Il markup di documenti

Fabio Vitali

3 dicembre 1999

«Quando io uso una parola», disse Humpty Dumpty in tono piuttosto sprezzante, «significa quello che io scelgo che significhi - né più, né meno.»

Lewis Carroll

“Attraverso lo specchio”

Introduzione

Oggi esaminiamo:

- Cos’è il markup
- Che tipi di markup esistono
- Caratteristiche di markup procedurale e dichiarativo
- Una brevissima storia del markup su computer

Perché tutto questo chiasso?

- Quando si porta una collezione di documenti in forma elettronica, si ha di solito in mente in generale una specifica applicazione (metterla in rete, prepararla per la stampa, ecc.).
- Di solito, si cerca di trasformare il documento nella forma più opportuna perché venga utilizzato nell'applicazione suddetta.
- Spesso per questo si fanno delle scelte che impediscono o ostacolano notevolmente un ulteriore riuso della stessa collezione per una diversa applicazione. L'impaginato poco si adatta ad un'indicizzazione per la rete, o viceversa i linguaggi di visualizzazione su rete sono troppo poco sofisticati per una produzione tipografica di buon livello, ecc.
- I linguaggi di markup derivati da SGML sono i linguaggi più opportuni per strutturare e marcare i documenti in maniera indipendente dall'applicazione, favorendo la ***riusabilità***, la ***flessibilità*** e la ***apertura ad applicazioni complesse***.

Cos’è il markup? (1)

- Definiamo markup ***ogni mezzo per rendere esplicita una particolare interpretazione di un testo***.
- Per esempio, tutte quelle aggiunte al testo scritto che permettono di renderlo più fruibile.
- Oltre a rendere il testo più leggibile, il markup permette anche di specificare ulteriori usi del testo.
- Con il markup per sistemi informatici (il nostro caso), specifichiamo le modalità esatte di utilizzo del testo nel sistema stesso.
- Il markup non è soltanto un’inevitabile e sgradevole risultato della informatizzazione dell’arte tipografica. Non è qualcosa che sta con noi a causa dell’informatica.

Cos’è il markup? (2)

- Quando un autore scrive, da millenni a questa parte, specifica anche i delimitatori di parola (chiamati ***spazi***), i delimitatori di frase (chiamati virgole) e i delimitatori di periodo (chiamati punti).
- La numerazione delle pagine o l’uso dei margini per creare effetti sul contenuto sono noti da centinaia di anni.
- Eppure questo a stretto rigore non fa parte del testo, ma del markup: nessuno dirà ad alta voce ‘virgola’ o ‘punto’ nel leggere un testo, ma creerà adeguati comportamenti paralinguistici (espressioni, toni, pause) per migliorare in chi ascolta la comprensione del testo.

Tipi di markup

Il markup assolve a diversi ruoli a seconda del sistema di elaborazione, dell'applicazione, dello scopo a cui il documento è soggetto.

- Puntuazionale
- Presentazionale
- Procedurale
- Descrittivo
- Referenziale
- Metamarkup

Markup puntuazionale

- Il markup puntuazionale consiste nell'usare un insieme prefissato di segni per fornire informazioni perlopiù sintattiche sul testo.
- Le regole di punteggiatura sono sostanzialmente stabili, note agli autori, e frequenti nei documenti. Per questo gli autori tipicamente forniscono il loro markup puntuazionale autonomamente.
- Esistono tuttavia notevoli problemi nell'uso della punteggiatura:
  - Incertezze strutturali (virgola, punto e virgola o punto?),
  - Incertezze grafiche (virgolette aperte e chiuse o neutre?),
  - ambiguità procedurali (il punto viene usato sia per segnare la fine di una frase, che l'esistenza di un'abbreviazione, senza contare i tre puntini di sospensione).

Markup presentazionale

- Il markup presentazionale consiste nell'indicare effetti (grafici o altro) per rendere più chiara la presentazione del contenuto.
- Nel testo, possono essere cambi di paragrafo o di pagina, interlinea, pallini per liste, ecc.
- E' altresì markup presentazionale: cambiare pagina all'inizio di una nuova sezione, scrivere "Capitolo 3" in cima alla pagina, ecc.

