



UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

RELAZIONE

**Comic Book Markup Language:  
integrazioni ed esempi d'uso**

*Relatore*

Mirko Tavosanis

*Candidato*

Francesca Pesciatini

*Correlatore*

Vincenzo Gervasi

Anno Accademico 2010-2011

# Indice

<b>Introduzione.....</b>	<b>4</b>
<b>Capitolo 1 - La codifica del fumetto.....</b>	<b>6</b>
1.1 Quali aspetti devono essere codificati.....	6
1.1.1 Metadati.....	6
1.1.2 Macrostruttura.....	7
1.1.3 Contenuti testuali.....	7
1.1.3.1 Elementi tipografici.....	8
1.1.3.2 Aspetti linguistici.....	8
<b>Capitolo 2 - Progetti di codifica esistenti: ricerca e valutazioni.....</b>	<b>10</b>
2.1 La fase di ricerca: esistono schemi di codifica XML per il fumetto?.....	10
2.2 I risultati della ricerca.....	10
2.2.1 ComicsML.....	11
2.2.2 CBML.....	12
2.3 Scelta dello schema.....	14
<b>Capitolo 3 - Criteri per la codifica: esempi di applicazione.....</b>	<b>15</b>
3.1 Utilizzo del <i>tutto maiuscolo</i> .....	15
3.2 Legame tra elementi iconici e testuali.....	16
3.3 Posizione della nuvoletta rispetto al locutore.....	17
3.4 Numerazione dell'ordine di battuta.....	19
3.5 Assegnazione di un ID in vignette con un elevato numero di personaggi non identificabili.....	19
3.6 Personaggio invisibile.....	20
3.7 Nuvolette multiple collegate.....	21
3.8 Duplice funzione del balloon.....	22
3.9 Elementi testuali esterni ai balloon.....	23

3.9.1 Suoni/Rumori.....	24
3.9.2 Frammenti testuali.....	25
3.10 Caption: forma del contenitore.....	26
3.11 Caratteristiche linguistiche.....	27

<b>Capitolo 4 - Oltre lo studio linguistico: immagini, note e visualizzazione dei dati.....</b>	<b>30</b>
4.1 L'elemento figure.....	30
4.2 Riferimenti incrociati.....	31
4.3 La visualizzazione dei dati.....	31
<b>Conclusioni.....</b>	<b>33</b>

## ***Introduzione***

Nel suo saggio *Il sistema fumetto*, Thierry Groensteen concepisce il fumetto come un “edificio sofisticato” nel quale ogni elemento (testo, immagini, layout, etc.) coopera con gli altri per dare vita ad un vero e proprio linguaggio che non può essere confuso con altri tipi di linguaggio; una *macrostruttura* contenente tavole, vignette ed elementi interni alle vignette.

Questa descrizione esprime chiaramente quanto un genere come il fumetto si distingua dagli altri. A differenza di un prodotto letterario di carattere più tradizionale come un romanzo o una poesia, costituito esclusivamente dall'elemento testuale, il fumetto può essere definito come un genere a dominante visiva (Groensteen 2011, p. 12.), ovvero un genere in cui molte delle informazioni semantiche e narrative sono veicolate dalle immagini. Ciò non significa che l'immagine sia più importante del testo e degli aspetti linguistici; tuttavia i vari studi dedicati al linguaggio fumettistico si concentrano prevalentemente su analisi semiotiche relative al rapporto immagine/testo, mentre poca è l'attenzione rivolta agli aspetti relativi alla lingua.

Avvalendosi degli strumenti informatici, in particolare della codifica XML, che consente di strutturare un documento marcandone gli aspetti ritenuti rilevanti non solo dal punto di vista strutturale ma anche da quello linguistico, può essere effettuato uno studio più completo di una *specie* narrativa come il fumetto (Groensteen 2011, p. 12.).

Questa tesi si pone come punto di partenza di un progetto più ampio finalizzato alla definizione di criteri di codifica per il linguaggio del fumetto. Questo primo approccio consiste, in particolare, nell'analisi di progetti esistenti e nella successiva individuazione e applicazione di uno schema XML-TEI che possa essere una base valida su cui elaborare eventuali integrazioni future.

I capitoli di questa relazione illustrano le fasi in cui il lavoro si è articolato.

Nel primo capitolo vengono trattati ad un livello teorico gli aspetti generali da considerare per eseguire la codifica del fumetto. Tali aspetti sono raggruppati in

categorie in base alla loro natura e discussi individualmente.

Nel secondo capitolo è descritta la ricerca preliminare di schemi di codifica XML per il fumetto già esistenti. Vengono poi esposte le modalità di analisi dei progetti individuati e le motivazioni che hanno portato a scegliere il *Comic Book Markup Language* come base per la codifica.

Il terzo capitolo riporta i criteri adottati per la codifica, motivati dal punto di vista linguistico e supportati da esempi di applicazione pratica.

Nel quarto capitolo sono trattati aspetti più generali legati al progetto di codifica, in particolare l'inserimento di immagini e di riferimenti interni per le note. Un ultimo sguardo è riservato alle possibilità offerte da strumenti come fogli di stile XSL per la visualizzazione del documento in un browser.

# 1

## La codifica del fumetto

“Codificare un messaggio già codificato – un testo – significa per prima cosa esplorare il codice di partenza, studiarne funzioni e regolarità, comprenderlo in profondità, de-codificarlo: solo in questo modo sarà poi possibile ri-codificare lo stesso messaggio/testo in modo che possa essere comprensibile/utilizzabile da una macchina.” (Pierazzo, 2005 p.16).

