

Introduzione

Cos'è un documento

Un documento è una collezione di dati di diverso formato.

Tipi di documento:

- testi: romanzi, racconti, poesie, saggi, manuali, articoli, voci di enciclopedia, lettere, ricette ecc.
- non-testi: immagini, registrazioni audio, tracciati musicali, schemi, progetti, fotografie, filmati, animazioni, mondi virtuali, ecc.
- quasi-testi: annunci economici, elenchi telefonici, schede cliniche, bilanci, bollette, programmi di computer

È possibile gestire tali documenti tramite computer, aiutando gli utenti a svolgere i loro compiti meglio, più velocemente, più efficacemente.

Operazioni su un documento

Acquisizione: via scanner, tastiera, rete, penna ottica, ecc.

Modifica: strutturazione, aggiornamento, correzione, montaggio, ecc.

Uso: stampa, trasmissione, conversione, ecc.

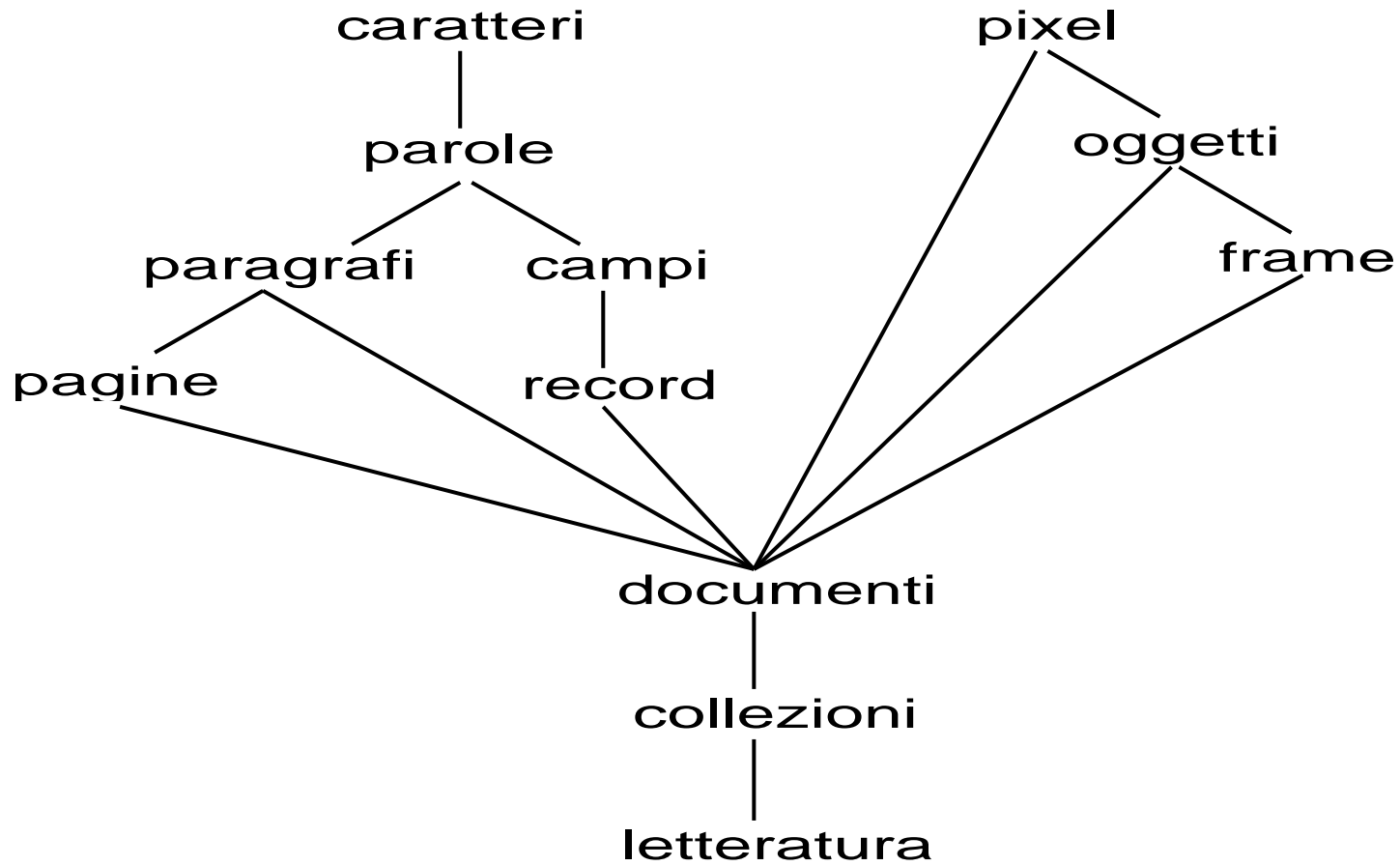
Lettura (ovviamente): su carta, su uno schermo di computer, su uno scanner, su un terminale braille, letto ad alta voce da una persona o da un meccanismo automatico, ecc.