Markup procedurale

- Il markup procedurale ponsiste nell'indicare con precisione ad un sistema automatico che effetto attivare e che procedura (serie di istruzioni) eseguire nella visualizzazione del contenuto.
- In definitiva, utilizzo le capacità del sistema di presentazione per avere con precisione l'effetto voluto.
- Esempio: ***Wordstar Dot Commands***
  .PL 66
  .MT 6
  .MB 9
  .LH 12

Markup descrittivo

- Il markup descrittivo consiste nell'identificare strutturalmente il tipo di ogni elemento del contenuto.
- Invece di specificare effetti grafici come l'allineamento o l'interlinea, ne individuo il ruolo all'interno del documento, specificando che un elemento è un titolo, un paragrafo, o una citazione.

Markup referenziale

- Il markup referenziale consiste nel fare riferimento ad entità esterne al documento per fornire significato o effetto grafico ad elementi del documento. Per esempio, utilizzare una sigla nota che venga poi sostituita dalla parola intera durante la stampa
- Es.: l'autore scrive "CdL" e il sistema trasforma automaticamente l'input in "Corso di Laurea"

Metamarkup

- Il metamarkup consiste nel fornire regole di interpretazione del markup e permette di estendere o controllare il significato del markup.
- Ad esempio, la possibilità di definire macro per l'interpretazione e la visualizzazione del documento, o il processo di definizione degli elementi e delle procedure valide di un documento.

Un testo su carta

Tre Uomini in Barca

Jerome K. Jerome

1889

Capitolo primo

Tre invalidi - Le sofferenze di George e Harris - La vittima di centosette malattie inguaribili - […]

Eravamo in quattro: George, William Samuel Harris, e io, Montmorency. Standocene seduti in camera mia, fumavamo e parlavamo di quanto fossimo malridotti… malridotti, dal punto di vista della salute, intendo, naturalmente.
Ci sentivamo tutti piuttosto giù di corda, ...

3

Il testo senza markup

Questo è il testo completamente senza markup, come poteva essere scritto su un papiro della biblioteca di Alessandria, nel II o III secolo a.C.

Treuominiinbarcajeromekjerome1889capitoloprimotreinvalidilesofferenzedigeorgeeharrislavittimadicentosettemalattieinguaribilieravamoinquattrogeorge,williamsamuelharriseiomontmorencystandocenesedutiincameramiafumavamoeparlavamodiquantofossimomalridottimalridottidalpuntodivistadellasaluteintendonaturalmentecisentivamotuttipiuttostogiùdicorda

Markup metabolizzato

Aggiungiamo markup puntuazionale e presentazionale: maiuscole/minuscole, punteggiatura, spazi e ritorni a capo sono essi stessi elementi di markup.

Tre Uomini in Barca
Jerome K. Jerome (1889)
Capitolo primo
Tre invalidi - Le sofferenze di George e Harris - La vittima di centosette malattie inguaribili - […]
Eravamo in quattro: George, William Samuel Harris, e io, Montmorency. Standocene seduti in camera mia, fumavamo e parlavamo di quanto fossimo malridotti… malridotti, dal punto di vista della salute, intendo, naturalmente.
Ci sentivamo tutti piuttosto giù di corda, ...

Markup procedurale

Sono comandi, o istruzioni che il sistema di lettura (umano o elettronico) deve eseguire sul testo. Ad esempio, istruzioni su come andare a capo, come decidere i margini, ecc. Questo è RTF.

{\rtf1 \mac \ansicpg10000 \uc1 \pard \plain \s15 \qc \widctlpar \adjustright \f4 \fs48 \cgrid {**Tre Uomini in Barca** \par } \pard \plain \widctlpar \adjustright \f4 \cgrid { \line \par \par \par \par \par } \pard \plain \s1 \qc \keepn \widctlpar \outlinelevel0 \adjustright \i \f4 \fs36 \cgrid {**Jerome K. Jerome** \par } \pard \plain \qc \widctlpar \adjustright \f4 \cgrid { \fs36 […] \par **1889** \line \par } \pard \widctlpar \adjustright { \page } { \b \fs36 **Capitolo primo**} { \par \par \par \line } { \i **Tre invalidi - Le sofferenze di George e Harris - La vittima di centosette malattie inguaribili - [** \u8230 \'c9**]** \par } { \line } { \fs28 **Eravamo in quattro: George, William Samuel Harris, e io, Montmorency. Standocene seduti in camera mia, fumavamo e parlavamo di quanto fossimo malridotti** \u8230 \'c9 **malridotti, dal punto di vista della salute, intendo, naturalmente. \line Ci sentivamo tutti piuttosto gi** \u249 \'9d **di corda, ...** \par }}

