

Introduzione alla codifica XML per i testi umanistici

Daniele Silvi, Domenico Fiormonte, Fabio Ciotti
fiormont@uniroma3.it - silvi@lettere.uniroma2.it - ciotti@lettere.uniroma2.it

La digitalizzazione dei testi / 1

- Negli ultimi anni si è verificata una vera e propria esplosione delle iniziative di digitalizzazione
- Tuttavia non sempre l'aspetto qualitativo dell'operazione di digitalizzazione è tenuto nel dovuto conto

La digitalizzazione dei testi / 2

- Con **aspetto qualitativo** intendiamo la considerazione di tutti quegli elementi teorici e tecnologici che rendano le risorse testuali digitalizzate:
 - intellettualmente integre
 - Accessibili nello spazio e nel tempo

Requisiti di un'edizione scientifica elettronica / 1

- Un'edizione scientifica elettronica può offrire una serie di strumenti analitici e di apparati che la rendono proficuamente utilizzabile da un ampio spettro di utenti (non necessariamente specialisti)
- La creazione di edizioni elettroniche di testi scientificamente e criticamente adeguate è un compito intellettuale tanto opportuno ai fini del miglioramento della ricerca scientifica quanto necessario alla preservazione del patrimonio culturale

Requisiti di un'edizione scientifica elettronica / 3

- Un'edizione scientifica elettronica deve assumere una forma tale da garantire il migliore equilibrio tra necessità e vincoli imposti dal medium digitale e requisiti di **integrità intellettuale**

Requisiti di un'edizione scientifica elettronica / 4

- L'integrità intellettuale implica:
 - rispetto tanto per il **contenuto** che per la **forma** originaria del documento da digitalizzare. A tale fine si renderanno necessarie:
 - annotazioni editoriali che indichino al lettore come utilizzare il testo
 - rappresentazione esplicita della tradizione testuale dell'opera mediante un apparato di varianti selettivo o esaustivo
 - specificazione delle fonti primarie utilizzate per la costituzione del testo ed eventuale loro trascrizione comprensiva di tutti i dati relativi al loro stato (cancellazioni, omissioni, lacune etc.)
 - indicazione esplicita delle correzioni e delle congetture interpretative effettuate dall'editore nella costituzione del testo, con specificazione del livello di certezza che l'editore assegna a ciascuna di esse

Requisiti di un'edizione scientifica elettronica / 5

- Un'edizione scientifica elettronica deve essere accessibile al maggior numero di utenti possibile
- Un'edizione scientifica elettronica deve godere di una sufficiente longevità, almeno pari a quella di un'edizione cartacea
 - “sufficiente” significa atta a garantire la preservazione a lungo termine delle risorse digitalizzate

Requisiti di un'edizione scientifica elettronica / 6

- La base per la creazione di un'edizione scientifica digitale (o di un archivio di edizioni digitali) non può in nessun modo essere il software

- Perché?

Requisiti di un'edizione scientifica elettronica / 7

- Nessun software è adeguato per ogni genere di utilizzazione
- Nessun software, per quanto efficiente e versatile può essere gradito a tutti i possibili utilizzatori di una risorsa testuale digitalizzata
- Ma soprattutto, i software hanno un ciclo di vita estremamente ridotto e sono legati a una particolare piattaforma informatica

Requisiti di un'edizione scientifica elettronica / 8

- Ma se il software non è la risposta, come è possibile realizzare edizioni scientifiche digitali che rispettino i requisiti di **integrità** e **accessibilità**?
- Innanzitutto dobbiamo scegliere strumenti in grado di inglobare la maggior parte delle informazioni rilevanti a livello di rappresentazione dei dati (o *codifica*)
- Ciò si traduce nella scelta del **sistema di codifica** ottimale per la rappresentazione delle informazioni testuali

Ma se il software non è la risposta, come è possibile realizzare edizioni elettroniche che siano a un tempo dotate dei requisiti intellettuali sopraelencati, portabili su più piattaforme informatiche e utilizzabili in molteplici contesti applicativi? Non esiste una soluzione ultima e definitiva a questo problema. Si tratta piuttosto di adottare un insieme di strategie che possano approssimare la soluzione ottimale, in base a una serie di considerazioni.