### 1.1 Quali aspetti devono essere codificati

Al fine di codificare il linguaggio fumetto è necessario, in primo luogo, porsi a un certo livello di astrazione per individuare e valutare agli aspetti lo caratterizzano. Tali caratteristiche possono essere raggruppate come di seguito:

- metadati
- macrostruttura
- contenuti testuali
  - aspetti tipografici
  - aspetti linguistici

#### *1.1.1 Metadati*

Una prima tipologia di informazioni che uno schema di codifica deve prevedere comprende i dati generali sull'opera: l'editore, il titolo, l'autore, la data di pubblicazione e, eventualmente, la metodologia utilizzata per la digitalizzazione.

### ***1.1.2 Macrostruttura***

Con riferimento a quanto afferma Groensteen è possibile suddividere su più livelli di astrazione tutti gli elementi che costituiscono il linguaggio del fumetto individuando uno scheletro nel quale si inseriscono le diverse componenti.

Innanzitutto le *immagini solidali* (Groensteen 2011, p. 22.) che costituiscono il fumetto si dividono uno spazio e gli spazi specifici del fumetto possono essere considerati:

- tavola
- vignetta
- nuvoletta

La distribuzione degli spazi può essere immaginata come una forma mentale rispondente ad un preciso schema narrativo. Questa astrazione deve tradursi nella definizione di elementi di codifica che rispondano ad una determinata gerarchia in cui una tavola contiene una o più vignette e ciascuna vignetta può contenere elementi interni come nuvolette e didascalie.

### ***1.1.3 Contenuti testuali***

Al fine di utilizzare la codifica per studi di tipo linguistico assume un'importanza fondamentale definire come marcare i contenuti testuali, nelle varie forme in cui si presentano nel fumetto, e selezionare consapevolmente gli elementi che costituiranno oggetto di studio. I contenuti delle vignette possono essere classificati in tre tipologie:

- testi interni alle nuvolette (battute pronunciate dai personaggi, pensieri, etc.)
- testi contenuti nelle didascalie
- elementi testuali esterni alle nuvolette come ad esempio i suoni

All'interno delle nuvolette sono contenute le battute dialogiche tra i personaggi, ma anche i loro monologhi, pensieri, etc. e la forma stessa delle nuvolette sottolinea la funzione del testo.

Le didascalie possono contenere testi con differenti funzioni: funzione narrativa in prima o terza persona, funzione descrittiva, riassuntiva e di nota al testo (quando spiegano un particolare riferimento culturale).

La parola è inoltre presente nel fumetto anche al di fuori delle nuvolette e delle didascalie. In molti casi si tratta di rumori o suoni scritti solitamente al lato dell'immagine degli oggetti che li producono, in altri casi di testi contenuti in elementi della scena (numeri di targa, insegne di negozi, e qualunque altro testo).

#### *1.1.3.1 Elementi tipografici*

È opportuno che all'interno della codifica del fumetto a fini linguistici vengano rese il più fedelmente possibile le scelte tipografiche adottate nell'originale come ad esempio il tutto maiuscolo, i grassetto, i corsivi, etc.

#### *1.1.3.2 Aspetti linguistici*

All'interno di un progetto di codifica finalizzato allo studio linguistico è necessario prevedere un sistema per marcare gli aspetti funzionali a tale studio.

L'attenzione deve essere rivolta alla componente verbale contenuta nelle nuvolette ma allo stesso tempo è opportuno codificare dal punto di vista linguistico anche altre componenti testuali discusse in precedenza, in particolare i suoni.

Gli ideofoni utilizzati fuori dalle nuvolette rappresentano infatti un punto di innovazione linguistica nel fumetto.

Gli ideofoni sono una delle caratteristiche più macroscopiche del linguaggio del fumetto. Essi rispondono all'intento di imitare acusticamente la realtà riproducendone suoni e rumori e in tal senso svolgono una funzione descrittiva di commento alle immagini delle quali costituiscono una sorta di colonna sonora. L'abbondanza di ideofoni va quindi collegata alla tendenza

all'iconicità del linguaggio fumettistico in cui, grazie all'impiego di ideofoni, il significato può venire rappresentato fonicamente dal significante (Pietrini 2008, p. 155).

## Progetti di codifica esistenti: ricerca e valutazioni

### 2.1 La fase di ricerca: esistono schemi di codifica XML per il fumetto?

L'analisi preliminare delle caratteristiche proprie del testo fumetto descritta nel capitolo precedente ha trovato una logica prosecuzione in una ricerca di progetti di codifica già esistenti.

Un dato interessante emerso da tale ricerca è la scarsità di progetti sviluppati: se a una prima riflessione questo fatto può sembrare anomalo considerando la diffusione e popolarità del fumetto, una più attenta riflessione porta a trovare motivazioni di diversa natura.

Da un punto di vista strettamente tecnico, codificare un fumetto richiede schemi di marcatura caratterizzati da elementi più complessi rispetto a quelli necessari per generi testuali tradizionali. Basti pensare all'esigenza di codificare in maniera adeguata i legami inscindibili tra gli elementi testuali e quelli grafici. Inoltre, da un punto di vista culturale, le cause del limitato interesse riscontrato potrebbero essere ricondotte alla tendenza a considerare il fumetto come un genere “di serie B”, non degno della medesima attenzione che viene riservata ad altri generi.

### 2.2 I risultati della ricerca

Tra i progetti nati con l'intento di codificare il fumetto due si distinguono per completezza:

- ComicsML
- CBML (*Comic Book Markup Language*)

Una volta selezionati, questi due progetti sono stati analizzati sulla base di al-

cuni criteri:

- Diffusione/Utilizzo
- Compatibilità
- Completezza della codifica
- Facilità d'uso

### 2.2.1 ComicsML

Basato sul linguaggio XML, ComicsML è stato realizzato nel 2001 dal consulente software americano Jason McIntosh con l'obiettivo di codificare fumetti on-line. La descrizione dei contenuti dei fumetti ha come scopo principale la maggiore visibilità, accessibilità e flessibilità degli stessi.