Markup descrittivo

Sono informazioni (descrizioni) sugli elementi del documenti, che ne specificano il ruolo, la giustificazione, la relazione con gli altri elementi.

```
<ROMANZO><TITOLO>Tre Uomini in Barca</TITOLO>
<AUTORE>Jerome K. Jerome</AUTORE>
<ANNO>1889</ANNO>
<CAPITOLO><TITOLO>Capitolo primo</TITOLO>
<INDICE><EL>Tre invalidi</EL><EL>Le sofferenze di George e Harris
</EL><EL> La vittima di centosette malattie inguaribili </EL>[…]</INDICE>
<PARA>Eravamo in quattro: George, William Samuel Harris, e io,
Montmorency. Standocene seduti in camera mia, fumavamo e parlavamo di
quanto fossimo malridotti… malridotti, dal punto di vista della salute, intendo,
naturalmente. </PARA>
<PARA>Ci sentivamo tutti piuttosto giù di corda, …</PARA>
</CAPITOLO>… </ROMANZO>
```


Il markup procedurale (1)

- Basato sull'aspetto
  - Ad ogni elemento del documento viene associata la procedura per visualizzarlo in maniera voluta: font, dimensione, corsivi, grassetti, margini, interlinea, ecc.
- Dipendente dal sistema
  - ogni sistema di visualizzazione impone le proprie regole e la propria sintassi, dipendendo da:
    - Filosofia sintattica (comandi-punto per troff, comandi-barra per RTF, ecc.)
    - Capacità di raggruppamento (parentesi graffe in RTF, comando di disattivazione esplicita in troff, ecc.)
    - Supporto di specifiche funzionalità

Il markup procedurale (2)

- Associato agli individui
  - ogni elemento possiede le proprie procedure per la visualizzazione, che possono anche essere tutte diverse anche per elementi dello stesso tipo.
- Non contestuale
  - Le regole di visualizzazione non dipendono dal contesto in cui vengono fatte, ma ognuna fa specie a sé.
    - Ad esempio, una lista in **`troff`** è fatta come segue:

```
.li
        .it elemento 1
        .it elemento 2
.el
```

    - Nessun controllo impone di chiudere la lista alla fine, o di usare i comandi .it solo dentro alla lista.

Il markup dichiarativo (1)

Basato sul ruolo

- Di ogni elemento viene descritto il ruolo all'interno del testo, più che le regole per la sua visualizzazione:
- Ad esempio: "questo è un titolo, questo è un paragrafo, questo è il nome dell'autore, questa è una citazione."

Indipendente dal sistema

- Poiché il markup dichiarativo assegna ruoli (e non regole di visualizzazione) agli elementi del testo, questi sono intrinseci agli elementi stessi, e non alle funzionalità disponibili nel sistema di visualizzazione.
- Un sistema incapace di variare l'interlinea, o con un elenco limitato di font e dimensioni, può aver problemi ad interpretare un markup procedurale troppo ricco, ma la differenza tra (per esempio) un "titolo" o un "elenco" o un "paragrafo" non dipende dalla sofisticazione del sistema di visualizzazione.

Il markup dichiarativo (2)

Basato su categorie

- I ruoli sono categorie. Ogni elemento è associato ad una categoria, e ne riflette tutte le caratteristiche automaticamente.

Contestuale

- Con il markup dichiarativo è possibile definire delle regole che permettano o impediscano l'assegnazione di una categoria (ruolo) ad un elemento del testo a seconda del contesto.
- Ad esempio, si può richiedere che il titolo vada all'inizio del testo, o che una lista sia composta solo di elementi della lista, e non da paragrafi, ecc.

Il markup dichiarativo (3)

Il markup generico sfrutta il ***late binding***: le scelte specifiche si fanno all'ultimo momento utile.

Nel nostro caso, la formattazione viene decisa nel momento in cui il documento viene visualizzato, o stampato, piuttosto che quando il documento viene codificato.