Rielaborazione: selezione, ricerca, ordinamento, ecc.

Elaborazioni: calcolo di statistiche e totali, esecuzione di istruzioni, ecc.

Una visione telescopica dei documenti

Una struttura di dipendenze



Caratteri

Attribuiamo un valore convenzionale a sequenze di bit. Una sequenza di n bit può assumere 2^n valori diversi, quindi può rappresentare 2^n segni.

Il codice EBCDIC utilizza 8 bit ($2^8=256$ valori diversi). Viene utilizzato su mainframe per uso perlopiù gestionale.

Il codice ASCII utilizza 7 bit ($2^7=128$ valori diversi) su 8 che occupa (un gruppo di 8 bit viene chiamato byte). L'ottavo bit viene ignorato o utilizzato come codice di controllo.

Esistono codici ASCII a 8 bit. Ogni venditore di sistemi operativi ha creato una sua estensione dell'ASCII diversa dall'altra per i rimanenti 128 valori. Questo è il motivo per cui esistono problemi di conversione tra file di sistemi diversi (es. Windows e Macintosh).

Il codice ISO Latin 1 è un estensione a 8 bit del codice ASCII. Mentre è molto diffuso negli usi di rete (è indipendente dal venditore), i sistemi operativi non l'hanno ancora adottato diffusamente.

Caratteri in Internet: Unicode

Il codice UNICODE utilizza 16 bit. I possibili valori non sono ancora stati tutti assegnati (34168 su 655236). Il codice ISO Latin 1 corrisponde ai caratteri Unicode i cui primi 8 bit sono tutti zero.

Unicode raddoppia lo spazio occupato da testi scritti in caratteri latini, ma permette la rappresentazione dei caratteri di molti alfabeti non latini: arabo, ebraico, cinese, giapponese, coreano, thailandese, ecc, nonché caratteri speciali (matematici, tecnici, frecce, elementi di disegno (dingbats), ecc).

Dunque i caratteri del tipo *0000 0000 annn nnnn* è un carattere ISO Latin 1. Se *a* è 0, è un carattere ASCII.

Parole

Le parole sono sequenze di caratteri divise da separatori di parola.

Tipicamente come separatori si usano spazi e segni di punteggiatura.

Ma ci sono eccezioni: in molti linguaggi di programmazione, la punteggiatura ha altri scopi.

Conteggio, controllo ortografico e divisione delle parole (*hyphenation*) sono operazioni interessanti sulle parole.

Paragrafi

Un paragrafo è una collezione di parole segnata da separatori di paragrafo. Tipicamente un ritorno a capo è un separatore di questo tipo. Word processor diversi aggiungono concetti intermedi (sezione, colonna, ecc), che richiedono ulteriori separatori.

Sistemi diversi utilizzano separatori diversi per i paragrafi

- DOS e Windows: Carriage Return + Line Feed (ASCII 13 e ASCII 10)
- Mac: Carriage Return (ASCII 13)
- Unix: Line Feed (ASCII 10)

Campi, record e tabelle

Collezioni omogenee di informazioni sono strutturate in una sequenza definita più o meno rigorosamente di *campi*.

