A mamma e babbo

Indice

1 INTRODUZIONE

1.1 Presentazione del progetto *piccola Biblioteca Digitale Romanza*

Nel corso di questo lavoro si è preso in esame il progetto *piccola Biblioteca Digitale Romanza*, che consiste appunto in una raccolta di testi di vario tipo digitalizzati grazie all'utilizzo della codifica TEI XML. La piccola BDR permette di condividere le conoscenze e il sapere grazie al fatto di essere uno strumento ad accesso libero e gratuito. I testi presenti in formato elettronico possono essere consultati in qualsiasi momento e, attraverso software ideati per gli studi umanistici, l'utente può interrogarli ed usufruire dei vantaggi che possono dare, sia nel campo della didattica sia in quello della ricerca. "Nella piccola BDR vengono rispettati i criteri di tutte le biblioteche digitali, ovvero: standard di qualità certificati per i documenti archiviati; strategie di preservazione delle risorse digitali; modalità di distribuzione; strumenti di accesso per gli utenti".[1]

1.2 Come funziona

La piccola BDR prevede la presenza di quattro "macrosezioni": strumenti, poesia, teatro e narrativa. Al momento solo nelle prime due sezioni è presente del materiale, ma è in corso una ricerca e un lavoro per reperire e codificare documenti anche per le altre due. Qualsiasi utente può leggere i testi, salvarli in formato pdf, stamparli ed eseguire ricerche, dalle più semplici, come la ricerca di una semplice parola o di una breve frase, fino ai periodi più complessi e ricerche più dettagliate "sia sull'intero corpus ma anche su una rosa di testi selezionati ad hoc",[2]come figure retoriche, parti in lingue straniere e così via. Tutto questo è possibile grazie all'utilizzo di strumenti

[1] *http://piccolabdr.humnet.unipi.it/*
[2] *Ibidem*

di codifica XML, secondo lo schema TEI Lite, e all'uso di software di ricerca *full-text* come Philologic. I testi codificati sono in formato XML, strumento molto elastico, infatti è possibile ampliare o modificare la codifica fatta originariamente. Ogni utente registrato può arricchire o correggere la codifica dei testi, magari usufruendo di una raccolta di strumenti necessari. Ognuno può quindi realizzare un vero laboratorio editoriale in continua evoluzione, condivisibile con altri studiosi.

1.3 Sezioni

1.3.1 Sezione strumenti

Attraverso questa sezione, un utente è in grado di usufruire degli strumenti necessari per un utilizzo facilitato e per una ricerca avanzata. La ricerca sarà convogliata principalmente sui documenti più rari, rintracciabili esclusivamente in pochissime biblioteche. Esistono anche documenti che, trovandosi solo in formato immagine, sono inadatti ad una ricerca dettagliata e permettono la sola lettura. Ma, grazie a questo strumento, verranno digitalizzati e messi in rete e a disposizione di chiunque voglia consultarli ed esaminarli.[3]

1.3.2 Sezione poesia

Questa è la prima sezione di raccolta testi. Vi si può trovare una raccolta di poesie del poeta francese Remy Belleau, *Les Petites Inventions*, che non hanno mai avuto un'edizione propria. È stata pubblicata dopo le *Odes d'Anacréon*, una sorta di appendice per tutte le opere del poeta. Il primo problema dell'editore di questa collezione è di cercare di ricostituire l'intera raccolta per costruire la genesi de *Les Petites Inventions*.[4]

[3] *Ibidem*
[4] *Ibidem*

1.4 Lavoro svolto

Lo scopo di questo lavoro di tesi è stato aggiornare la codifica TEI Lite P4 dei testi già esistenti migrando alla versione P5 più recente. Poi si è passati ad un arricchimento della raccolta, codificando e aggiungendo nuovi brani di nuovi autori. Infine si è fatta un'analisi del software di ricerca adottato fin'ora nella piccola BDR (**Philologic**) e di quello che eventualmente lo sostituirà in futuro (**eXist**).
Vedremo in dettaglio nei prossimi capitoli.

2 XML (*eXtensible Markup Language*)

XML è un linguaggio di markup basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo.

XML può essere indicato anche come la versione semplificata di SGML[5] che vuole definire nuovi linguaggi di markup da usare in ambito web. È definito estensibile in quanto permette di aggiungere elementi personalizzati.

2.1 Storia

Negli anni novanta Microsoft e Netscape introducevano un'estensione proprietaria all'HTML ufficiale. In seguito a questa situazione, chiamata "la guerra dei browser", il *World Wide Web Consortium* (W3C)[6] fu costretto a seguire le individuali estensioni al linguaggio HTML e dovette decidere quali caratteristiche standardizzare e quali lasciare fuori.

Fu in questo contesto che iniziò a delinearsi la necessità di un linguaggio di markup che desse più libertà nella definizione dei tag, ma rimanendo sempre in uno standard.

Negli anni novanta ebbe inizio quindi il "progetto XML" che suscitò un così forte interesse che la W3C creò un gruppo di lavoro chiamato XML *Working Group,* composto da esperti mondiali delle tecnologie SGML e da una commissione con il compito di stendere le specifiche del progetto.

[5] Lo ***S****tandard* ***G****eneralized* ***M****arkup* ***L****anguage* (SGML), è un metalinguaggio definito come standard ISO (ISO 8879:1986 SGML) avente lo scopo di definire linguaggi da utilizzare per la stesura di testi destinati ad essere trasmessi ed archiviati con strumenti informatici, ossia per la stesura di documenti in forma leggibile da computer (*machine readable form*).(*http://it.wikipedia.org/wiki/Standard_Generalized_Markup_Language*)

[6] Nell'ottobre del 1994 Tim Berners Lee, padre del Web, fondò al MIT (*Massachusetts Institute of Technology*), in collaborazione con il CERN (il laboratorio dal quale proveniva), un'associazione di nome *World Wide Web Consortium* (abbreviato W3C), con lo scopo di migliorare gli esistenti protocolli e linguaggi per il World Wide Web e di aiutare il web a sviluppare tutte le sue potenzialità.(*http://it.wikipedia.org/wiki/World_Wide_Web_Consortium*)

Così, nel 1998, queste specifiche divennero ufficiali con il nome di *Extensible Mark-up Language*, versione 1.0[7], strumento che permetteva di essere utilizzato in diversi contesti, dalla rappresentazione di immagini alla definizione di formati di dati, dalla composizione della struttura di documenti allo scambio delle informazioni tra sistemi diversi.

2.2 Utilizzo

A differenza dell' HTML, l'XML ha uno scopo ben diverso: è un metalinguaggio.[8]HTML ha un insieme di elementi ben definito, mentre con XML è possibile crearne nuovi a seconda delle proprie esigenze.