Il markup dichiarativo (4)

Ci sono vari vantaggi in questa tecnica:

- **Facilità nella creazione**: l'autore si concentra sul ruolo organizzativo delle singole parti di testo, piuttosto che sul loro aspetto stampato.
- **Indipendenza dalla formattazione**: riformattare un documento secondo nuove regole richiede semplicemente di ricodificare dei parametri esterni, non di modificare in alcuna maniera il documento
- **Flessibilità**: riusare un documento in un nuovo contesto e' facile, perché non è necessario rimuovere la vecchia informazione per far posto alla nuova.
- **Visioni di documenti dinamicamente riconfigurabili**: è possibile evidenziare di volta in volta caratteristiche del documento diverse (caratteristica degli outline processor, per esempio)

Modi del markup (1)

Il markup può dunque essere:

- visibile o nascosto
- espresso tramite simboli speciali o tramite testo leggibile
- inserito all'inizio del testo, embedded nel testo o contenuto in un oggetto a parte.
  Se è embedded, esisteranno dei delimitatori per distinguere tra le istruzioni di markup ed il contenuto del documento vero e proprio.

Modi del markup (2)

Tradizionalmente, editor (sistemi di inserimento dati) e formatter (sistemi di creazione di output) erano separati. LaTeX ancora riflette questa distinzione, MS Word no.

Il markup viene utilizzato da un formatter per la creazione dell'output, ma l'editor può operare sul markup se ne conosce le caratteristiche. In questo caso il markup sarà:

- **Esposto**: il sistema mostra il markup e ne mostra gli effetti
- **Travestito**: il sistema mostra nel contenuto un simbolo che attiva il markup
- **Nascosto**: il sistema nasconde il markup e ne mostra solo gli effetti
- **Visualizzato**: il sistema non interpreta in markup e lo mostra insieme al contenuto.

Uso del markup (su computer)

- Possiamo sistemare, riorganizzare e formattare un testo per la stampa.
- Possiamo sistemare, riorganizzare e formattare un testo per la lettura a video.
- Possiamo sistemare, riorganizzare e formattare un testo per l'uso di handicappati fisici (ad esempio ciechi con l'aiuto di un lettore Braille)
- Possiamo sistemare, riorganizzare e formattare un testo per la lettura del testo ad alta voce in situazioni di impedimento temporaneo (ad esempio, mentre stiamo guidando).
- Possiamo permettere facilmente una ricerca sui contenuti del testo.
- Possiamo verificare la adeguatezza del testo rispetto a regole più o meno formali di strutturazione.

La storia del markup (1)

Anni ‘60: primo WP in IBM

- Uomini e animali sono già nello spazio grazie ai computer, le aziende telefoniche ed elettriche già calcolano le bollette con i computer, eppure prima del 1964 nessuno aveva ancora trattato i testi con i computer.

GCA-GenCode e IBM-GML (1968-70)

- GenCode è il risultato della standardizzazione dei codici di tipografia (***Graphics Communications Association***)
- ***Generalized Markup Language*** di IBM è il linguaggio di markup per la documentazione interna e il prodotto BookMaster.

La storia del markup (2)

Anni ‘80: WP WYSIWYG e DTP: un passo indietro?

- Si diffondono i primi WP: WordStar, Word Perfect, MS Word
- Nasce il concetto di WYSIWYG (***What You See Is What You Get***): MacWrite e poi MS Word, ecc. L’enfasi è sulla verosimiglianza tra ciò che si vede sullo schermo e ciò che risulta sulla carta.
- Nascono i primi prodotti da editoria professionale (DTP: ***Desk Top Publishing***): PageMaker, Ventura, Quark Xpress, ecc.

1986: SGML è standard ISO 8879

- Giudizi misti: “***Sounds Great, Maybe Later***”: manca il software, è considerato complicato e sofisticato.
- Nel 1988 il dip. della difesa americano (DoD) adotta SGML per l’iniziativa CALS (***Continuous Aided Logistic Support***)
- 1990: TEI (***Text Encoding Initiative***): un gruppo di umanisti generano linee guida per la strutturazione dei testi umanistici (prosa, teatro, poesia, ecc.).

La storia del markup (3)

1991: HTML

- Tim Berners Lee (CERN - Ginevra) inventa un sistema ipertestuale per la rete Internet chiamato “World Wide Web”. Il WWW è basato su tre protocolli, URI, HTTP e HTML
- HTML non nasce come un’applicazione SGML, ma solo come “ispirato” ad SGML. Solo in seguito verrà corretto per adeguarsi a SGML.