La codifica dei testi / 1

- L'edizione digitale di un documento richiede la rappresentazione dell'informazione contenuta in una fonte testuale in un formato utilizzabile da un elaboratore (*Machine Readable Form*), ovvero una **codifica**
- A tale fine occorre utilizzare un apposito **linguaggio informatico** che deve rispondere ai vincoli formali imposti dalla elaborazione automatica e allo stesso tempo deve essere sufficientemente espressivo per rappresentare la complessità dell'oggetto "testo"

La codifica dei testi / 2

- La codifica è la **rappresentazione formale** di un testo a un qualche livello descrittivo mediante un linguaggio informatico
- Si tratta dunque di un processo **rappresentazionale** che implica una serie di operazioni di selezione e classificazione degli elementi rilevanti in funzione di un determinato punto di vista

I linguaggi di markup / 1

- Per conseguire un'adeguata rappresentazione delle caratteristiche di un testo sono stati sviluppati i *markup language*, linguaggi di codifica del testo
- I ML sono costituiti da un insieme di istruzioni, ciascuna dotata di particolari funzioni
- Le istruzioni di un markup language sono costituite da stringhe di caratteri visibili e delimitate da caratteri, dette **tag** (etichette)
- Una **sintassi** regola l'uso la forma e i rapporti tra i tag

XML: cosa è

- XML: Extensible Markup Language:
 - è un *linguaggio* che consente la rappresentazione di documenti e dati strutturati su supporto digitale
 - è uno dei più potenti e versatili sistemi per la creazione, archiviazione, preservazione e disseminazione di documenti digitali...
 - ... ma la sua sintassi rigorosa e al contempo flessibile ne rende possibile l'applicazione anche nella rappresentazione di dati strutturati, fornendo una soluzione alternativa ai tradizionali sistemi DBMS relazionali

XML: le origini

- XML è stato sviluppato dal World Wide Web Consortium (<http://www.w3.org>)
- Le specifiche sono state rilasciate come *W3C Recommendation* nel 1998 e aggiornate nel 2004
- XML deriva da SGML, un linguaggio di mark-up dichiarativo sviluppato dalla International Standardization Organization (ISO), e pubblicato ufficialmente nel 1986 con la sigla ISO 8879
- XML nasce come un sottoinsieme semplificato di SGML orientato alla utilizzazione su World Wide Web...
- ... ma ha assunto ormai un ruolo autonomo e una diffusione ben maggiore del suo progenitore

XML: caratteristiche

- XML è un metalinguaggio, che permette di definire sintatticamente linguaggi di mark-up
- Un linguaggio XML permette di esplicitare la (le) struttura(e) di un documento in modo formale mediante marcatori (*mark-up*) che vanno inclusi all'interno del testo (*character data*)

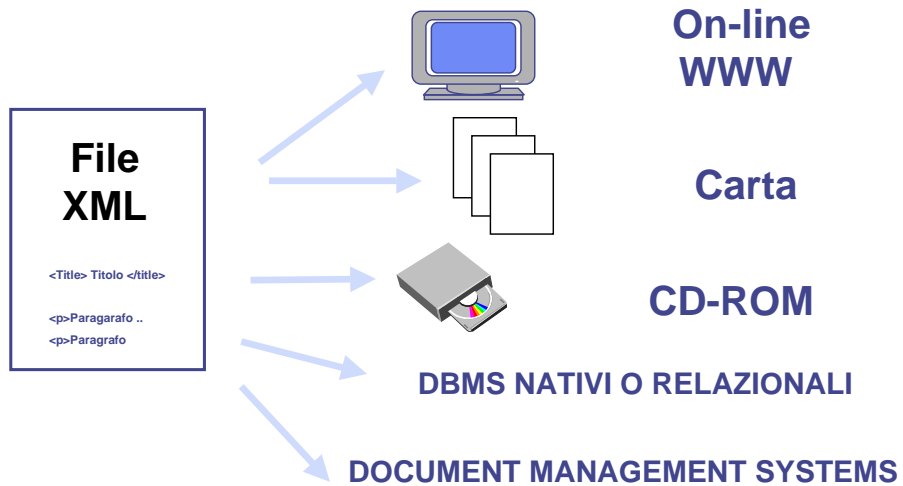
XML: caratteristiche / 2

- XML adotta un formato di file di tipo testuale: sia il mark-up sia il testo sono stringhe di caratteri
- XML si basa sul sistema di codifica dei caratteri ISO 10646/UNICODE
- Un documento XML è "leggibile" da un utente umano senza la mediazione di software specifico

XML: caratteristiche / 3

- XML è indipendente dal tipo di piattaforma hardware e software su cui viene utilizzato
- XML permette la rappresentazione di qualsiasi tipo di documento (e di struttura testuale) indipendentemente dalle finalità applicative
- XML è indipendente dai dispositivi di archiviazione e visualizzazione
 - un documento XML può essere archiviato su qualsiasi tipo di supporto digitale
 - un documento XML può essere visualizzato su qualsiasi dispositivo di output