Ad esempio, in un elenco telefonico ogni riga è composta (salvo eccezioni) dai campi cognome, nome, indirizzo, numero di telefono.

Ogni riga dell'elenco telefonico è strutturalmente identica alle altre, e viene detta “*record*”. Ogni record è composto da campi di tipo diverso.

Per ogni campo il tipo di dato (stringa o numero), il significato del dato (cognome o numero di telefono) e la posizione (prima il cognome, poi il nome, ecc) sono prefissati.

Una *tabella* è una collezione di record.

I campi, avendo struttura identica su tutti i record di una tabella, sono definiti una volta per tutte in uno “schema” della tabella.

schema

Cognome	Nome	Indirizzo	Telefono
Rossi	Andrea	Via Indipendenza 1	221145
	Beatrice	Via Ugo Bassi 4	337541
	Davide	Via Zamboni 22	657423
	Filippo	Via Mazzini 10	354487
	Marco	Via Irnerio 7	652311
	Pietro	Via Indipendenza 6	487522
Rossini	Davide	Via Marconi 3	356595

campo record

Pagine

Alcuni sistemi di creazione di documenti sono orientati alla stampa di pagine.

Le *pagine* sono record di dimensioni prefissate, ma esistono formati diversi usati nazionalmente o a seconda degli scopi: in Europa esistono i formati A, definiti a partire dal formato A0 (84x118.8 cm) e dimezzando ogni volta il lato più lungo. Il formato più comune è A4 (21 x 29.7 cm). In America usano comunemente altri due formati, detti *lettera* e *legale*.

Altri tipi di pagina diversi da questi possono essere usati per scopi particolari: buste, inviti, biglietti da visita, ecc.

Schermi

Alcuni sistemi di creazione di documenti sono invece orientati alla visualizzazione su schermo.

Gli schermi sono di capacità diverse: dimensioni, risoluzione, numero di colori. Una schermata che risulta piacevole e ben armonizzata su uno schermo può risultare tagliata o ridotta o illeggibile in uno schermo diverso per dimensione, risoluzione o numero di colori.

Documenti

Un documento è un oggetto complesso composto da ben più del suo contenuto.

La sua organizzazione (strutturale o visiva), ed il numero di servizi aggiunti sono caratteristiche importanti.

Sommario della struttura.

Sommario delle figure o delle tabelle.

Indice analitico.

Bibliografia.

Riferimenti incrociati.

Integrazione di grafica ed animazione.

Meccanismi di navigazione (anche nei libri: cavalieri e pagine colorate).

Raccolte

Una raccolta di documenti (un quotidiano, una antologia di brani, un CD ROM, un sito Web, ecc) è caratterizzata da caratteristiche importanti che vanno preservate:

- aspetti strutturali comuni
- uniformità di aspetto
- presenza di collegamenti ed interconnessioni
- uniformità nei meccanismi di accesso
- presenza di servizi aggiuntivi (indici, presentazioni, sistemi di ricerca ecc.)

Letteratura

Intendiamo per letteratura un insieme di raccolte e documenti singoli che formano un universo completo su un determinato argomento (la letteratura medica sull'AIDS, la letteratura critica su Proust, i documenti sulle aperture negli scacchi, ecc).

Formata tipicamente in momenti, modi e forme diverse, per scopi diversi, da persone diverse.

Variamente interconnessa con citazione esplicite e riferimenti impliciti.

Il markup

Introduzione

Quando un autore scrive qualcosa, aggiunge del “markup”, ovvero dei segni esterni al contenuto del documento che indicano “effetti” sul testo.

Questo è vero non solo se impaginiamo o correggiamo bozze, o scriviamo qualcosa in grassetto o lo sottolineiamo, ma anche quando aggiungiamo la punteggiatura o gli spazi tra le parole.

Infatti il markup puntuazionale o quello presentazionale sono talmente compenetrati nel nostro modo di fare e scrivere che non ce ne accorgiamo neanche.

