddisfare tutte le richieste di un utente.

### Livelli di annotazione

- E' possibile annotare testi a diversi livelli:
  - tipografico
  - fonologico
  - lessicale
  - morfologico
  - sintattico
  - semantico
  - pragmatico

### Annotazione tipografica

- E' il primo passo per un adeguata marcatura del testo.
- Parte da convenzioni tipografiche per poi arrivare a più astratte caratteristiche interpretative e semantiche.

**ES:** Carlo sedette su una sedia  
**Carlo sedette su una sedia**  
**carlo sedette su una sedia**

## Annotazione fonologica

- Passare dal suono ad una sua rappresentazione, o ad una digitalizzazione di un input vocale per poi elaborarlo e/o analizzarlo, non è certo compito facile.
- L' **ORGANIZZAZIONE** e la **CLASSIFICAZIONE** dei suoni che hanno una funzione distintiva all'interno di un dato sistema linguistico



Ci informa su quali suoni facciano parte di una lingua e quali no.

## Fonologia

- Contiene le regole **FONOTATTICHE**
- Ci informano sulle **COMBINAZIONI DI SUONI** e **RESTRIZIONI** che possono occorrere all'interno di una parola

Esempi:

```
<Tono ascendente>
<Volume alto>
  "Cosa?"
</Volume alto>
</Tono ascendente>
```

```
Gnocchi:
<occlusione nasale>
  Gn </occlusione nasale>
<vocale centrale> o <vocale
centrale>
```

## Annotazione lessicale

- Stretta conseguenza dell' *Analisi lessicale* che prevede:
  - RICONOSCIMENTO DI FORME (tokens)
- Questa è una fase fondamentale in quanto fornisce i dati per l' Annotazione lessicale (tags)

Es. `<token> Tasso di interesse </token>`  
`<token> Banca di Italia</token>`

## Esempi di annotazione morfologica

**Sostantivo(Radice("gnocch",desin.masch.plur.(i)))**

```
<w type="U">C'è</w>
<w type="PD">I'</w>
<w type="V">hai</w>
<w type="AT">un</w>
<w type="N">bar</w>
<w type="PN">?</w>
```



Simbolo/Catg.	Esempio
N (nome)	Cane, Mario
PD (pronome/determinatore)	Te ,I' , questo ...
V (verbo)	Hai, è , visto
PN	?, !, :
AT (articolo)	Una ,la ..

## Annotazione sintattica (1)

- Con l'**annotazione sintattica** riusciamo ad organizzare un testo in funzione delle proprie componenti sintattiche.
- Un testo annotato sintatticamente fornisce la base per elaborazioni più complesse e profonde.

## Annotazione sintattica (2)

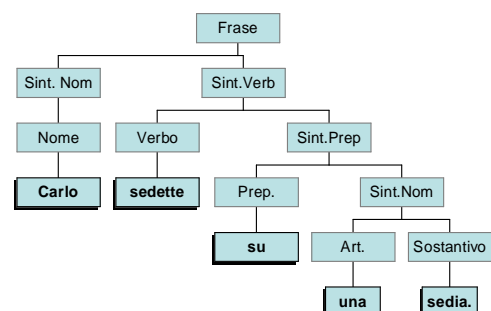
- Una volta identificata la categoria di ciascuna parola, è possibile pensare di marcare il testo in modo da mettere in evidenza la *struttura sintattica* delle parole che lo compongono.
- Questo processo (parsing) comporta l'esistenza di etichette che si riferiscono a *gruppi di parole (costituenti sintattico o sintagmi)*

## Esempio (1)

-Carlo sedette su una sedia. (Usando una marcatura XML)

```
<Frase>
  <Sint.Nom>
    <Nome> Carlo</Nome>
  </Sint.Nom>
  <Sint.Verb>
    <Verbo> sedette </Verbo>
  <Sint.Prep>
    <Prep> su </Prep>
  <Sint.Nom>
    <Art.> una </Art.>
    <Sostant.> sedia. </Sostant.>
  </Sint.Nom>
</Sint.Prep>
</Sint.Verb>
</Frase>
```

## Esempio (2)



## Annotazione semantica

- Attraverso *annotazione semantica* si possono dedurre diversi risultati per le frasi.
- E' possibile catturare informazione semantica a partire da una grammatica generativa, ma è abbastanza intuitivo che un approccio di questo genere è guidato in primo luogo dalla sintassi.