2.3 Sintassi

Un esempio della sintassi tipica dei file XML è la seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
<utenti>
    <utente>
        <nome>Luca</nome>
        <cognome>Ricci</cognome>
        <indirizzo>Milano</indirizzo>
    </utente>
    <utente>
        <nome>Max</nome>
        <cognome>Rossi</cognome>
        <indirizzo>Roma</indirizzo>
    </utente>
```

[7] *http://it.wikipedia.org/wiki/XML*

[8] Per metalinguaggio si intende un linguaggio formalmente definito che ha come scopo la definizione di altri linguaggi artificiali. *(http://it.wikipedia.org/wiki/Metalinguaggio)*

</**utenti**>

Nella prima riga si indica la versione di XML in uso e si specifica la codifica UTF-8[9] per la corretta interpretazione dei dati.

I caratteri speciali che renderebbero il documento mal formato vanno sostituiti con le rispettive entità[10] XML:

Carattere	**Entità**
&	&amp;
<	&lt;
>	&gt;
“	&quot;
’	&apos;

2.3.1 Gli elementi

Similmente all'HTML, anche l'XML utilizza dei marcatori: un tag di apertura e uno di chiusura.

Essendo un linguaggio molto rigido sulla sintassi da seguire (al contrario dell'HTML) è necessario rispettare alcune regole:

1. i tag non possono iniziare con numero o con caratteri speciali e non possono contenere spazi;

2. i tag di apertura e di chiusura devono essere bilanciati, ovvero non sono consentiti errori di annidamento (*es.* <**lg**><**l**>Une femme est encor trop sage.</**l**></**lg**> e NON <**lg**><**l**>Une femme est encor trop sage.</**lg**></**l**>);

[9] UTF-8 (*Unicode Transformation Format*, 8 bit) è una codifica dei caratteri *Unicode* in sequenze di lunghezza variabile di *byte*. (*http://it.wikipedia.org/wiki/UTF-8*)

[10] In linguaggi di markup quali HTML, XML e derivati, le entità (in inglese *entity*) sono una codifica testuale usata per inserire alcuni caratteri speciali in maniera indipendente dalla tastiera e dal sistema operativo usato. La loro forma generale è: "&" + codice identificativo + ";". (*http://it.wikipedia.org/wiki/Entit%C3%A0_(markup)*)

3. se il documento XML non contiene errori si dice *well formed* ('ben formato'). Se il documento è *well formed* e rispetta i requisiti strutturali definiti nel file DTD[11] o schema XML associato viene detto Valid (valido).

Alcuni esempi:

1. In questo caso l'elemento <**cognome**> non è stato chiuso, quindi il file risulta non ben formato:

<**rubrica**>
    <**nome**>Alessandro</**nome**>
    <**cognome**>Bianchi
</**rubrica**>

2. L'elemento <**cognome**> viene chiuso dopo quello del tag </**rubrica**> e anche qui il file non è ben formato:

<**rubrica**>
    <**nome**>Alessandro</**nome**>
    <**cognome**>Bianchi
</**rubrica**></**cognome**>

3. XML è case sensitive[12] , quindi <**cognome**> e <**COGNOME**> vengono considerati come due elementi diversi e l'XML è non valido:

<**rubrica**>
    <**nome**>Alessandro</**nome**>
    <**cognome**>Bianchi</**COGNOME**>

[11] Il *Document Type Definition* (definizione del tipo di documento) è uno strumento utilizzato dai programmatori il cui scopo è quello di definire le componenti ammesse nella costruzione di un documento XML. (*http://it.wikipedia.org/wiki/Document_Type_Definition*)

[12] L'applicazione fa distinzione tra le lettere maiuscole e quelle minuscole. *(http://www.pc-facile.com/glossario/case_sensitive/)*

</**rubrica**>

Questa è la forma corretta:

<**rubrica**>
  <**nome**>Alessandro</**nome**>
  <**cognome**>Bianchi</**cognome**>
</**rubrica**>

Gli elementi possono essere anche definiti vuoti e quindi vengono chiusi immediatamente:

<**rubrica**></**rubrica**>

Un altro modo di scrivere un elemento vuoto è quello abbreviato:

<**rubrica**/>

Un file XML, per essere interpretato correttamente da un browser, deve essere fornito delle seguenti caratteristiche:

- un **prologo**, prima istruzione che appare scritta nel documento: <?xml version="1.0" encoding="UTF-8"?>;
- un unico **elemento radice** (il nodo principale chiamato root element) che contiene tutti gli altri nodi del documento;
- un **foglio di stile**, perché il browser sa interpretare solo (X)HTML.

Tutti **gli elementi** devono essere bilanciati all'interno del documento.

2.4 Alcune tecnologie di supporto a XML

- **DTD** (*Document Type Definition*): attraverso una serie di regole grammaticali, questo tipo di documento specifica le caratteristiche strutturali di un documento XML. In particolare definisce l'insieme degli elementi utilizzabili nel documento XML, le relazioni gerarchiche tra gli elementi, l'ordine di apparizione nel documento XML e quali elementi e attributi sono opzionali o no.

- **XML Schema**: la sua sigla è XSD (*XML Schema Definition*), serve anch'esso a definire un documento XML. Il W3C consiglia di adottarlo al posto della DTD stessa, essendo più recente ed avanzato.

- **XSL** (*eXtensible Stylesheet Language*): linguaggio con cui si descrive il foglio di stile di un documento XML. È composto a sua volta da due linguaggi: XSLT (la T sta per *Transformation*) e XSL-FO (FO sta per *Formatting Objects*).

- **XQuery**: linguaggio di query concepito per essere applicabile a qualsiasi sorta di documento XML. Ha funzionalità che consentono di poter attingere da fonti di dati multiple per la ricerca, per filtrare i documenti o riunire i contenuti di interesse.

2.5 (X)HTML

(X)HTML è un linguaggio di markup utilizzato per visualizzare pagine Web tramite browser (come per l'HTML) e, essendo implementato in XML, ne rispetta la semantica. In questo caso gli elementi vuoti vanno chiusi con uno slash finale (/), gli attributi vuoti vanno perfezionati con **true** e **false**. Alcuni elementi e attributi sono scomparsi rispetto all'HTML 4.0 ed esiste una DTD dedicata.

Il vantaggio di una pagina (X)HTML è che presenta tutti i pregi dell'XML, come

la validazione facilitata e un'interpretazione schematica. Le pagine sono più accessibili grazie alla semantica XML.[13]

[13] *http://it.wikipedia.org/wiki/XML*

3 TEI