Tipi di markup

Puntuazionale

Il markup *puntuazionale* consiste nell'usare un insieme prefissato di segni per fornire informazioni perlopiù sintattiche sul testo.

Esistono notevoli incertezze, strutturali (quando usare virgole, punti e virgole o punti), o grafiche (se adoperare virgolette aperte e chiuse), ed ambiguità (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).

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.

Ancora: cambiare pagina all'inizio di una nuova sezione, scrivere “Capitolo 3” in cima alla pagina, ecc.

Procedurale

Il markup *procedurale* consiste nell'indicare con precisione ad un sistema automatico che effetto attivare e che procedura (serie di istruzioni) eseguire nella visualizzazione del contenuto.

Invece di creare l'effetto a mano, utilizzo le capacità del sistema di presentazione per ottenere con più precisione l'effetto.

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.

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 "sdc" e il sistema trasforma automaticamente l'input in "Scienze della Comunicazione"

Metamarkup

Il *metamarkup* consiste nel fornire regole di interpretazione del markup e permette di estendere o controllare il significato del markup. Per esempio

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.

Usi del markup

- Lettura
- Formattazione e stampa
- Ri-elaborazione
- Varie applicazioni

Modi del markup

- Esposto
- Travestito
- Nascosto
- Visualizzato

Altri criteri di valutazione

- Manutenibilità
- Portabilità
- Flessibilità
- Ri-uso

Sistemi di Text Markup

Introduzione: Il ruolo del markup

Per markup si intendono le informazioni aggiunte ad un testo preparato elettronicamente per definirne la struttura o il formato in cui esso appare.

Può 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.

La distinzione più importante è tra markup specifici e generici

Markup specifico

Il markup *specifico* dipende fortemente dal sistema che ha generato il documento, e definisce un linguaggio che è specifico del programma usato.

In generale ha un effetto diretto sul modo in cui il contenuto viene reso (visualizzato o stampato), sia per l'aspetto dei caratteri (scelta dei font e delle dimensioni, grassetto, corsivo, etc.) che per il loro posizionamento (interlinea, margini, giustificazione, etc.)

Può essere isolato su una linea, e richiedere quindi solo un delimitatore d'inizio (il fine linea è il delimitatore finale) oppure può essere inserito nel testo, e richiedere sia il delimitatore iniziale che quello finale.

- Esemplio: Wordstar Dot Commands

```
.PL 66  
.MT 6  
.MB 9  
.LH 12  
.UJ ON  
^A^BCapitolo 1  
INTRODUZIONE^B  
Per markup si intende...
```

- Esemplio: Word processor PeachText:

```
\PL66, TM6, BM9, SP+1, JUSTC, CPI12, BF@\ @Capitolo 1  
INTRODUZIONE@  
Per markup si intende...
```

Markup generico

(o descrittivo o di contenuto)

È indipendente dal sistema che genera i documenti, e descrive lo scopo del testo, piuttosto che il suo aspetto. Esso precisa la struttura di base del documento indicando (mark up) l'inizio e la fine delle varie componenti del testo.

Piuttosto che concetti come il font, il grassetto, etc., il markup generalizzato identifica elementi come l'intestazione, il titolo, le citazioni, etc.

Sarà poi il programma destinato al rendering, sia esso un driver di stampa o un programma di visualizzazione su schermo, a decidere quali caratteristiche visuali associare ad ogni elemento strutturale del testo.

Due sono quindi le premesse fondamentali:

- Il markup deve descrivere la struttura di un documento piuttosto che le sue caratteristiche fisiche
- Il markup dovrebbe essere così rigoroso da permettere una interpretazione non ambigua sia per un lettore umano che per un interprete software.

Esempio: DCF GML (Goldfarb, 1970)

```
:book.  
:body:  
:h1.INTRODUZIONE  
:p.Per markup si intende...
```

Esempio: Ventura Publisher

```
@CHAPTERHEAD=Capitolo 1<R>INTRODUZIONE  
Per markup si intende...
```

Il markup generico sfrutta il *late binding*: la scelta si fa 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.