## Esempio (1)

Vogliamo rappresentare in un documento XML una lista di dipendenti di una azienda.

```
< dipendenti >
  < dipendente >
    < nome > Marco < / nome >
    < cognome > Cardinali < / cognome >
    < ufficio > Segreteria < / ufficio >
  < / dipendente >
  < dipendente >
    < nome > Daniele < / nome >
    < cognome > De Luca < / cognome >
    < ufficio > Ricerca e sviluppo < / ufficio >
  < / dipendente >
< / dipendenti >
```

## Esempio (2)

Con l'aggiunta di "*ricchezza semantica*", si rende possibile effettuare una ricerca sui documenti del tipo:

Cerca tutti i dipendenti di nome Marco

senza correre il rischio di ottenere risultati in cui Marco non sia il nome di una persona ma, ad esempio, il nome di un prodotto:

```
< prodotto >
  < nome > Marco < / nome >
  < tipo > maglia < / tipo >
< / prodotto >
```

## Annotazione pragmatica (1)

- Consiste nell'identificare la funzione che un segmento ha nei confronti del contesto, inteso come contesto verbale.

## Annotazione pragmatica (2)

- Ai fini del processo di analisi automatica di un testo, una interpretazione soddisfacente è costituita da una collezione di concetti e istanze di una base di conoscenza e da una descrizione delle relazioni che li collegano.
- Le entità corrispondenti ai sintagmi nominali devono essere messe in corrispondenza con gli elementi della base di conoscenza.

## Annotazione pragmatica (3)

- Se ad una occorrenza di una entità deve corrispondere l'introduzione di un nuovo elemento si parla di ``*riferimento non anaforico*``, mentre nel caso in cui l'occorrenza si riferisca ad un oggetto già introdotto in precedenza si parla di ``*riferimento anaforico*``

## Esempi (1)

- Ho visto Mario. *Gli* ho detto che partiremo domani.  
(Il sintagma nominale ``Mario`` costituisce un riferimento non anaforico, mentre ``gli`` introduce un riferimento anaforico.)
- Il riferimento a elementi di insiemi precedentemente introdotti richiede quindi particolare accortezza:
- Nel giardino ci sono molti alberi. *Il più vecchio* è una quercia.  
(Casi come questo richiedono che il sistema sia in grado di trovare il corretto antecedente all'interno di una rappresentazione strutturata del testo)

## Esempi (2)

Indizi utili *per trovare il riferimento* corretto sono forniti dal tipo semantico, dagli aggettivi dimostrativi e dai pronomi che compaiono nel sintagma nominale, oltre che dalla presenza di un nome proprio, ma non è possibile trovare regole semplici e affidabili.

Il problema è complicato dal presentarsi di una gran varietà di fenomeni linguistici.



## PARTE QUARTA

### ESEMPIO DI ANNOTAZIONE XML

### Un esempio di annotazione XML: Obiettivo

Creare sistema che lavora su materiale letterario con strutture di recupero di informazioni, adatto a recuperare sia il contenuto di un testo sia il contesto sociale, culturale e storico.

Capace inoltre di analizzare e capire le complesse domande che un utente può formulare sul dominio del testo

### Un esempio di annotazione XML: Cosa esaminiamo?

- Testo in esame: “*La Divina Commedia*”, organizzato usando il meta-linguaggio XML
- Consideriamo i personaggi dell’ *al di là* di Dante, che prevede un dominio consistente di dati e relazioni complesse

### Un esempio di annotazione XML: Cosa deve fare? (1)

Costruire attraverso un *mark-up XML* un **meta-testo**, con informazioni semi-strutturate, riguardanti i personaggi dell’*al di là* (cioè tutte le persone, i luoghi e le istituzioni, anche quelle soltanto nominate)