La ***Text Encoding Initiative*** è un consorzio di istituzioni internazionali, di ambito linguistico e letterario, che ha sviluppato uno standard per la rappresentazione dei testi in formato digitale.

Il suo scopo è quello di creare delle linee guida di alta qualità per la codifica di documenti umanistici e per far si che chiunque, istituzioni o singoli individui, possa usufruirne.

3.1 Breve storia[14]

La TEI è stata istituita nel 1987. Grazie alla documentazione per gli schemi di codifica, le *Guidelines for Electronic Text Encoding and Interchange*, la TEI definisce un linguaggio di markup (in XML) per la digitalizzazione di testi, utile soprattutto per creare collezioni, archivi e banche di dati testuali.

I benefici che porta la codifica di testi sono molteplici:

- portabilità dei testi;
- facilità di archiviazione;
- facilità di gestione attraverso gli strumenti informatici.

I principi fondamentali su cui è stabilito il consorzio TEI sono:

- Gratuità delle linee guida, DTD e altre documentazioni;
- Partecipazione alle attività TEI aperta a tutti gli utenti;
- Rendere il consorzio stesso organo di rappresentazione a livello internazionale.

[14] *http://it.wikipedia.org/wiki/Text_Encoding_Initiative#Le_linee_guida_del_progetto*

3.1.1 Il progetto

Il primo progetto creato, P1 (la P sta per *proposta*), risale al 1990. Tra il 1990-1993 si riunirono altri 15 gruppi di lavoro e creano un aggiornamento della precedente versione, arrivando alla P2 e includendo notevoli quantità di nuovi materiali.

Nel 1994 viene rilasciata la prima versione ufficiale denominata P3.

In seguito all'adozione di questo metalinguaggio standard, divenne necessario aggiornare le linee guida TEI per renderle compatibili con questo nuovo formalismo.

La versione P4 è stata pubblicata nel 2002 con poche modifiche essenziali ai vincoli espressi negli schemi e con qualche correzione della precedente versione.

Sono iniziati subito i lavori per una versione P5, progetto che è stato pensato come una completa revisione delle versioni precedenti, con lo sviluppo di una nuova serie di settori fondamentali non trattati precedentemente, tra cui la codifica dei caratteri, la grafica, la descrizione manoscritta, biografica e geografica dei dati. Quest'ultima versione è stata rilasciata il 1 novembre 2007.

3.2 Le linee guida

Le linee guida TEI danno indicazioni sui metodi di rappresentazione adeguati, in modo da mettere in evidenza determinate caratteristiche di un testo e facilitarne l'elaborazione su un computer, indipendentemente dalla piattaforma utilizzata.

Vengono specificati una serie di elementi da inserire nel testo, per contrassegnare il documento. Utile per facilitare lo scambio dei dati tra utenti o per gruppi di ricerca che usano programmi diversi. Si possono applicare a qualsiasi tipo di testo.

Le principali caratteristiche:

- devono essere semplici, chiare e concrete;
- devono essere di semplice utilizzo da parte degli utenti, senza il bisogno di ricorrere a software specializzati;

- devono permettere una rigorosa definizione e un efficiente elaborazione dei testi;
- devono essere conformi agli standard esistenti o in procinto di essere adottati;
- devono consentire estensioni definite dall'utente.

Richiedono un continuo sviluppo e ricerca poiché cercano di descrivere un dominio testuale che ancora è oggetto di studio e in costante evoluzione. Tutto ciò porta alla creazione di elementi sempre nuovi per marcare caratteristiche che in precedenza sono state tralasciate e per aggiornare i moduli preesistenti.[15]

Il consorzio mette a disposizione gli strumenti e le istruzioni adatti a migrare da P4 a P5, oltre a documentazione come *wiki* e *tutorial*.

3.3 TEI Lite

La TEI Lite è un sottoinsieme degli schemi TEI.

Permette la creazione, semplice e rapida, di documenti compatibili con l'intero schema TEI. Per fare ciò è stato necessario prefissare degli obiettivi, elencati di seguito:[16]

- Includere la maggior parte dei marcatori fondamentali TEI;
- poter trattare in modo adeguato il maggior numero di tipologie di testi;
- poter essere utilizzabile con la maggior parte dei software XML esistenti;
- essere derivabile dalle DTD TEI, escludendo gli elementi in base alle descrizioni delle linee guida;
- essere conciso e semplice.

[15] *Ibidem*
[16] *Ibidem*

3.4 La TEI oggi[17]

La TEI, al giorno d'oggi, è riconosciuta come uno strumento di fondamentale importanza a livello internazionale per la conservazione a lungo termine dei dati elettronici. Strumento scelto come schema di codifica per documenti scientifici e letterari, per la gestione e produzione di metadati associati a testi elettronici.

Grazie a questo tipo di codifica si è aperta la strada per una migliore conservazione e distribuzione del nostro patrimonio culturale e tutto ciò potrà essere messo a disposizione di studenti, ricercatori, studiosi ma anche di persone comuni.

[17] *Ibidem*

4 Codifica dei testi della piccola BDR

4.1 Codifica originaria in TEI Lite P4

4.1.1 Struttura di un testo TEI

Inizialmente la codifica adottata per i testi della piccola Biblioteca Digitale Romanza fu quella della TEI P4, la penultima delle versioni uscite fin'ora.

La struttura di un testo in P4 contiene un'intestazione (marcata con l'elemento <teiHeader>) e la trascrizione del testo vero e proprio (marcata con l'elemento <text>).

L'intestazione è importante perché contiene le informazioni riguardanti il documento codificato. Può essere composta da 4 diverse parti che contengono una descrizione bibliografica del testo, una descrizione del modo in cui è stato codificato, una descrizione non bibliografica e un elenco delle varie revisioni.

Può essere composto da un unico testo oppure da una collezione di più testi. Sia nel primo che nel secondo caso c'è la possibilità di aggiungere un peritesto iniziale o finale. Nel mezzo si troverà il corpo del testo o dei testi nel caso di una collezione.

Una struttura generale di un documento composto da un unico testo può essere la seguente:

```
<TEI.2>
<teiHeader> [informazioni dell'intestazione TEI] </teiHeader>
   <text>
     <front> [materiali del peritesto iniziale] </front>
     <body> [testo unitario] </body>
     <back> [materiali del peritesto finale] </back>
   </text>
<TEI.2>
```

Un documento composto da un gruppo o più gruppi di testi sarà codificato con una struttura globale come la seguente:

```
<TEI.2>
<teiHeader> [intestazione del testo composito] </teiHeader>
    <text>
     <front> [peritesto iniziale del testo composito] </front>
     <group>
     <text>
       <front>[peritesto iniziale del primo testo] </front>
       <body> [primo testo unitario] </body>
       <back> [peritesto finale del primo testo] </back>
     </text>
     <text>
       <front>[peritesto iniziale del secondo testo] </front>
       <body> [secondo testo unitario] </body>
       <back> [peritesto finale del secondo testo] </back>
    </text> [ altri testi o gruppi di testi] </group>
 <back> [peritesto finale del testo composito] </back>
  </text>
<TEI.2>
```

C'è anche la possibilità di definire una collezione di testi TEI (TEI corpus) ciascuno con la propria intestazione:

```
<teiCorpus.2>
<teiHeader> [intestazione del corpus] </teiHeader>
<TEI.2>
  <teiHeader> [intestazione del primo testo] </teiHeader>
     <text> [primo testo nel corpus] </text>
</TEI.2>
```

<TEI.2>

<teiHeader> [intestazione del secondo testo] </teiHeader>

<text> [secondo testo nel corpus]</text>

[...]

</TEI.2>

</teiCorpus.2>

4.1.2 Peritesto iniziale

Come già anticipato, la parte iniziale del documento (frontespizio, prefazione...) è di fondamentale importanza in quanto fornisce delle informazioni base di tipo linguistico e sociale utilissime per la finalità di ricerca.

1. *FRONTESPIZIO:* di seguito gli elementi che caratterizzano una buona codifica di un frontespizio:

- **<titlePage>:** contiene la pagina di frontespizio di un testo;
- **<docTitle>:** indica il titolo di un documento;
- **<titlePart>:** contiene una suddivisione del titolo di un'opera, come appare sul frontespizio;
- **<docAuthor>:** indica il nome dell'autore del documento;
- **<docDate>:** indica la data di edizione del documento;
- **<docEdition>:** contiene la dichiarazione dell'edizione;
- **<docImprint>:** contiene la dichiarazione delle note tipografico-editoriali;
- **<eprigraph>:** contiene una citazione che appare all'inizio di una sezione o capitolo.

Esempio:

```
<front>
<head type="titleRecueil"> Le Theâtre Italien </head>
<head type="subtitleRecueil"> de Gherardi</head>
<head type="subtitleSub">TOME I</head>
  <titlePage>
    <docTitle>
      <titlePart type="main">LE BANQUEROUTIER,</titlePart>
      <titlePart type="subt" rend="italics">COMEDIE EN TROIS ACTES.</titlePart>
    </docTitle>
  </titlePage>
</front>
```

2. IL FRONTESPIZIO ELETTRONICO: fornisce informazioni descrittive e dichiarative. Precede qualsiasi testo TEI-conforme :

- **<teiHeader>:** introduce l'intestazione;
- **<fileDesc>:** descrizione bibliografica del documento digitale;
- **<encodingDesc>:** documenta la relazione fra un documento elettronico e la fonte/le fonti da cui è derivato.;
- **<profileDesc>:** contiene una descrizione dettagliata degli aspetti non bibliografici del documento, come le lingue e i dialetti usati, i partecipanti e l'ambiente in cui sono si sono svolte le interviste per la costruzione della lingua parlata;
- **<revisionDesc>:** riassume la storia delle revisioni del documento elettronico.

3. L'elemento **<fileDesc>** è obbligatorio e contiene i seguenti elementi :

- **<titleStmt>**(nota stmt sta per statement): raggruppa le informazioni su titolo dell'opera;
- **<editionStmt>:** raggruppa le info relative ad una data edizione del testo;
- **<publicationStmt>:** raggruppa le info relative alla pubblicazione e distribuzione del testo;
- **<sourceDesc>:** fornisce una descrizione bibliografica del testo da cui è stato tratto il testo elettronico.

Esempio di <teiHeader>:

```
<teiHeader>
<fileDesc>
<titleStmt>
<title>Le Theâtre Italien: edizione elettronica</title>
<author>
<name type="forename">Evaristo</name>
<name type="surname">Gherardi</name>
</author>
<respStmt>
<resp>Digitalizzazione e Codifica elettronica a cura di</resp>
<name>Roberto Rosselli Del Turco</name>
<name>Barbara Sommovigo</name>
<name>Elisa Ledda</name>
</respStmt>
</titleStmt>
<editionStmt>
<edition>Prima edizione elettronica
<date>2012</date>
</edition>
</editionStmt>
<publicationStmt>
<publisher>Università degli studi di Pisa; Corso di Studi in Informatica Umanistica;</publisher>
<availability status="restricted"> <p>Questo progetto è protetto dal diritto d'autore</p> </availability>
<date>2012</date>
</publicationStmt>
<sourceDesc>
```

```
<biblFull>
<titleStmt>
<title type="main">Le Theâtre Italien</title>
<author>Evaristo Gherardi</author>
</titleStmt>
<editionStmt>
<edition>I edizione</edition>
</editionStmt>
<extent/>
<publicationStmt>
<publisher>J.-B. Cusson e P. Wiite</publisher>
<pubPlace>Parigi</pubPlace>
<date>1700</date>
</publicationStmt>
<seriesStmt>
<title> Le Theâtre Italien de Gherardi, [...]</title>
</seriesStmt>
</biblFull>
</sourceDesc>
</fileDesc>
<encodingDesc>
<projectDesc> <p> Al momento della codifica, quando si è deciso come il testo doveva [...].</p>
</projectDesc>
<editorialDecl> <p> Grazie alla codifica del testo nel formato TEI XML è [...]. </p> </editorialDecl>
<refsDecl> <p> I tag utilizzati nella seguente codifica servono per [...].</p> </refsDecl>
</encodingDesc>
<profileDesc>
<creation>
<date when="xx-xx-2012">2012</date>
</creation>
<langUsage>
<language ident="FR">Francese</language>
<language ident="LA">Latino</language>
</langUsage>
</profileDesc>
<revisionDesc>
<change>
<date when="xx-xx-2012">2012</date>
<name>Barbara Sommovigo</name>
<desc>correzione e normalizzazione in base al testo di riferimento: <bibl>
<author>Evaristo Gherardi</author>
```

```
        <title>Le Theâtre Italien</title>
        <publisher/>
        <date/>
        </bibl>
        </desc>
    </change>
    <change> [...] </change>
    <change> [...] </change>
  </revisionDesc>
  </teiHeader>
```

4.1.3 Il titolo

1. <**titleStms**> può contenere:

- <**title**>**:** contiene il titolo di un'opera;
- <**author**>**:** contiene il nome dell'autore di un'opera e costituisce la dichiarazione di responsabilità primaria per ogni unità bibliografica.

Esempio:

```
<titleStmt>
  <title type="main"> Le Theâtre Italien</title>
  <author>Evaristo Gherardi</author>
</titleStmt>
```

2. L'elemento <**respStmt**> contiene i seguenti sotto-componenti:

- <**resp**>**:** contiene una frase che descrive la natura della responsabilità intellettuale di una persona;
- <**name**>**:** il nome della persona responsabile della cura dell'edizione.

Esempio:

<**respStmt**>
  <**resp**>Digitalizzazione e Codifica elettronica a cura di </**resp**>
  <**name**>Roberto Rosselli Del Turco</**name**>
  <name>Barbara Sommovigo</name>
  <name>Elisa Ledda</name>
</**respStmt**>

3. Nell'area dell'**edizione** vengono date informazioni relative alla data dell'edizione di un testo.

Esempio:

<**editionStmt**>
  <**edition**>Prima edizione elettronica
    <**date**>2012</**date**>
  </**edition**>
</**editionStmt**>

4. Nell'area della **pubblicazione** vengono date informazioni relative all'organizzazione responsabile (<**publisher**>), al nome di una persona responsabile della distribuzione del documento (<**distributor**>), al luogo e alla data di pubblicazione (<**pubPlace**>, <**date**>).

Esempio:

```
<publicationStmt>
   <publisher>Università degli studi di Pisa;</publisher>
  <availability status="restricted">Questo progetto è protetto dal diritto d'autore</availability>
   <date>2012</date>
</publicationStmt>
```

4.2 Elementi base della codifica

Un semplice documento TEI è composto dai seguenti elementi base:

- **<front>:** contiene il materiale che precede il testo vero e proprio (frontespizio, dediche, prefazioni...);
- **<group>:** raggruppa un insieme di testi unitari o di gruppi di testi;
- **<body>:** contiene l'intero corpo del testo;
- **<back>:** contiene l'appendice che segue il testo.

4.3 Elementi per partizioni testuali

Un testo in prosa, come nel caso di alcuni documenti contenuti nella piccola BDR, è composto da una serie di paragrafi che eventualmente si possono trovare anche raggruppati insieme in capitoli, sezioni, sottosezioni.

I paragrafi semplici vengono marcati con l'elemento **<p>**, mentre l'elemento ** ** viene utilizzato per contenere una sezione del peritesto o del corpo di un testo, che può essere a sua volta diviso in sottosezioni (**<div1>**, **<div2>**...) che possono arrivare fino a sette. Se sono necessarie più di sette sottosezioni è possibile modificare la DTD TEI, oppure utilizzare il più generico ** ** senza aver bisogno di specificare il numero.

Ognuno di questi elementi può contenere degli attributi:

- **type:** indica il nome convenzionale per questa categoria (es. Libro, Capitolo...);
- **id:** specifica un indicatore unico per la sezione;
- **n:** specifica un numero per la sezione, a volte si preferisce a id.

Esempio:

```
<div type="scene" n="1">
  <sp>
    <speaker rend="center">MEZZETIN.</speaker>
    <p n="1">D'où viens-tu-, mon Amy ?</p>
  </sp>
</div>
```

4.3.1 Prosa, versi e testi drammatici

Il corpus di cui ci siamo occupati è composto da testi eterogenei. Infatti in un unico testo è possibile trovare dei versi in poesia alternati a parti in prosa.

Per distinguere al meglio le diverse sezioni si ricorre a degli elementi specifici:

- **<l>:** contiene una riga di poesia;
- **<lg>:** contiene un gruppo di versi;
- **<sp>:** contiene una singola battuta in un testo drammatico;
- **<speaker>:** contiene un'etichetta che specifica il nome di uno o più parlanti nel testo;
- **<stage>:** contiene delle didascalie o direttive di scena all'interno di un testo.

Esempio:

```
<sp>
  <speaker rend="center">LE DOCTEUR</speaker>
 <stage type="business" rend="italics">suivy de plusieurs Archers arrive, [...]</stage>
    <lg>
      <l>Pour vivre heureux, bis.</l>
      <l>N'ayez pour objet de vos vœux</l>
      <l>Que les ris et les jeux.</l>
      <l>[...]</l>
   </lg>
</sp>
```

4.3.2 Numeri di pagine e di riga

Per le interruzioni di pagina e di linea si sono usati i seguenti elementi vuoti:

- **<pb>:** indica il passaggio da una pagina all'altra del testo originale (*es.* <**pb n="518"**/>);
- **<lb>:** indica l'inizio di una nuova riga tipografica.

4.3.3 Elementi più specifici utilizzati

- **<emph>:** indica parole messe in risalto per uso linguistico o retorico (*es.* <**emph**>**è meglio un ottimo Arlecchino in libertà che un filosofo alla Pastiglia**</**emph**>);
- **<foreign>:** indica un'espressione o una parola che appartiene a una lingua diversa da quella del resto del testo (*es.* <**foreign xml:lang="latino" rend="italics"**>**Omnia bona mea mecum porto.**</**foreign**>);
- **<title>:** contiene il titolo di una qualsiasi opera.

4.4 Nuova codifica dei testi della piccola BDR in P5

In seguito alla creazione di una nuova versione della TEI, si è resa necessaria una migrazione della vecchia codifica dei testi già esistenti nella piccola BDR (codificati appunto in TEI Lite P4) alla nuova TEI P5.

La migrazione è avvenuta grazie ad un foglio di stile .xsl, messo a disposizione dallo studioso Yeates nel 2008 (versione più recente), con il quale, mediante una semplice associazione ai file con lo strumento di editor *Oxygen*, è stato possibile trasformare la codifica TEI da P4 a P5.

4.4.1 DTD creata per i documenti in P5

Grazie al *tool* Roma della TEI è stato possibile quindi creare un nuovo schema .dtd apposito che rendesse validi i nuovi testi codificati in P5 e quelli già esistenti dopo

la migrazione alla P5.

Oltre ai moduli base **core**, **tei**, **header**, **textstructure**, sono stati aggiunti nuovi elementi necessari al tipo dei file codificati. Nel nostro caso i testi contengono parti in prosa ma anche parti in versi, quindi è stata preso come base di partenza lo schema di codifica 'TEI with drama'.

Sono stati quindi aggiunti i seguenti moduli[18]:

- **analysis**: *simple analytic mechanisms*;
- **drama**: *performance texts*;
- **linking**: *linking, segmentation and alignment*;
- **namesdates**: *names and dates.*

Per ognuno di questi moduli sono stati poi inclusi gli elementi specifici adatti ai testi a cui poi verrà associato il file .dtd.

4.4.2 Differenze tra P4 e P5

Essendo i testi della piccola BDR finalizzati ad una ricerca semplice e diretta, anche la codifica di conseguenza è stata fatta con gli elementi base descritti nei capitoli precedenti.

Un prima differenza fra P4 e P5 è possibile notarla nell'intestazione del documenti XML.

Intestazione TEI Lite P4

```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE TEI.2 SYSTEM "teixlite.dtd">
  <TEI.2>
    [...]
  </TEI.2>
```

[18] *http://www.tei-c.org/Roma/startroma.php?mode=main*

Intestazione P5

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE TEI SYSTEM "TEI-PBDR.dtd">
  <TEI xmlns="http://www.tei-c.org/ns/1.0">
  [...]
  </TEI>
```

Un altro aspetto che differenzia le due versioni sta nella dichiarazione degli elementi marcatori e un esempio può essere quello dell'utilizzo dell'elemento <choice>[19], mancante nella P4 e utilizzato nella P5:

- **P4:** <corr sic="o">c</corr>h'hoggidì;
- **P5:** <choice>
  <corr>c</corr>
  <sic>o</sic>
  h'hoggidì </choice>

Nei testi successivamente codificati in P5, le entità sono state definite nel file .dtd, mentre in quelli già esistenti e codificati in P4 le entità erano indicate direttamente nel documento .xml, quindi le lettere con accenti e caratteri speciali erano sostituiti dal rispettivo codice. In generale inserire le entità nella DTD è meglio che metterle direttamente in un file XML, ma la soluzione migliore resta quella di un file di entità separato.