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 è facile, perché non è necessario rimuovere la vecchia informazione per far posto alla nuova.
- **Visioni di documenti dinamicamente riconfigurabili,** ovvero la possibilità di evidenziare di volta in volta caratteristiche del documento diverse (caratteristica degli outline processor, per esempio)

SGML: Introduzione

SGML (Standard Generalized Markup Language) è uno standard ISO (n. 8879) pubblicato nel 1986 e lievemente modificato nel 1988. È un meccanismo flessibile e portabile per rappresentare documenti elettronici.

SGML è stato adottato da istituzioni importanti per la codifica dei loro documenti interni ed esterni. Tra essi, le maggiori società produttrici di aerei o automobili, e il DoD (dipartimento della difesa americano).

Esso deriva dalle esperienze di Goldfarb con GML (primi anni 70), e Reid con il sistema Scribe (Carnegie-Mellon, fine anni 70). Essi crearono un comitato ISO per la standardizzazione delle caratteristiche strutturali dei testi, che produsse il linguaggio SGML.

Inizialmente disponibile solo su computer molto potenti (mainframe), adesso viene messo a disposizione su moltissime architetture. Gli strumenti SGML principali sono analizzatori sintattici, editor WYSIWYG, sistemi di document management e retrieval.

Caratteristiche

- **Descriptive Markup:** Un documento SGML consiste in oggetti di varie classi (capitoli, titoli, riferimenti, oggetti grafici, etc.), non sequenze di istruzioni di formattazione. Questi oggetti sono chiamati elementi, ed appartengono ad un determinato tipo di elementi. SGML identifica gli estremi degli elementi tramite tags iniziali e finali. :

`<ref>Vedi anche Rossi 1990</ref>`

- **Struttura gerarchica:** Gli elementi si possono contenere reciprocamente in una gerarchia: un capitolo contiene un titolo ed una o più sezioni che a loro volta contengono altri elementi, etc.).

Si è dimostrato più volte che gli alberi permettono strutture notevolmente più complesse che non le liste semplici.

- **Flessibilità:** SGML non impone il tipo di componenti che debbono apparire in un documento, né il modo in cui essi sono posti in relazione.

Permette invece di dichiarare i tipi e le loro relazioni, e permette di verificare la correttezza dei documenti rispetto a queste specificazioni.

- **Specifiche formali:** I tipi definiti in uno o più documenti SGML sono definiti formalmente prima dell'inizio del documento, in un DTD (Document Type Declaration).

Questo permette al parser di affermare la correttezza formale del documento in corso di lettura

- **Rappresentazione leggibile:** SGML utilizza un formato di testo semplice per contenere tutte le informazioni da esso codificate.

Questo permette di scrivere documenti SGML non soltanto con sofisticati tool SGML, ma anche con semplici text-editor.

I documenti SGML

Un documento SGML si divide in tre parti:

La dichiarazione SGML

Una dichiarazione SGML fornisce informazioni generali sulle diverse opzioni sintattiche di SGML. In particolare, permette di specificare la lunghezza dei nomi, il set di caratteri usato, le strinche di delimitazione dei tag (usualmente “<“ e “>”, etc.

Laddove la dichiarazione SGML manchi, si usano i setting di default, chiamati “reference concrete syntax”.

Il DTD (Document Type Declaration)

Elenca i tipi degli elementi e degli altri componenti.

Ne specifica inoltre tutte le caratteristiche e relazioni.

Le istanze di documenti

L'istanza di un documento specifica il contenuto testuale vero e proprio del documento, annotato dove necessario con i tag del markup.

Essendo possibile includere file diversi dentro al documento SGML, è possibile creare un DTD globale per una classe intera di documenti.

Questo permette di generalizzare struttura e aspetto esteriore di molti documenti dello stesso tipo, ed è uno degli aspetti più apprezzati di SGML.

Sono parti importanti di un'istanza di un documento:

- Elementi
- Attributi
- Data content
- Entity References

Elementi

Gli elementi dividono il documento in componenti logiche e/o significative. Sono identificate dalla presenza di tags all'inizio e alla fine, e sono attribuite ad un tipo che ne indica ruolo e proprietà.