Il sito web [http://jmac.org/projects/comics\\_ml/index.html](http://jmac.org/projects/comics_ml/index.html) contiene tutte le informazioni, ben strutturate, sul progetto di codifica. Analizzando la ComicsML DTD<sup>1</sup> si nota immediatamente la semplicità della struttura mantenuta dall'autore.

È possibile dire che la codifica, anche se semplicistica, è chiara e ben fatta. Sicuramente è molto utile il massiccio uso di commenti nella DTD per la spiegazione di ogni sua parte ciò rende la DTD facile da consultare e probabilmente da riutilizzare o modificare.

La DTD contiene:

- Metadati sul fumetto in generale (*title, person, last-built, etc.*).
- Elementi che identificano strisce e vignette.
- Elementi per il markup del testo (*emphasis, strong, soft*).
- Elementi utilizzati per marcare effetti “vocali” (*interro, bang, interrobang, silence*).
- Attributi utilizzati per descrivere le vignette (*narration, thought, speech, action*).

---

<sup>1</sup> [http://jmac.org/projects/comics\\_ml/comics\\_ml.dtd](http://jmac.org/projects/comics_ml/comics_ml.dtd)

- Attributi utilizzati per descrivere la forma delle nuvolette in relazione al parlato (*normal, shout, whisper, cold, wobbly*).

Questa tipologia di elementi e relativi attributi serve a descrivere la macrostruttura di un fumetto ma non prevede una ripartizione dei ruoli (protagonista, eroe, etc.) né un'associazione delle battute ai personaggi (v. 3.2).

Un altro fattore che scoraggia l'uso di questo schema è la presenza di elementi creati dall'autore non conformi a standard quali la TEI.

### 2.2.2 CBML

CBML (*Comic Book Markup Language*) è stato ideato tra il 2002 e il 2004 da John A. Walsh, professore della facoltà di Biblioteconomia e scienze dell'informazione presso l'università dell'Indiana.

Analizzando la DTD<sup>2</sup> di CBML è possibile verificare la conformità alla TEI P4 e gli adattamenti che l'autore vi ha apportato.

La DTD TEI permette la combinazione di sottogruppi (*DTD subset*) per la definizione di una nuova DTD più specifica per i propri scopi. La *Doctype Declaration* di CBML infatti è così strutturata:

```
<!DOCTYPE cbml PUBLIC "-//TEI P4//DTD Main DTD
  Driver File//EN" [
  <!ENTITY % TEI.XML          'INCLUDE' >
  <!ENTITY % TEI.drama        'INCLUDE' >
  <!ENTITY % TEI.figures      'INCLUDE' >
  <!ENTITY % TEI.linking      'INCLUDE' >
  <!ENTITY % TEI.names.dates 'INCLUDE' >
  <!ENTITY % TEI.extensions.ent SYSTEM
    'cbml.extensions.ent' >
  <!ENTITY % TEI.extensions.dtd SYSTEM
    'cbml.extensions.dtd'>
  ]>
```

---

<sup>2</sup> <http://www.cbml.org/downloads/cbml.dtd>

La prima istruzione `TEI.XML` è necessaria per dichiarare che la DTD che si sta selezionando seguirà la sintassi XML e non quella SGML. È questo il meccanismo attraverso il quale si possono costruire DTD-TEI adatte di volta in volta agli scopi stabiliti per la codifica.

Per la rappresentazione di un fumetto sono stati dichiarati i seguenti set TEI:

`TEI.drama` – insieme di elementi per un testo di tipo drammatico/teatrale.

`TEI.figures` - potranno essere inserite illustrazioni nel testo.

`TEI.linking` - potranno essere utilizzati riferimenti interni o esterni al testo.

`TEI.names.dates` - potranno essere marcati in maniera analitica i nomi propri e le date.

Le ultime due estensioni contengono le modifiche apportate dall'autore alla DTD TEI P4 seguendo le istruzioni presenti nel Capitolo 24 delle linee guida "*Modifying and Customizing the TEI DTD*":

`TEI.extensions.ent` – contiene le informazioni necessarie per modifiche locali alle entità parametriche.

`TEI.extensions.dtd` – dichiarazioni di nuovi elementi e dichiarazioni di elementi modificati.

Nella `TEI.extensions.ent` sono rinominate alcune entità parametriche per adattare meglio ai fumetti:

- `role` diventa `persona`
- `sp` (indica una singola battuta) diventa `balloon`
- `castGroup` diventa `characterGroup` e così via.

Per facilità di consultazione, all'interno del sito<sup>3</sup> è possibile vedere l'elenco (anche se non completo) dei tag utilizzati. Quelli forse più interessanti, in aggiunta ai

---

<sup>3</sup> <http://www.cbml.org/cbmlTagSetDoc.html>

tag strutturali di base presenti anche in ComicsML, sono quelli che fanno parte del Tagset Base Drama<sup>4</sup>. È presente anche una versione estesa non commentata (cbml.flat.dtd).

### 2.3 Scelta dello schema

Entrambe le codifiche esaminate sono spesso citate ed apprezzate sul web.

ComicsML, forse perché pensata per i fumetti on-line, attira principalmente la comunità degli ideatori/lettori di fumetti digitali che ha dato un suo contributo con piccoli software (es. Metrophix – *Comic hosting software*) che utilizzano questa codifica.

CBML si concentra sui supporti tradizionali (fumetti cartacei) ed è progettato anche con lo scopo di preservarli attraverso il processo di digitalizzazione. Sebbene non abbia avuto ancora riscontri pratici di una certa rilevanza, questo formato è stato oggetto di studio e di attenzione in ambito accademico. Il lavoro di Walsh è stato introdotto e spiegato in un articolo della rivista *Digital Humanities Quarterly*<sup>5</sup> ed è stato anche al centro dell'intervento<sup>6</sup> di Ernesto Priego in occasione del “*London Seminar in Digital Text and Scholarship*” che si è svolto nel gennaio 2011.