XML: caratteristiche / 4



XML: sintesi principi fondamentali

- XML adotta un paradigma di codifica dichiarativo e descrittivo
- XML descrive un documento come una struttura ad albero
- XML introduce il concetto di "tipo di documento" e di "sintassi del documento"
- XML si basa su ISO 10646 /UNICODE

Nozioni di base per creare e visualizzare documenti XML

Il concetto di modello

- Prima della codifica di un qualsiasi documento è necessario studiarne la **natura**, le **caratteristiche** e le possibili **funzionalità**
- In questa fase perciò scegliamo non solo *come* ma *che cosa* vogliamo rappresentare/codificare
- Dal punto di vista della codifica informatica, questo processo analitico coincide con la creazione di un **modello** del documento fonte
- Questo modello, una specie di matrice, in XML si chiama *tipo di documento*, Document Type Definition

Il concetto di tipo di documento

- Un'applicazione XML si basa su un determinato tipo di documento
- Un tipo di documento descrive le caratteristiche di una classe di documenti strutturalmente omogenei
- Il tipo di documento è il fondamento della sintassi e della semantica di una applicazione XML

La Document Type Definition

- Una DTD è costituita da un elenco di dichiarazioni (**markup declaration**) che descrivono la struttura del documento
- Le dichiarazioni di una DTD definiscono:
 - gli elementi strutturali (**element**) di un documento mediante un identificatore generico
 - il modello di contenuto di ogni elemento (**content model**) ovvero gli elementi che contiene e i loro rapporti (un elemento può essere vuoto)
 - la lista degli **attributi** associati a ciascun elemento e il loro tipo

Gli ingredienti necessari

- Si può generare il file .dtd con qualunque editor di testo (TextPad va benissimo)
- Gli elementi da inserire sono:
 - Elemento radice
 - Elementi figli
 - Attributi
 - Indicatori di occorrenza

Elemento radice

- Si tratta dell'elemento che dovrà contenere tutti gli altri, quello che apparirà in cima all'albero di codifica XML nel documento
- Nella TEI l'elemento radice si chiama "TEI.2" e contiene i diversi elementi figli (TeiHeader, Text, ecc)
- Nell'esempio seguente l'elemento radice è:
 - `<!ELEMENT libro (introduzione*, corpo+, epilogo*, indice*)>`

```
<!ENTITY % Contenuti "titolo?, autore?, para+, firma*">
<!ELEMENT libro (introduzione*,corpo+, epilogo*, indice*)>
<!ELEMENT introduzione (%Contenuti;)>
<!ELEMENT corpo (titolo*,(poesia|prosa)+)>
<!ELEMENT epilogo (%Contenuti;)>
<!ELEMENT indice ANY>
<!ELEMENT titolo (#PCDATA)>
<!ELEMENT autore (#PCDATA)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT poesia (titolo?,strofa+)>
<!ELEMENT strofa (verso+)>
<!ELEMENT verso (#PCDATA)>
<!ELEMENT prosa (#PCDATA)>
<!ELEMENT firma (#PCDATA)>
```

Entità parametrica

Modello di contenuto

Elemento radice

Caratteri di testo

Indicatori di occorrenza e connettori

- **Indicatori di occorrenza**
 - Punto interrogativo (?), zero o una occorrenza
 - Segno più (+), una o più occorrenze;
 - Asterisco (*), zero, una o più occorrenze
- **Connettori**
 - Virgola (,), indica sequenza degli elementi
 - Barra verticale (|), indica alternanza degli elementi.