1997: la convergenza su XML

- Lo sviluppo e la correttezza degli standard connessi con il WWW sono gestiti dal W3C (World Wide Web Consortium).
- Nel 1995 la commissione sui linguaggi di markup decise di creare un nuovo linguaggio di markup con la completezza di SGML e la semplicità di HTML: ***Extended Markup Language*** (XML).
- Nel 1997 è uscito il primo standard per il linguaggio di markup (XML 1.0). In seguito i linguaggi connessi (XML-Naming, X-Pointer, X-Link, XSL, ecc.).

SGML

- SGML (Standard Generalized Markup Language) è uno standard.
- SGML è un meta-linguaggio non proprietario di markup dichiarativo.
- Facilita markup leggibili, generici, strutturali, gerarchici.

Linguaggio standard

SGML è uno standard ISO (International Standard Organization) n. 8879 del 1986. ISO è l'organizzazione mondiale degli standard, la più importante.

Essere uno standard per SGML significa che esso è il risultato di discussioni e compromessi di rappresentanti di tutte le comunità interessate, che è un investimento nel tempo di queste comunità, e che non dipende dai piani commerciali o dai capricci di una singola casa produttrice.

Meta-linguaggio di markup

Un meta-linguaggio è un linguaggio per definire linguaggi, una grammatica di costruzione di linguaggi.

SGML non è un linguaggio di markup, ma un linguaggio con cui definiamo linguaggi di markup.

SGML non dà dunque tutte le risposte di markup che possano sorgere a chi vuole arricchire testi per qualunque motivo, 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.

Linguaggio non proprietario

Non proprietario significa che non esiste un’unica ditta o casa produttrice che ne detiene il controllo.

Viceversa RTF è © Microsoft, mentre PostScript e Acrobat sono © Adobe.

Essendo standard non proprietario, non dipende da un singolo programma, da una singola architettura. Gli stessi dati possono essere portate da un programma all’altro, da una piattaforma all’altra senza perdita di informazione.

Inoltre, anche l’evoluzione nel tempo è assicurata per lo stesso motivo.

Markup leggibile

In SGML il markup è posto in maniera leggibile a fianco degli elementi del testo a cui si riferiscono.

Essi possono sia essere usati da un programma, sia letti da un essere umano. Possono sia essere aggiunti da un programma (editor), sia da un essere umano con un programma non specifico.

Il markup in SGML è semplice testo facilmente interpretabile.

Markup dichiarativo

Il markup in SGML non è pensato unicamente per la stampa su carta. E’ possibile combinare markup utile per scopi o applicazioni diverse, ed in ogni contesto considerare o ignorare di volta in volta i markup non rilevanti.

Markup strutturato

SGML permette di definire delle strutture, suggerite o imposte, a cui i documenti si debbono adeguare.

Ad esempio, si può imporre che un testo sia diviso in capitoli, ognuno dei quali dotato di un titolo, una breve descrizione iniziale e almeno un paragrafo di contenuto.

È cioè possibile definire una serie di regole affinché il testo sia considerabile strutturalmente corretto.

Markup gerarchico

Le strutture imposte da SGML sono tipicamente a livelli di dettaglio successivi.

Gli elementi del testo possono comporsi gli uni con gli altri, permettendo di specificare la struttura in maniera gerarchica.

Ad esempio, si può imporre che il libro sia fatto di capitoli, e che questi a loro volta siano fatti di un una descrizione e molti paragrafi. Una descrizione sarà quindi fatta di elementi, ciascuno dei quali sarà una frase composta di testo; i paragrafi saranno testo eventualmente contenenti anche elementi in evidenza o figure.

Conclusioni

Oggi abbiamo parlato di

- Importanza del markup nel testo
- Rilevanza del markup dichiarativo per applicare scopi multipli allo stesso testo
- Qualche distinzione tra tipi di markup
- Un po’ di storia del markup

Riferimenti

Wilde's WWW, capitolo 4.1

Altri testi:

- J. H. Coombs, A. H. Renear, S. J. DeRose, Markup Systems and the future of Scholarly Text Processing, ***Communications of the ACM***, 30(11), November 1987.
- "I.2 A Gentle Introduction to SGML", in C.M. Sperberg-McQueen and L. Burnard (eds.), ***Guidelines for Electronic Text Encoding and Interchange***, 1994, http://etext.virginia.edu/TEI.html
- E. Maler, J. El Andaloussi, ***Developing SGML DTDs***, Prentice Hall, 1996