Il confronto tra ComicsML e CBML fa sicuramente preferire l'utilizzo di quest'ultimo per progetti di codifica del fumetto, in particolare in riferimento a un fattore chiave come la conformità allo standard TEI. Tale conformità, infatti, garantisce la possibilità di riutilizzare e di ampliare lo schema arricchendolo con *tagset* preconfezionati, inclusi quelli rilevanti per l'analisi linguistica, fornendo di volta in volta una *vista* sugli aspetti ritenuti importanti.

La scelta finale dunque ricade sul *Comic Book Markup Language*, considerato nel complesso il migliore strumento attualmente disponibile per gli scopi di questo lavoro.

---

4 <http://www.tei-c.org/release/doc/tei-p4-doc/html/DR.html>

5 Riferimenti alla pubblicazione e ad altri interventi con soggetto CBML sono contenuti all'interno del Curriculum dell'autore <http://www.slis.indiana.edu/faculty/jawalsh/cv/>

6 Per l'abstract <http://ies.sas.ac.uk/events/seminars/Digitaltext/2010-2011.htm>

## Criteria per la codifica: esempi di applicazione

Nel primo capitolo sono stati discussi gli aspetti che dovrebbero essere codificati in un fumetto.

Una volta stabilita la maggiore completezza di CBML come formato di codifica per i fumetti, l'applicazione dello schema ha fatto emergere la necessità di definire dei criteri che non sono esplicitamente illustrati fra le specifiche di CBML ma sono importanti per verificare l'adeguatezza di uno schema di codifica al fine di svolgere analisi linguistiche sul fumetto.

### 3.1 Utilizzo del *tutto maiuscolo*

Fra i criteri della codifica rientra la resa dei caratteri che compongono i testi interni alle didascalie e alle nuvolette. Si tratta di un aspetto variabile: nel caso dei fumetti italiani la soluzione più adottata è l'uso dello stampatello maiuscolo, fumetti prodotti in altri paesi invece hanno caratteristiche diverse. In Germania si fa uso di caratteri tipografici, mentre in altri casi all'uso del maiuscolo si preferisce l'uso del minuscolo. Nel marcare fumetti caratterizzati dal tutto maiuscolo è opportuno mantenere nella codifica questa forma, in quanto normalizzarla sulla base delle regole dei testi a stampa che prevedono l'uso delle maiuscole solo in alcuni punti del testo (a inizio frase, per l'iniziale di nomi propri di persone o luoghi, etc.), potrebbe non coincidere con le intenzioni originali dell'autore.

```
<panel n="8">
  <balloon n="1" type="speech" who="quo"><p>L'ULTIMO CALO
  DI BORSA HA TRAVOLTO ANCHE LO ZIONE!</p></balloon>
  <balloon n="2" type="speech" who="qui"><p>QUO, PREPARA
  BENDE E CEROTTI!</p></balloon>
</panel>
```

### 3.2 Legame tra elementi iconici e testuali

Nel fumetto ciò che porta avanti la narrazione è la cooperazione di elementi iconici ed elementi testuali. L'immagine del personaggio aggiunge senso alla parola e viceversa. Per questo motivo è necessario codificare la corrispondenza tra la nuvoletta di parlato ed il personaggio. “Non esiste nuvoletta che non rinvii e non debba essere attribuita a un locutore noto o presunto (Groensteen 2011, p. 72)” e per questo motivo uno schema di codifica deve prevedere necessariamente un meccanismo di associazione chiara tra nuvoletta e parlante. Come visto nel capitolo precedente ComicsML non prevede questo meccanismo al contrario di CBML. Quanto appena detto assume un'ulteriore importanza in ottica linguistica in quanto spesso le battute pronunciate denotano proprietà linguistiche particolari del parlante e questo rende necessario creare un collegamento fra il personaggio e ciò che dice. Dallo studio effettuato da Daniela Pietrini sulla lingua dei fumetti Disney pubblicati in Italia emerge ad esempio come il personaggio Pippo, e solo quello, usi in maniera sistematica l'interiezione resa attraverso il termine onomatopeico *YUK!* o come l'alieno Eta Beta anteponga una -p ad alcuni lemmi.

Nell'*header* del documento XML è inserita la lista dei personaggi presenti nel fumetto. Ogni personaggio, `persona`, è identificato in modo univoco con un `id`. L'attributo `type` consente di specificare in modo facoltativo il ruolo assunto dal personaggio all'interno della storia (eroe, protagonista, antagonista, aiutante, etc.).

```
<characterList>
  <characterItem>
    <persona id="dylan_dog" type="protagonista">
      Dylan Dog
    </persona>
    <persona id="amy">Amy</persona>
    <persona id="daniel">Daniel</persona>
  </characterItem>
</characterList>
```

Nel corpo del documento, il contenuto dei balloon al personaggio che parla è stato associato attraverso l'attributo `who` dell'elemento `balloon`,

```
<balloon type="speech" who="dylan_dog">
```

L'attributo `type` permette invece di specificare se il contenuto della nuvoletta viene pronunciato o è un pensiero del personaggio. L'autore di CBML suggerisce come valori per l'attributo `type` dell'elemento `balloon`:

- *thought* per nuvolette di pensiero
- *speech* per nuvolette di parlato
- *broadcast* per le nuvolette che contengono testo o suoni emessi dalla radio, televisione o qualsiasi dispositivo tecnologico.

```
<balloon type="thought" who="dylan_dog">
```

### 3.3 Posizione della nuvoletta rispetto al locutore

Esistono casi in cui la nuvoletta di parlato non è posizionata vicina al locutore; in casi limite si trova anche in una vignetta diversa da quella in cui si trova il personaggio. In questi casi la codifica dà priorità alla corrispondenza tra nuvoletta e parlante, ritenuta più significativa dal punto di vista linguistico rispetto alla distribuzione grafica adottata dall'autore.