## Un esempio di annotazione XML: Cosa deve fare? (2)

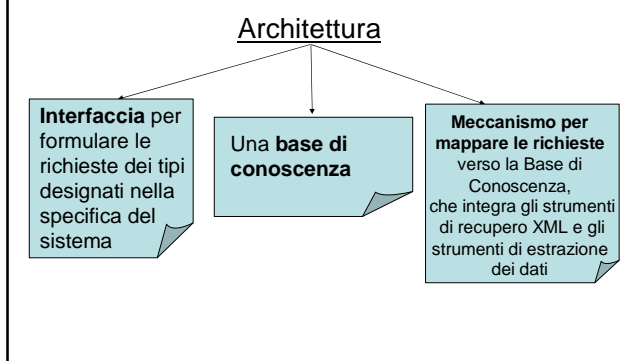
- Il sistema deve essere capace di rispondere a domande di questo tipo:
  - Quanti e quali sono i ghibellini e quanti e quali i guelfi presenti nella Cantica Inferno della Divina Commedia?
  - Quante e quali figure della mitologia classica vengono evocate nell'Inferno?
  - Classificare l'atteggiamento di Dante verso un personaggio in base alle caratteristiche del personaggio.

## Un esempio di annotazione XML: Come funziona?

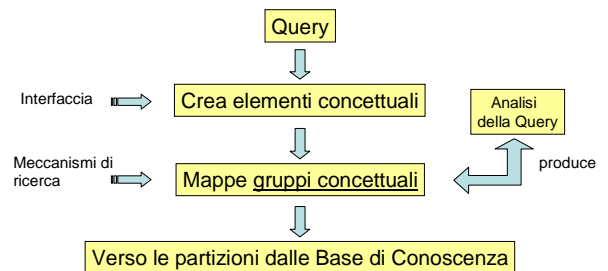
- Elabora un testo marcato

Etichettatura tramite LdM, costituisce un buon mezzo per rappresentare informazioni testuali, con le loro relazioni in strutture complicate di dati progressivamente e dinamicamente aggiornati ("updating"). Manipolabili tramite computer

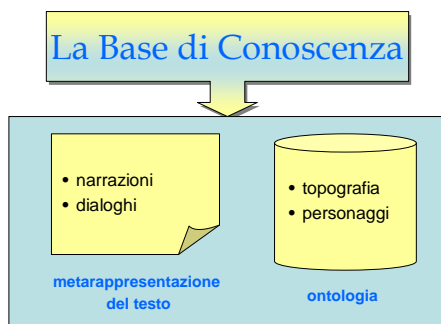
## Un esempio di annotazione XML: Architettura del sistema?



## Un esempio di annotazione XML: Ruolo dell'Interfaccia



## Un esempio di annotazione XML: Base di Conoscenza



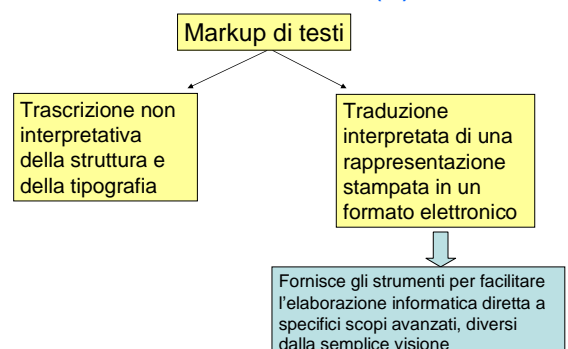
## Un esempio di annotazione XML: Ontologia

- L'ontologia rappresenta le relazioni tra tutti i concetti che descrivono la conoscenza del dominio
- Ogni oggetto è localmente descritto da un insieme di caratteristiche in accordo con il DTD generale
- Ogni oggetto è parte di una *gerarchia*

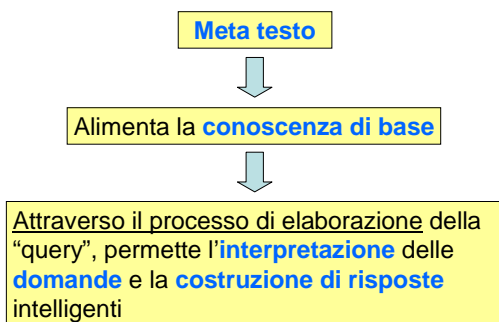
## Un esempio di annotazione XML: Ontologia (2)