Esempi:

- theâtre = theâtre
- vérités = vérités
- & = &

[19]Gruppi di un certo numero di codifiche alternative per lo stesso punto in un testo.

Un'altra differenza sostanziale è la presenza di attributi all'interno dell'elemento **<body>**.

Nella codifica originaria dei testi appartenenti alla collezione già presente nella piccola BDR, *Les Petites Inventions*, si può notare che l'elemento **<body>** presenta in particolare l'attributo **"type"**. Nella nuova versione P5, questo tipo di marcatura non è possibile, in quanto non renderebbe valido il file, per questo l'attributo è stato spostato all'interno di un altro elemento.

Esempi:

- **P4:** <body type="poeme">
  <lg n="I">
- **P5:** <body>
  <lg n="I" type="poeme">

Un altro aspetto che differenzia le due versioni è la scrittura degli attributi **"id"** e **"lang"**. Infatti nel caso della P5 si deve fare riferimento a un namespace e scrivere **xml:id** o **xml:lang**, non id e basta. Il valore dell'attributo inoltre non può essere semplicemente un numero ma deve iniziare con un carattere.

Esempi:

- **P4:** <text **id="PI0819DM" lang="LA"**>
- **P5:** <text **xml:id="PI0819DM" xml:lang="LA"**>

5 I nuovi testi codificati per la piccola BDR

La nuova raccolta che entrerà a far parte della piccola Biblioteca Digitale Romanza sarà quella dell'autore Evaristo Gherardi[20] intitolata *Theâtre Italien ou Recueil de toutes le scenes françoises,* una raccolta di testi teatrali in francese (ma sono presenti anche dei passi in italiano e in latino), sia in prosa che in versi.

La raccolta è composta da sei volumi e i brani codificati nel mio lavoro di tesi fanno parte del primo e del quarto volume e sono delle trascrizioni della prima edizione a stampa del 1700.

5.1 *La Baguette de Vulcain*

La Baguette de Vulcain (in italiano *La Bacchetta di Vulcano*) si trova nel IV volume della raccolta di Gherardi. L'opera è divisa in due parti:

- ***La Baguette de Vulcain***: commedia in un atto e 6 scene;
- ***L'augmentation de la Baguette de Vulcain:*** 3 scene + canti.

Quest'opera, come la maggior parte dei brani che formano il corpus di Gherardi, è composta da parti in prosa e da parti in versi.

5.1.1 Riassunto

La bella Bradamante (Isabella) è prigioniera di un gigante. Viene liberata dal prode cavaliere Roger (Arlecchino) che, grazie ad una bacchetta magica donatagli da Vulcano, uccide il gigante e risveglia la ragazza da un sonno durato 200 anni.

Oltre a trasformare la grotta in un meraviglioso giardino, Roger risveglia anche la sua domestica, Mélisse (Colombina), che ritroverà suo marito.

[20]Nato a Prato verso il 1662 e morto a Parigi il 31 agosto 1700, famoso interprete della maschera di Arlecchino.(*http://biblioteca.accademiadeifilodrammatici.it/biblio/scheda/125725*)

In seguito avverrà l'incontro tra Roger e un druido, famoso per la sua specialità nel donare consigli ai mariti cornuti. Il druido riceverà la visita di Floristan (sospetta che la moglie gli sia infedele) e della coppia Zerbine e Gabrine, angosciati dal fatto che non hanno figli (ma non hanno mai dormito assieme).

L'opera si conclude con canti e danze.

5.2 *Le Banqueroutier*

Le Banqueroutier (in italiano *Il bancarottiere*) si trova nel I volume della raccolta di Gherardi.

Una delle caratteristiche di questo testo è che, nonostante nel titolo sia indicato "*En trois actes*", l'opera non è divisa in atti ma solo in "*SCENES*". Questa commedia presenta un prologo iniziale, il testo centrale e un canto finale. A parte il canto centrale composto in versi, tutto il resto dell'opera è in prosa.

5.3 La codifica

Al momento della codifica di entrambi i testi, quando si è deciso come dovevano essere visualizzati nel web, si è scelto di mettere in risalto gli aspetti narrativi cercando di rimanere il più possibile fedeli alla visualizzazione originale della commedia. È possibile individuare immediatamente le immagini, le parti in versi e quelle in prosa, così come le didascalie e le indicazioni di scena, grazie alla marcatura centrata nel risaltare le diverse parti.

I personaggi presenti nel palco o il set in cui si svolge la scena sono messi in evidenza dalla codifica con l'aiuto di elementi specifici e l'utente che ne usufruirà sarà in grado di fare una ricerca mirata verso ciò che gli servirà.

Nella prima parte è presente la lista degli attori che prendono parte alla commedia ed è indicata con l'elemento <**castList**>**,** inoltre viene specificato il luogo della scena in cui si svolge l'opera (un'isola incantata) con l'elemento <**set**>**.**

Le parti in corsivo vengono identificate con il valore **“italics”**.