La codifica del documento

Mettere tutto insieme

- Una volta definita la DTD si passa alla codifica del documento
- Nell'esempio propongo la codifica di una raccolta di poesie con tag inventati

```

1 <?xml version="1.0"?>
2 <!DOCTYPE libro SYSTEM 'test.dtd'>
3 <?xml-stylesheet type="text/xsl" href="test.xsl"?>
4 <libro>
5   <introduzione>
6     <titolo>Introduzione</titolo>
7     <para>In questo testo....</para>
8     <para>Si deduce che l'autore....</para>
9     <firma/>
10  </introduzione>
11 <corpo>
12   <titolo>Titolo raccolta</titolo>
13   <titolo>Sottotitolo raccolta</titolo>
14 <poesia>
15   <titolo>0de a XML</titolo>
16   <strofa>
17     <verso metro="end" id="v1.1">0 Xml...</verso>
18     <verso metro="end">sdijfsc</verso>
19     <verso metro="end">cdwfdsc</verso>
20     <verso metro="end" id="id1">dsvgc</verso>
21   </strofa>
22   <strofa>
23     <verso metro="end" id="v1.2">0 Xml...</verso>
24     <verso metro="end">sdijfsc</verso>
25     <verso metro="end">cdwfdsc</verso>
26     <verso metro="end" id="id2">dsvgc</verso>
27   </strofa>
28   <strofa>
29     <verso metro="end" id="v1.3">0 Xml...</verso>
30     <verso metro="end">sdijfsc</verso>
31     <verso metro="end">cdwfdsc</verso>
32     <verso metro="end" id="voihifo">dsvgc</verso>
33   </strofa>
34   <strofa>
35     <verso metro="end" id="v1.3">0 Xml...</verso>
36     <verso metro="end">sdijfsc</verso>
37     <verso metro="end">cdwfdsc</verso>
38     <verso metro="end">dsvgc</verso>
39   </strofa>
40 </poesia>
41 <poesia>
42   <titolo>0de a XSL</titolo>
43   <strofa>
44     <verso metro="end" id="v1.5">0 xsl...</verso>
45     <verso metro="end">tu sei la mia certezza</verso>
46     <verso metro="end">tu dono al testo la bellezza</verso>
47     <verso metro="end">codificare è come ricevere una carezza</verso>
48   </strofa>
49   <strofa>

```

<pre> <!ENTITY % Contenuti "titolo?, autore?, para+, firma*"> <!ELEMENT libro (introduzione*,corpo+, epilogo*, indice*)> <!ELEMENT introduzione (%Contenuti;)> <!ELEMENT corpo (titolo*,(poesia prosa)+)> <!ELEMENT epilogo (%Contenuti;)> <!ELEMENT indice ANY> <!ELEMENT titolo (#PCDATA)> <!ELEMENT autore (#PCDATA)> <!ELEMENT para (#PCDATA)> <!ELEMENT poesia (titolo?,strofa+)> <!ELEMENT strofa (verso+)> <!ELEMENT verso (#PCDATA)> <!ELEMENT prosa (#PCDATA)> <!ELEMENT firma (#PCDATA)> </pre>	
<p style="text-align: center;">DTD</p>	<pre> <!DOCTYPE libro SYSTEM 'test.dtd'> <?xml-stylesheet type="text/xsl" href="test.xsl"?> <libro> <introduzione> <titolo>Introduzione</titolo> <para>In questo testo.....</para> <para>Si deduce che l'autore...</para> <firma/> </introduzione> <corpo> <titolo>Titolo raccolta</titolo> <titolo>Sottotitolo raccolta</titolo> <poesia> <titolo>Ode a XML</titolo> <strofa> <verso metro="end" id="v1">0 xml...</verso> <verso metro="end">sdijfso</verso> <verso metro="end">cdvfd</verso> <verso metro="end" id="idl">dsvg</verso> </strofa> </poesia> </corpo> </libro> </pre> <p style="text-align: center;">File codifica XML</p>

La sintassi XML

Validità e buona formazione

Aspetti di sintassi generale

- I nomi di elementi, attributi e entità sono sensibili alla differenza tra maiuscolo e minuscolo
- Il mark-up è separato dal contenuto testuale mediante caratteri speciali:
 - < > &
- Tali caratteri speciali non possono comparire come contenuto testuale e devono essere eventualmente sostituiti mediante i riferimenti a entità
 - < > &

> sta per "greater than" ovvero >

< sta per "less than" ovvero <

Vincoli di buona formazione

- Esiste un solo elemento radice
- Tutti gli elementi non vuoti devono presentare sia il tag iniziale sia il tag finale
- Tutti gli elementi devono essere correttamente annidati
- Tutti i valori di attributo devono essere racchiusi tra apici doppi o singoli

La codifica degli elementi

- Sintassi di un elemento



I quattro errori comuni

Spesso codificando in XML si può cadere in questi errori:

- 1. **Omettere i tag di chiusura:** ogni tag va aperto e chiuso
 - `<p>Oggi c'è il sole` (sbagliato)
 - `<p>Oggi c'è il sole</p>` (esatto)

- 2. Dimenticare che **XML è sensibile alle maiuscole e minuscole**:
 - `<PersName>Daniele</persname>` (sbagliato)
 - `<PersName>Daniele</PersName>` (esatto)

- 3. Inserire gli **spazi nel nome dell'elemento**:
 - <Pers Name> (sbagliato)
 - <PersName> (esatto)

- 4. Dimenticare le **virgolette per i valori degli attributi**:

- `<note place=foot>` (sbagliato)
- `<note place="foot">` (esatto)