Nella Fig. 1, la prima vignetta, mostra un personaggio, un principe, che pronuncia una battuta. La nuvoletta di parlato, in questo caso, non è posizionata vicino al locutore ma è posta fisicamente nella vignetta successiva.



Fig.1 – Tavola di Sergio Toppi, Sharaz-de (*Alter Alter*, Milano Libri Edizioni, 1979, pag.11)

<panel n="1">

[...]

<balloon n="2" type="speech" who="principe"><p>CHE SIA BANDITA UNA GRANDE CACCIA. TUTTI I PIU' VALENTI CACCIATORI SI RADUNINO AI MIEI COMANDI.</p></balloon>

</panel>

### 3.4 Numerazione dell'ordine di battuta

Quando siamo in presenza di più parlanti le nuvolette sono numerate nell'ordine in cui devono essere lette aggiungendo un attributo con valore numerico all'elemento `balloon`.



Fig. 2 – Vignetta tratta da Topolino (numero I-2271-6, 8 giugno 1999, pag. 156)

```
<balloon n="1" type="speech" who="battista"><p>LA PROGNOSE PER LE VOSTRE FINANZE E' INFAUSTA, SIGNORE!</p></balloon>  
<balloon n="2" type="speech" who="paperone"><p>HO DI CHE ACQUISTARE UNA <emph rend="bold">RAZIONE DI CICUTA?</emph></p></balloon>
```

### 3.5 Assegnazione di un ID in vignette con un elevato numero di personaggi non identificabili

In alcune vignette è possibile trovare un numero elevato di personaggi non identificabili; per essere in grado di collegare comunque gli enunciati verbali ad un parlante la soluzione adottata è quella di assegnare un ID univoco ad ogni personaggio, da sinistra verso destra, utilizzando semplicemente le lettere

dell'alfabeto dato che la numerazione è già stata utilizzata per codificare l'ordine di battuta.

Nella vignetta in Fig. 3 al primo personaggio non identificabile è stato assegnato l'ID `pg_A`, al secondo l'ID `pg_B` e così via.



Fig. 3 - Vignetta tratta dalla raccolta *Andrea Pazienza* (Editori del Grifo, 1991)

```
<characterItem><persona id="pg_A">Personaggio  
A</persona></characterItem>  
<characterItem><persona id="pg_B">Personaggio  
B</persona></characterItem>  
<characterItem><persona id="pg_C">Personaggio  
C</persona></characterItem>
```

### 3.6 Personaggio invisibile

Un personaggio può essere nascosto e non sempre rintracciabile magari perché mescolato ad una folla. Nella codifica la non identificabilità del personaggio è indicata con il valore `pg_non_identificabile` assegnato all'attributo `id` dell'elemento `persona`.





Fig. 5 - Vignetta tratta da *Dylan Dog - Alfa e omega* (Sergio Bonelli Editore, febbraio 1992, pag. 37)

Anche se contengono battute pronunciate dallo stesso personaggio, esse sono codificate attraverso una ripetizione dell'elemento `balloon`; si è deciso di non fare distinzione tra casi in cui le nuvolette sono fisicamente separate e casi in cui non c'è stacco.

```
<balloon type="speech" who="dylan_dog"><p>INTENDIAMOCI, PRIMA DI
  CONVINCERMI CHE UN <seg type="prestito_straniero" lang="en"
  ><emph rend="bold">UFO</emph></seg>E' ATTERRATO QUI VICINO
  PORTANDO <seg type="prestito_adattato"
  lang="fr">&quot;VENUSIANI&quot;</seg>E MOSTRI, VOGLIO VEDERLO
  CON I MIEI OCCHI...</p></balloon>
```

```
<balloon type="speech" who="dylan_dog"><p>MA CERTO, ORMAI, NON POSSO
  NEGARE CHE <emph rend="bold">&quot;QUALCOSA&quot;</emph>STA
  ACCADENDO, PURTROPPO.</p></balloon>
```

### 3.8 Duplice funzione del *balloon*

Non è detto che alle vignette sia assegnata sempre un'unica funzione (parlato o pensiero); capita che i pensieri/ricordi del personaggio comprendano anche una funzione didascalica in quanto introducono gli eventi facendo da filo conduttore alla narrazione. Per non tralasciare questo valore aggiunto, nella codifica è marcata questa duplice funzione all'interno dell'attributo `type` dell'elemento `balloon`. Nell'esempio in Fig. 6 la didascalia fa riferimento al tempo presente della narrazione e il testo contenuto nella nuvoletta rimanda a una scena avvenuta nel passato rispetto

al resto della storia.



Fig. 6 - Vignetta tratta da Zanardi, Edizione Critica (Baldini & Castoldi 1998, pag. 50)

<panel n="4">

<caption><emph rend="bold">POICHE' L'AMICO SAPEVA BENISSIMO CHE <seg type="modo\_di\_dire\_colloquiale">NON AVEVO MEZZA LIRA</seg>, <seg type="imprecazione">PERDIO</seg>, NON MI SUCCEDE QUASI MAI...</emph></caption>

<balloon type="thought-caption" who="maurizio"><p>STASERA ESCO CON QUELLA SABRINA, SOLO DEVO <seg type="gergo\_giovanile\_popolare">BECCARE</seg> GIGGETTO PERCHE' <seg type="gergo\_giovanile">HO SPESO PER LA ROBA E SONO NEI CASINI COI MIEI PER LA PILLA...</seg></p></balloon>

<stage type="titolo\_rivista">FRIGIDAIRE</stage>

<stage type="targa\_armadietto">ARMADIETTO SCIENZ</stage>

</panel>

### 3.9 Elementi testuali esterni alla nuvoletta

Come detto in precedenza il fumetto è ricco di elementi a metà tra disegno e parola quali le onomatopее, ma anche innumerevoli frammenti testuali (lettere, ritagli di giornale, scritte varie) inseriti nelle vignette e che costituiscono una forma di testo che si fa immagine.

