

Micro-dispensa

Versione 2.0 (aggiornamento: 16/03/2013)

Laboratorio di Informatica di base per Lettere Classiche, A. A. 2012/13, 2. semestre, Dr. Paolo Monella, lezione del 15/03/2013.

Modellizzazione

Processo consistente nel creare un modello (una rappresentazione articolata) di una parte della realtà (di un fenomeno etc.).

Ad esempio: la *Divina Commedia* è un poema. Un'edizione digitale della *Divina Commedia* è un modello digitale di quel poema, in quanto lo rappresenta ed è fatto di parti che interagiscono (le lettere che costituiscono il verso, la divisione tra titoli e versi, informazioni riguardanti l'autore etc.).

Formalizzazione

Il concetto è complesso. Passerò direttamente agli esempi.

Io posso *formalizzare* (cioè esprimere con un linguaggio formale) l'espressione (appartenente ad un linguaggio naturale, l'italiano) "L'età di Paolo è la metà dell'età di Giovanni" con l'espressione (appartenente ad un linguaggio formale, quello algebrico):

$$E_p = E_g / 2$$

È chiaro che un computer può gestire quest'ultima espressione, ma non c'è quasi speranza che possa gestire la prima.

O posso *formalizzare* l'espressione appartenente ad un linguaggio naturale "L'acqua è composta da un atomo di ossigeno e da due atomi di idrogeno" con l'espressione (appartenente ad un linguaggio formale, quello della chimica):



Oppure l'espressione "Se unisco un atomo di ossigeno e due atomi d'idrogeno, ottengo una molecola d'acqua, e viceversa" con l'espressione:



Dato che il computer può gestire solo linguaggi formali, prima di dargli in pasto alcunché (inclusa la *Divina Commedia*), bisogna esprimere persino la *Divina Commedia* con un linguaggio formale. Questo processo si chiama 'formalizzazione', e richiede competenze sia informatiche sia umanistiche (perché bisogna conoscere bene una realtà, per riuscire a elaborarne una rappresentazione formale).

Markup

```
Ciao a *tutti*
```

XML (che prima si chiamava SGML)

eXtensible

```
<nome>Paolo</nome>,  
<cognome>Monella</cognome>
```

Linguaggio dichiarativo

Qualche linguaggio XML-based (o non XML-based)

XML-based: HTML (che è un linguaggio procedurale).

Non XML-based: Mentre invece LaTeX, RTF e il formato di Word sono sempre Markup Languages (ML), cioè linguaggi di markup, ma *non* sono fondati su XML (non hanno le parentesi uncinete, le stesse barre etc.).

Tag

Tag di apertura:

```
<nome>
```

Tag di chiusura

```
</nome>
```

Un paio di ‘elementi’ XML (fatti da tag di apertura, contenuto e tag di chiusura). Si noti che “di” non fa parte di nessuno dei due elementi:

```
<title>La Divina Commedia</title>  
di <author>Dante Alighieri</author>
```

DTD (poi ‘Schema’)

Vocabolario di tag XML

XML + DTD = linguaggio di markup (ML)

TEI

Text Encoding Initiative: inventa un vocabolario di tag (una DTD) XML per codificare testi (letterari).

Un esempio di codifica XML/TEI

```
<div type="poem" n="42">  
<head>Tanto gentile e tanto onesta pare</head>  
<lg>  
  <l n="1">Tanto gentile e tanto onesta pare</l>  
  <l n="2">la donna mia quand'ella altrui saluta,</l>  
  <l>ch'ogne lingua deven tremando muta, </l>  
  <l>e gli occhi no l'ardiscon di guardare.</l>  
</lg>  
<lg>  
  <l>Ella si va, sentendosi laudare, </l>  
  <l>benignamente d'umiltà vestuta; </l>  
  <l>e par che sia una cosa venuta </l>  
  <l>da cielo in terra a miracol mostrare. </l>
```

```
</lg>  
</div>
```

N.B.: In realtà, come leggete sopra, il modo giusto per codificare inizio e fine di una poesia è tramite il tag <div> (=divisione del testo, sezione del testo), cui si attribuisce l'attributo @type (tipo di sezione del testo: qui 'poem').