Gli elementi base utilizzati sono i seguenti:

- **<speaker>**: indica l’attore (o gli attori) sul palco in quel momento e con l’attributo **“rend”** si indica la posizione del nome sul testo originale;

  Esempio:

  <**speaker rend**="center">ROGER.</speaker>

- **<stage>**: contiene ogni tipo di indicazione di scena e presenta come valori dell’attributo type **“business”** (descrive l’attività che si svolge sul palco), **“exit”** (descrive un’uscita), **“entrance”** (descrive un’entrata), **“delivery”** (descrive come un personaggio sta parlando), **“modifier”** (indica dettagli sul personaggio), **“mixed”** (un misto dei valori precedenti);

  Esempi:

  <**stage** type="**business**" rend="italics">ROGER venant au son des Trompettes et des Tambours.</stage>

  <**stage** type="**modifier**" rend="italics">se réveillant.</stage>

  <**stage** type="**delivery**" rend="italics">en pleurant.</stage>

- **<head type="titleScene">:** indica il titolo;

  Esempio:

  <head type="titleScene">SCENE IV.</head>

- **<lg>:** introduce una parte in versi;

  Esempio:

  <lg>

  <l>En me rendant le jour,</l>

<l>Rendez aussi le calme à mon amour.</l>
</lg>

- **<pb>:** indica un cambio di pagina e con l'attributo **"n"** si indica il numero della pagina;

  Esempio:

  <**pb** **n**="295"/>

- ** :** ogni scena viene introdotta da un in cui si specifica il numero della scena;

  Esempio:

  <**div** type="scene" **n**="5">

- **<graphic>:** indica l'url di un'immagine presa dal testo originale;

  Esempio:

  <**graphic** **url**="immagini/line.png"/>

- **<foreign>:** indica uno o più passi in lingua straniera rispetto al resto del testo. In questo caso, grazie al foglio di stile in css, è stato possibile metterli in evidenza colorandoli in modo diverso.

  Esempio:

  <**foreign** xml:lang="latino">Rara avis in terris</foreign>

6 Il motore di ricerca della piccola BDR

La piccola Biblioteca Digitale Romanza usufruisce di uno strumento *open-source* che permette all'utente di attuare una ricerca semplice ed efficace all'interno della collezione di testi presenti nel sito. Al momento il motore di ricerca utilizzato è Philologic ma, non essendoci una versione aggiornata dal 2010, si pensa di sostituirlo con un altro strumento altrettanto efficiente ma più recente come eXist. Vediamoli in dettaglio.

6.1 Philologic

Per il progetto della piccola Biblioteca Digitale Romanza si è fin'ora usato Philologic (versione 3, precedente alla versione beta 3.2 sviluppata nell'agosto 2010), software di ricerca ampiamente testato.

Philologic è uno strumento di ricerca *full-text*, finalizzato al recupero di documenti e all'analisi di svariati testi, grazie anche a delle implementazioni che rendono possibile una ricerca più accurata anche solo di parti di testi, come atti, scene, articoli e altro ancora. Sviluppato dall'ARTFL *Project* e dal *Digital Library Development Center* dell'università di Chicago, Philologic è un sofwtare libero utilizzato per le grandi raccolte di documenti in TEI-Lite.

Grazie alle immense raccolte di dati XML e alla recente implementazione di strumenti di elaborazione XML, esso rappresenta una grande opportunità per lo sviluppo collaborativo di alto livello, soprattutto per l'utilizzo di strumenti applicabili all'informatica umanistica.

Il potere e la sofisticazione della codifica TEI-XML supporta lo sviluppo della rappresentazione di ricchi dati testuali che incoraggia, a sua volta, lo sviluppo di una serie di strumenti utili allo sfruttamento delle caratteristiche insite del testo codificato, che serviranno per svolgere dei compiti particolari.

Per esempio uno strumento generale è possibile che non sia adatto ad uno specifico documento codificato, ma un insieme di più tecniche e di strumenti possono fornire meccanismi adatti alla distribuzione di applicazioni *end-user*.

Philologic è stato esteso in modo da riuscire a supportare una vasta gamma di documenti codificati raggruppati in basi di dati testuali utilizzabili da numerose istituzioni accademiche e recentemente anche da organizzazioni commerciali.

6.1.1 Vantaggi

Svariati sono i motivi per cui si dovrebbe scegliere Philologic come software di ricerca: è leggero e veloce, robusto e ampiamente testato. L'installazione è quasi del tutto automatica e si hanno molte opzioni di configurazione. Supporta Unicode, usufruisce dell'interoperabilità tra alcuni sistemi ed è un'applicazione *open source*.

Esistono altri motori di ricerca *full-text* che recuperano stringhe di caratteri. Ma al posto di ricercare due parole all'interno di una stessa frase o di uno stesso paragrafo (unità intellettuali), questi altri motori devono cercare due parole all'interno di un certo numero di caratteri, a prescindere che siano frasi o paragrafi.

Grazie a Philologic, gli studiosi sono in grado di sapere sempre in che punto del testo si trovano, in quanto l'impaginazione può essere visualizzata accanto ad altri oggetti. Ciò potrebbe far pensare che la velocità venga diminuita, ma Philologic è stato ottimizzato in modo che, anche con questo elevato livello di indicizzazione, risulti ancora incredibilmente veloce sul web.

6.1.2 Le 5 fasi della ricerca

La ricerca su cui si basa Philologic è divisa in 5 fasi distinte:

1. per prima cosa si dovrà definire un corpus, questo servirà per limitare la ricerca;

2. poi si definiranno le parole;

3. si avvierà una ricerca indicizzata della parola;

4. una volta trovata, si estrae dal testo;

5. infine si formatta il collegamento (per esempio si passa da SGML a HTML).

In pratica, una volta definita la tipologia dell'intero corpus, si può proseguire alla ricerca di un singolo termine o anche di un'intera frase.

Philologic, controllando gli indici delle parole in un database relazionale, estrae il blocco di testo che contiene la parola cercata, grazie a collegamenti con altri blocchi di testi più grandi. I blocchi estratti verranno poi formattati per renderli visibili in un browser e talvolta possono contenere anche link ad immagini, registrazioni sonore, altri testi o altri database.

Grazie all'utilizzo di espressioni regolari estese di tipo UNIX, utilizzabili per la ricerca di caratteri "jolly" e anche per alcune implementazioni di carattere morfologico e ortografico, ogni utente è in grado di effettuare ricerche più avanzate e sofisticate, andando oltre alla semplice ricerca di una parola o di una frase.

Questi strumenti possono essere combinati fra di loro utilizzando gli operatori booleani all'interno di una grande varietà di contesti.

Originariamente, Philologic fu progettato per la ricerca scientifica su banche di dati di collezioni letterarie, religiose, filosofiche e storiche, o importanti enciclopedie e dizionari storici.

L'utente può avviare una ricerca di script non romanizzati o di parole con segni diacritici (specificando la presenza di accenti, oppure ignorandoli scrivendo in maiuscolo).

Al momento, nel web si possono trovare una cinquantina di database in Philologic, composti da una grande varietà di lingue come il greco antico, latino, hindi, urdu e anche tutte le lingue dell'Europa occidentale.

Philologic può essere impostato in modo da riconoscere e ignorare completamente alcune notazioni tipiche dei manoscritti. Infatti, grazie al riconoscimento delle strutture tipiche di un testo come oggetti di dati reali, questo software è in grado di individuare le unità come parole, frasi, paragrafi, sezioni e pagine, permettendo una ricerca molto flessibile e un recupero accurato di questi oggetti testuali.