Gli elementi testuali esterni ai contenitori nuvoletta o didascalia possono dunque essere considerati di due tipi: suoni/rumori e frammenti testuali. Nella vignetta in Fig. 7 si riscontra l'utilizzo di suoni/rumori come TA-PUM e TRRR ed elementi testuali inseriti nella scena come la scritta VIA! sul cartello posizionato nel prato.



Fig. 7 – Vignetta tratta da Topolino (numero I-2271-6, 8 giugno 1999, p. 156)

### 3.9.1 Suoni/Rumori

Hanno un importante valore semantico in quanto aggiungono senso alle scene. La dimensione della rappresentazione grafica indica l'intensità del suono/rumore ma anche la vicinanza o meno della fonte. Questa dimensione può essere estremamente variabile, dal piccolissimo al grandissimo. Sulla base di considerazioni soggettive, nell'header del documento è dichiarata l'interpretazione del formato dei suoni limitando il numero di valori possibili a tre: "small" o "large" o "xx-large" che possono essere assegnati all'attributo `rend` di tipo IDREF nell'elemento `sound`.

```
<rendition type="dimensione_testo_piccolo"
id="small"></rendition>
<rendition type="dimensione_testo_grande"
id="large"></rendition>
```

```
<rendition type="dimensione_testo_molto_grande" id="xx-  
large"></rendition>
```



Fig. 8 - Vignetta tratta da *Dylan Dog. Alfa e omega* (Sergio Bonelli Editore, febbraio 1992, pag. 51)

```
<sound type="battere_il_pugno" rend="xx-large">TUMP TUMP</sound>
```

### 3.9.2 Frammenti testuali

Questa tipologia testuale è anch'essa portatrice di significato e può dare indicazioni aggiuntive sui personaggi e sul contesto in cui si svolgono gli eventi (Fig. 9).

Allo scopo di rappresentare attraverso la codifica il legame inscindibile tra immagine e testo è stato utilizzato l'elemento `stage` che nella TEI è usato per marcare qualsiasi tipo di didascalia e direttiva di scena all'interno di un testo o di un frammento di testo drammatico. Sfruttando l'attributo `type` è possibile specificare a quale elemento grafico si riferisce il testo.