<pre> &lt;Table&gt; &lt;Personaggio&gt; &lt;Personaggio&gt; &lt;Nome&gt; Manente &lt;/Nome&gt; &lt;Cognome&gt; degli Uberti &lt;/Cognome&gt; &lt;NotaCome&gt; Farinata &lt;/NotaCome&gt; &lt;Tipo&gt; Storico &lt;/Tipo&gt; &lt;Epoca&gt; Medioevo &lt;/Epoca&gt; &lt;Categoria&gt; Politico &lt;/Categoria&gt; &lt;Sottocategoria&gt; Capo ghibellino &lt;/Sottocategoria&gt; &lt;PosizionePolitica&gt; Ghibellino &lt;/PosizionePolitica&gt; &lt;NoteBiografiche&gt; &lt;Episodio&gt; &lt;Data&gt; 1239 &lt;/Data&gt; &lt;Avvenimento&gt; diventa capo del partito ghibellino &lt;/Avvenimento&gt; &lt;Luogo&gt; Firenze &lt;/Luogo&gt; &lt;/Episodio&gt; ..... &lt;/Personaggio&gt; </pre>	<pre> &lt;Personaggio&gt; &lt;Nome&gt; Ulisse &lt;/Nome&gt; &lt;NotaCome&gt; Ulisse &lt;/NotaCome&gt; &lt;Tipo&gt; Letterario &lt;/Tipo&gt; &lt;AmbitoAppartenenza&gt; Mitologia classica &lt;/AmbitoAppartenenza&gt; &lt;Categoria&gt; Eroe greco &lt;/Categoria&gt; &lt;Caratteristica&gt; Re di Itaca &lt;/Caratteristica&gt; &lt;/Personaggio&gt;  &lt;Personaggio&gt; &lt;NotaCome&gt; Lanza &lt;/NotaCome&gt; &lt;Tipo&gt; Creatura &lt;/Tipo&gt; &lt;Ruolo&gt; ostacolare l'ascesa di Dante al colle della salvezza &lt;/Ruolo&gt; &lt;SignificatoAllegorico&gt; lussuria &lt;/SignificatoAllegorico&gt; &lt;/Personaggio&gt; ..... </pre>
---	---

## Un esempio di annotazione XML: Obiettivo finale (1)



## Un esempio di annotazione XML: Obiettivo finale (2)



## Riferimenti-1

- ✓ [www.w3c.it/MarkUp](http://www.w3c.it/MarkUp)
- ✓ [www.hermesnet.it](http://www.hermesnet.it)
- ✓ [www.wikipedia.org/wiki.cgi.xml](http://www.wikipedia.org/wiki.cgi.xml)
- ✓ [M. Missikoff, F. Schiappelli, F. Taglino A Controlled Language for Semantic Annotation and Interoperability in e-Business Application IASI-CNR \(Italy\) 2003](#)
- ✓ [www.semanticweb.org](http://www.semanticweb.org)
- ✓ [Holzner, Steven. \*Xml, tutto e oltre\*. APOGEO srl, 2001.](#)
- ✓ [Deitel, Harvey M.; Deitel, Paul J.; Nieto, Tem R.; Lin, Ted M.; Sadhu, Praveen. \*Xml, corso di programmazione\*. APOGEO srl, 2002.](#)

## Riferimenti-2

- ✓ Jim Hendler e Deborah L. McGuinness, The DARPA Agent Markup Language, *IEEE Intelligent Systems* 16(6): 67-73. 2000.  
<http://www.ksl.stanford.edu/people/dlm/papers/ieee-daml01-abstract.html><http://www.daml.org/about.html>
- ✓ [Knowledge mining and discovery for searching in literary texts. \(A. Cappelli, M. N. Catarsi, M. Baglioni, P. Michelassi, L. Moretti, M. Tavoni, F. Turini\)](#)