Sono descritte da un nome, chiamato tag name o anche Generic Identifier (GI).

Sono definite nel DTD, e più sono comprensibili meglio è (<citazione> è meglio di <ct>, che è meglio di <tag37>).

Attributi

Attributi possono essere aggiunti ad ogni elemento e permettono di identificare parametri specifici dell'elemento stesso.

Data content

Sono le informazioni da visualizzare, il vero e proprio contenuto del documento.

Entity Reference

Entità sostituite dal parser di SGML con un valore corrispondente. Permettono di specificare in maniera standard parti ripetute di un testo.

Un esempio

Il documento

26 ottobre 1997

Caro Giovanni,

come va? Quando ti *decidi* a venire a trovarmi?

A sentirci a presto

Fabio

II DTD

```
<!DOCTYPE letter [  
<!ELEMENT letter - - (data, saluti, corpo,  
                        chiusura, firma) >  
<!ATTRLIST      letter  
    filecode      NUMBER      #REQUIRED  
    secret        (yes|no)    "no">  
  
<!ELEMENT      corpo - - (para)*>  
<!ELEMENT      (data, saluti, chiusura, firma)  
                - - (#PCDATA)>  
<!ELEMENT      para - - (#PCDATA | enfasi)>  
<!ELEMENT      enfasi - - (#PCDATA)>  
  
<!ENTITY      saluto "Caro">  
>
```

L'istanza del documento

```
<letter filecode="9710260023">  
  <data>26 ottobre 1997</data>  
  <saluti>&saluto Giovanni, </saluti>  
  <corpo>  
    <para>come va?</para>  
    <para>Quando ti <enfasi> decidi  
      </enfasi> a venire a trovarmi?  
    </para>  
  </corpo>  
  <chiusura>A sentirci a presto</chiusura>  
  <firma>Fabio</firma>  
</letter>
```

Un equivalente non-SGML (RTF)

```
\pard\plain \qr\sb120\sl280 \f16 26 ottobre 1997\par
\pard \qj\sb120\sl280 \par
Caro Giovanni, \par
\pard \qj\sb120\sl1480 come va? \par
\pard \qj\sb120\sl280 Quando ti {\i decidi} a venire a
trovarmi? \par
\par
\pard \qj\sb240\sl280 A sentirci a presto\par
\pard \qj\sb120\sl280 \par
{\b Fabio}\par
```


DSSSL

Sebbene non tutte le applicazioni di documenti SGML siano orientate alla visualizzazione, l'applicazione più comune dei documenti SGML rimane la stampa.

Poiché per principio i documenti SGML non contengono istruzioni di formattazione e di stampa, si ricorre ad un elemento esterno, detto “foglio di stile” o “stylesheet”.

Lo standard più diffuso per associare un foglio di stile ad un documento SGML è DSSSL (Document Style Semantics and Specification Language - si legge Diesel), ISO/IEC 10179 del 1995.

Un documento DSSSL permette di specificare quale “rendering” associare agli elementi di un documento strutturato tipo SGML.

DSSSL è un linguaggio complesso e difficile, basato su un linguaggio di programmazione completo di tipo funzionale chiamato Scheme.

Due sono gli elementi costitutivi di una specifica DSSSL:

- un meccanismo di selezione dell'elemento
 - “tutti gli elementi SEZIONE”,
 - “Gli elementi dispari di LISTA”,
 - “il terzo elemento di PARAGRAFO all'interno del quarto elemento di SEZIONE”,
 - ecc.
- una serie di proprietà grafiche da associare all'elemento
 - Nome del font, dimensione, stili, kerning, ecc.
 - Margini, interlinea, allineamento, ecc.
 - Effetti speciali (Drop cap, sfondi colorati, ecc.)

HyTime

Introduzione

HyTime è uno standard molto più recente (ISO 10744, del 1992) per l'introduzione di caratteristiche ipertestuali e dinamiche nello standard SGML. Esso è cioè un'estensione di SGML.