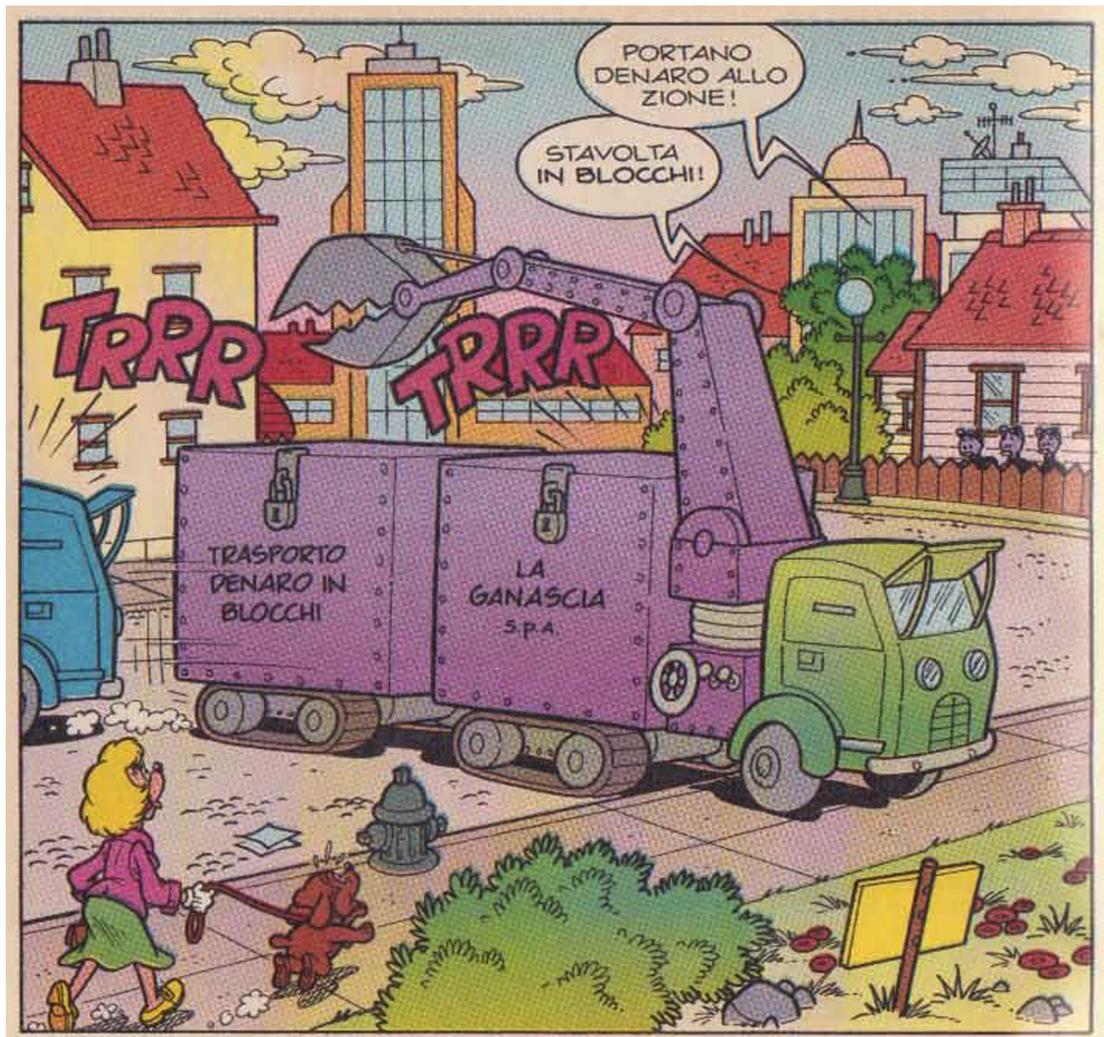


Fig. 9 – Vignetta tratta da Topolino (numero I-2271-6, 8 giugno 1999, pag. 154)

```

<stage type="scritta_sul_cingolato"><emph
rend="bold">TRASPORTO DENARO IN BLOCCHI</emph></stage>
<stage type="scritta_sul_cingolato"><emph rend="bold">LA
GANASCIA</emph>S.p.a.</stage>
  
```

### 3.10 Caption: forma del contenitore

Sebbene esistano casi in cui l'autore sceglie di utilizzare forme diverse da quella rettangolare per le didascalie, fra i criteri adottati per la codifica si è scelto di non inserire l'indicazione della forma dei contenitori *caption* poiché non si tratta di

un fenomeno rilevante per gli scopi di questo lavoro. Uno studio come quello di Daniela Pietrini sul linguaggio dei fumetti Disney è realizzato infatti senza tener conto di questo aspetto.

### 3.11 Caratteristiche linguistiche

Per quanto riguarda la marcatura di aspetti linguistici che non sono dichiarati nella DTD di CBML è stato adottato il tag `seg` che è un elemento di segmentazione generica. Il vantaggio di questo marcatore è che consente la nidificazione:

```
<seg type="frase">
  <seg type="parola"></seg>
  <seg type="parola"></seg>
  <seg type="parola"></seg>
</seg>
```

Il valore dell'attributo `type` di tipo CDATA non impone restrizioni alla scelta della stringa da utilizzare per descrivere il fenomeno linguistico.

L'elemento `seg` è utilizzato, ad esempio, per codificare una caratteristica peculiare di alcune vignette di Andrea Pazienza (Fig. 10) che spesso fa uso di un linguaggio curioso e strampalato molto vicino ad una parlata dialettale.

Nell'esempio riportato in figura è possibile riscontrare onomatopee e battute che ricordano i dialetti napoletano e pugliese.



Fig. 10 - Vignetta tratta dalla raccolta *Andrea Pazienza* (Editori del Grifo, 1991)

<panel n="1">

```

<balloon n="1" type="speech" who="pg_A"><p><seg
type="ideofono">HAWK!</seg></p></balloon>
<balloon n="2" type="speech" who="pg_B"><p><seg
type="dialetto_pugliese">NO SINCIERAMANT, MIK AVEV' KAPIT
CH'ERAVAT SKONVOLT D'MESKALIN! IO PENZAV: AVRANN FUMAT, SARANN
SCIEM!</seg></p></balloon>
<balloon n="3" type="speech" who="pg_C"><p><seg
type="dialetto_pugliese">A'MMESCALIN' CHIU' GROSS E BON
CHAGGHJ VIST!</seg></p></balloon>
<balloon n="4" type="speech" who="pg_C"><p>GIURO!
</p></balloon>
<balloon n="5" type="speech" who="pg_D"><p><seg
type="dialetto_pugliese">DOV' SIAM?</seg></p></balloon>
<balloon n="6" type="speech" who="pg_D"><p><seg
type="dialetto_napoletano">AKKA' NISIUNO O'SSAPE!
</seg></p></balloon>
<balloon n="7" type="speech" who="pg_E"><p><seg
type="ideofono">GNIGNIGNI</seg></p></balloon>
<balloon n="8" type="speech" who="pg_F"><p><seg
type="dialetto_pugliese">MACCHETTIMPORT, <seg
type="gergale_scherzoso">FRICCHETTO'</seg>, ADESS AI TRUVAT

```

```

A'CCHIAVE! STAI CUNTENT!</seg></p></balloon>
<balloon n="9" type="speech" who="pg_G"><p><seg
type="dialetto_pugliese">...<seg type="gergale_scherzoso">E
FRIK'T'N</seg></seg></p></balloon>
<balloon n="10" type="speech" who="stefania"><p><seg
type="serie_ideofonica">HAWK HAWK HAWK</seg> <seg
type="dialetto_pugliese">OGNEANN, OGNEANN TUTT'ELLANN AKKA
TRUWAMM I FUNCHI!</seg></p></balloon>
<balloon n="11" type="speech" who="pg_L"><p>TACI
STEFANIA.</p></balloon>
<stage type="cartello_tenuto_dal_pg_M">fin</stage>
</panel>

```

Visti gli esempi di applicazione della codifica appena esaminati è opportuno sottolineare che l'elemento *seg*, per quanto utile, è pur sempre generico e non consente di differenziare tra fenomeni linguistici diversi, se non con l'attributo *type*. Un'integrazione apportata alla DTD del CBML indispensabile a fini linguistici è quindi l'aggiunta del tagset *analysis* previsto dalla TEI che contiene marcatori specifici come *s* per la frase, *w* per la parola (con *lemma* come attributo), *m* per il morfema, *c* per la punteggiatura e così via.

```
<!ENTITY % TEI.analysis 'INCLUDE' >
```

## Oltre lo studio linguistico: immagini, note e visualizzazione dei dati

Nei precedenti capitoli sono stati discussi e approfonditi i principali criteri di codifica e la loro applicazione pratica per scopi linguistici attraverso l'uso di CBML.

In questo lavoro sono state prese in considerazione anche ulteriori esigenze che, pur non riguardando la sfera linguistica, devono comunque essere considerate per favorire una fruizione più completa della codifica.

Nel caso del fumetto può risultare molto utile, ad esempio, inserire riferimenti alle immagini delle diverse pagine codificate e aggiungere elementi XML che permettano di creare riferimenti a note o ad altri punti interni al documento; questi particolari aspetti sono infatti previsti dal CBML con la dichiarazione dei tagset *figures* e *linking*.

Oltre a questi aspetti funzionali è stato anche affrontato il problema della visualizzazione dei dati in una forma più gradevole all'interno di un browser.

### 4.1 L'elemento *figure*

Fin qui è stato più volte evidenziato lo stretto legame che esiste fra elemento testuale ed elemento grafico nel fumetto. Con riferimento a questo legame risulta utile avere un costante riscontro visivo delle pagine oggetto di codifica.

La soluzione che può essere adottata per soddisfare questa esigenza è l'utilizzo dell'elemento XML *figure* per associare a ciascuna pagina codificata l'immagine corrispondente.

```
<figure entity="figural1"/>
```

L'attributo `entity` è usato per indicare un'entità XML che contiene l'immagine

e il suo funzionamento si basa sull'aggiunta della seguente definizione interna al documento:

```
[  
<!NOTATION JPEG SYSTEM 'image/jpeg'>  
<!ENTITY figural SYSTEM "figural.jpg" NDATA JPEG>  
]
```

## 4.2 Riferimenti incrociati

Esistono esempi di opere a fumetti in cui la storia narrata è approfondita attraverso l'uso di note relative alle scelte dell'autore sui personaggi, sulle scene e i luoghi che fanno da contesto alle vicende narrate. È il caso di *From Hell, graphic novel* ispirata al personaggio e ai delitti di Jack lo Squartatore.

Per gestire opportunamente i riferimenti incrociati interni al testo si può pensare ad un sistema di ancore nella struttura XML. La TEI fornisce diverse soluzioni per la creazione di riferimenti: una scelta possibile può essere l'utilizzo dei marcatori `anchor` e `ref` che, usati in combinazione, permettono di collegare fra loro elementi interni al testo.

## 4.3 La visualizzazione dei dati

Come accennato in precedenza, per fruire al meglio della codifica può essere necessario visualizzare il risultato su browser in una forma più funzionale e gradevole rispetto al semplice albero XML.

A tale scopo si può far uso di una serie di strumenti attraverso i quali trasformare la struttura XML in una struttura HTML alla quale applicare successivamente un foglio di stile per gestire l'aspetto grafico del documento.

Il primo passo da effettuare è la creazione di un foglio di stile XSL (*eXtensible Stylesheet Language*). Questo linguaggio è composto da tre parti: un linguaggio di trasformazione (XSLT), un linguaggio per la navigazione nei documenti (XPath) e un linguaggio per la formattazione (XSL:FO).

La sintassi XPath può essere usata per navigare l'albero XML e selezionare elementi in base ai legami gerarchici con gli altri elementi o in base ai valori dei loro attributi; XSLT può invece essere utilizzato per effettuare la trasformazione in codice HTML e l'inserimento in questo codice dei contenuti selezionati attraverso le espressioni XPath.

Per una resa grafica più completa del documento codificato può essere infine utilizzato un foglio di stile CSS tramite il quale applicare regole di formattazione agli elementi HTML appena creati a partire dall'XML di origine.

È opportuno sottolineare che l'applicazione di questi strumenti per la visualizzazione non si porrebbe lo scopo di rendere fedelmente le pagine del fumetto codificato all'interno di un browser, ma soltanto di fornire una rappresentazione grafica tramite la quale poter mettere in evidenza i diversi elementi codificati, compresi quelli di interesse prettamente linguistico.

## Conclusioni

Premesso che indipendentemente dallo schema di codifica utilizzato *non* tutte le caratteristiche del fumetto possono essere codificate, si può affermare che CBML offre la possibilità di marcare gli elementi strutturali del fumetto in modo accurato. Non si riscontra altrettanta accuratezza per quanto riguarda l'attenzione agli aspetti linguistici: probabilmente ciò non rientrava tra gli obiettivi primari dell'autore. Come dimostra Pietrini in *Parola di Papero* ogni aspetto ritenuto rilevante a fini linguistici (segni di interpunzione, prefissi, suffissi, struttura di una proposizione, etc.) deve poter essere individuato in modo preciso. Per questo motivo l'aggiunta del tagset *analysis* risulta indispensabile e rappresenta un primo passo per rendere questo schema un valido strumento per analisi linguistiche sul fumetto.

## Bibliografia e Sitografia

Groensteen Thierry (2011), *Il sistema fumetto*, Prospettiva Globale Edizioni, Genova.

Pierazzo Elena (2005), *La codifica di testi. Un'introduzione*, Carocci, Roma.

Pietrini Daniela (2008), *Parola di papero. Storia e tecniche della lingua dei fumetti Disney*, Franco Cesati Editore, Firenze.

Troiani Gianluca (2005), *CSS. Guida completa*, Apogeo, Milano.

### Sitografia

TEI, *Text Encoding Initiative*

<http://www.tei-c.org> (visitato il 25 ottobre 2011)

W3C, *World Wide Web Consortium*

<http://www.w3.org/> (visitato il 29 dicembre 2011)

ComicsML, *Comics Markup Language*

[http://jmac.org/projects/comics\\_ml/](http://jmac.org/projects/comics_ml/) (visitato il 15 ottobre 2011)

CBML, *Comic Book Markup Language*

<http://www.cbml.org/> (visitato il 15 ottobre 2011)