L'origine di HyTime venne dalla descrizione di spartiti musicali nel progetto SDML (Standard Music Description Language).

Esso venne sviluppato da un comitato industriale presieduto dal Goldfarb e diretto da Newcomb

HyTime permette di dare una sintassi standard per esprimere locazioni in vari tipi di dati.

Tra i più importanti, locazioni di partenza ed arrivo di salti ipertestuali, spazi di coordinate in oggetti grafici n-dimensionali, frame specifici in dati dipendenti dal tempo (animazioni, filmati, simulazioni scientifiche, etc.).

Descrizione

HyTime permette di creare hyperdocumenti, che possono essere o singoli documenti SGML che usano costrutti HyTime, o complessi insiemi di documenti altamente interconnessi in una grande varietà di media e formati.

Spesso una collezione di documenti SGML può essere resa conforme a HyTime semplicemente modificandone il DTD.

Alla base di questo insieme c'è uno hub document, il documento da cui il motore HyTime inizia a processare le informazioni. Lo hub document determina anche lo scope dell'hyperdocumento in questione, ovvero l'insieme di documenti in qualche maniera raggiungibili dallo hub.

Le architectural form

Le architectural form sono una tecnica implementativa e definitoria usata per integrare HyTime ed altre estensioni nel corpo di SGML.

Permette alle applicazioni HyTime di rimanere generali e a non obbligarle ad un singolo DTD (così avviene per HTML, come vedremo).

Architectural form è il nome per una classe di elementi che fanno riferimento ad un unico concetto.

Il motore HyTime controlla la presenza e la natura dell'elemento HyTime osservando l'attributo HyTime nella sua lista di attributi.

Esempio:

```
<!ELEMENT      mylink      - - (p)*>
<!ATTLIST      mylink
  linkend      CDATA      #REQUIRED
  HyTime       NAME       "clink">
```

Caratteristiche

Lo standard HyTime è diviso in varie clause, ognuna divisa in varie sub-clause, etc. Dopo la parte introduttiva, ogni clause seguente introduce un modulo, ovvero una serie di costrutti che identifica una comune funzionalità.

I moduli HyTime sono i seguenti:

Base module

Questo modulo definisce i concetti di base, come le architectural form, i moduli HyTime, e una serie di opzioni e parametri generali.

Measurement module

Questo modulo permette di definire ed esprimere unità di misura in qualunque formato, in spazio, tempo o altri domini.

Questo viene usato in parte dal modulo ipertestuale.

Location address module

Questo modulo usa misure, nomi formali ed altre tecniche per indicare locazioni all'interno di strutture come stringhe, alberi, e dati non-SGML.

Fornisce cioè un meccanismo di puntamento utilizzato nel modulo ipertestuale per la creazione di link.

Hyperlinks module

Questo modulo facilita la creazione di link ipertestuali tra locazioni in documenti ed altri tipi di dati.

Definisce i *clink* e gli *ilink*.

I *clink* (*contextual link*) sono link le cui istanze debbono essere presenti all'interno del contesto in cui il link esiste.

Gli *ilink* (*independent link*) sono invece link che possono essere posti fuori linea, cioè non risiedere negli estremi del link.

Scheduling module

Questo modulo fornisce il meccanismo per specificare eventi e schedule nello spazio e nel tempo.

Vengono usati soprattutto per istanze di sincronizzazione tra elementi diversi in dati temporali (es. video, o musica).

Rendition module

Questo modulo permette di descrivere aspetti di presentazione e rendering dei dati.

Ricapitolo

Cosa ricordare per il momento di SGML

- Distinzione tra markup specifico e generico
- Specificazione degli elementi ammessi in un DTD
- Scopo e differenze di elementi, attributi, data content ed entità
- Differenza tra un elemento ed il suo tag
- SGML non si occupa visualizzazione. Per questo c'è DSSSL
- SGML non parla di azioni o di rapporti tra le parti. Per questo c'è HyTime