6.2 eXist

eXist è un sistema di gestione *open source* di database interamente basato sulla tecnologia XML, chiamato anche "database nativo in XML". A differenza di molti altri sistemi di gestione di database relazionali, eXist utilizza XQuery per manipolare i propri dati. Rilasciato secondo i termini della licenza GNU LGPL.

6.2.1 Le origini

Fondato alla fine del 2000 da Wolfgang Meier, eXist è sempre in fase di sviluppo. Wolfgang è ancora il leader di questo progetto e lo sviluppatore principale.

Durante tutto questo tempo, i piani originali del leader sono cambiati, soprattutto in risposta alle richieste degli utenti. Inizialmente, l'obiettivo principale di eXist era centrato sui documenti centrali di una raccolta. Ora, grazie all'aggiornamento recente dei documenti, l'attenzione si è concentrata sulle applicazioni sviluppate da eXist basate sui dati.

Supporta le interfacce REST per l'interfacciamento con i moduli web di tipo AJAX.[21]

Si ha la possibilità di salvare i propri dati utilizzando pochissime righe di codice e permette agli utenti di caricare e cancellare dei file XML direttamente nel database. Poiché eXist indicizza automaticamente i documenti usando delle parole chiave, è molto facile creare sistemi di ricerca di documenti ad alto livello.

[21] *http://en.wikipedia.org/wiki/EXist*

6.2.2 Le versioni

Nel settembre del 2006 ha raggiunto la versione 1.0 e 1.1.

Attualmente si sta lavorando alle versioni 1.4.x e si hanno nuovi sviluppi sulla versione 1.5dev che verrà rilasciata come 2.0.0. eXist è stato nominato come miglior database XML dell'anno da *InfoWorld* nel 2006.[22]

6.2.3 Tecnologie supportate

Supporta i seguenti standard e tecnologie:

- **XPath**: linguaggio XML
- **XQuery**: linguaggio di query XML
- **XInclude**: server-side che include l'elaborazione dei file
- **XML-RPC**: un protocollo di chiamata di procedura remota
- **XQuery API for Java**

Grazie ad una *mailing list*, tutti gli utenti possono contribuire allo sviluppo di applicazioni per eXist.

6.2.4 XQuery

XQuery è un linguaggio di richiesta e programmazione funzionale progettato per interrogare le collezioni di dati XML.

Versioni

La versione 1.0 è stata sviluppata dal gruppo di lavoro del W3C addetto alle query XML.

[22] *Ibidem*

Il lavoro è stato in stretto coordinamento con lo sviluppo di XSLT 2.0 del gruppo di lavoro XSL e questi due gruppi condividono la responsabilità di XPath 2.0 che è un sottoinsieme di XQuery 1.0.[23]

La sua missione è quella di fornire un servizio flessibile di query per estrarre dati dai documenti reali e virtuali sul World Wide Web, fornendo quindi finalmente l'interazione necessaria tra il mondo del Web e quello dei database. Le collezioni di file XML saranno accessibili come i database.[24]

Vantaggi e svantaggi

XQuery fornisce i mezzi per estrarre e manipolare i dati provenienti da documenti XML o altre fonti di dati che possono essere visti come XML, come per esempio database relazionali o documenti di ufficio.

Usa la sintassi delle espressioni XPath per gestire parti specifiche dei documenti XML.

Questo linguaggio permette anche di costruire nuovi documenti XML e si basa su una struttura ad albero che contiene sette tipi di nodi: nodi documento, elementi, attributi, nodi di testo, commenti, istruzioni di elaborazione e spazi dei nomi.

Il sistema tipo dei modelli di questo linguaggio pone tutti i valori come sequenze. Gli elementi in una sequenza possono essere dei nodi o dei valori atomici, questi ultimi possono a loro volta essere interi, stringhe, booleani e così via.

La versione 1.0 non include funzionalità per l'aggiornamento dei documenti XML o dei database, ma manca anche della capacità di ricerca *full-text*. Queste caratteristiche sono sotto sviluppo per una versione successiva del linguaggio. I nuovi standard però supportano le funzionalità di aggiornamento.

Caratteristiche

XQuery è un linguaggio di programmazione che può esprimere un XML arbitrario

[23] *http://en.wikipedia.org/wiki/XQuery*
[24] *Ibidem*

per le trasformazioni di dati XML con le seguenti caratteristiche:

- Indipendenza dei dati logici/fisici
- Dichiarativo
- Elevato
- Senza *Side-effect*
- Fortemente tipizzato

Applicazioni[25]

Elenco dei modi in cui XQuery può essere utilizzato:

1. Estrazione di informazioni da un database per l'utilizzo di un servizio web;
2. generazione di report riassuntivi dei dati memorizzati in un database XML;
3. ricerca di informazioni pertinenti in documenti testuali nel Web e elaborazione dei risultati;
4. selezione e trasformazione dei dati XML in (X)HTML da pubblicare sul web;
5. estrazione di dati dai database da utilizzare per l'integrazione delle applicazioni;
6. divisione in più documenti XML di un singolo documento che rappresenta più transizioni.

[25] *Ibidem*

7 Conclusioni

Lo studio svolto sui motori di ricerca XML ci ha portato alla conclusione che Philologic è uno strumento ottimo per il tipo di ricerca che si svolge nell'ambito del progetto della piccola BDR, in quanto, grazie alle sue caratteristiche, permette un lavoro pulito e diretto. Purtroppo, come detto nei capitoli precedenti, non esce una versione aggiornata da molto tempo. L'ultima versione è stata pubblicata nel 2010, e oltretutto è una versione beta non ancora testata a sufficienza.

Per il sito serve quindi un software più recente e con aggiornamenti continui rispetto a Philologic. Per questo si è pensato di analizzare un altro strumento come eXist, efficace quanto Philologic e più recente, sempre in continuo sviluppo. eXist è stato preso in considerazione come un valido sostituto dell'attuale motore di ricerca del sito della piccola BDR.

Nel frattempo il lavoro che si è svolto è stato quello di aggiornare sempre di più la collezione dei testi presente nella piccola BDR, aggiungendo e codificando nuove opere, cercando di arricchire maggiormente il database.

Ovviamente il lavoro non è finito, in quanto si può e si deve raccogliere ancora più materiale da analizzare, codificare e memorizzare.

Un passo decisivo è stato il passaggio dalla "vecchia" codifica in versione TEI Lite P4 a quella più recente, la P5. Questo lavoro fa si che la piccola BDR rimanga sempre aggiornata e che i testi, recenti e meno recenti, con la rispettiva codifica, siano sempre uniformi fra di loro.

8 Webliografia

- http://en.wikipedia.org/wiki/EXist
- http://en.wikipedia.org/wiki/XQuery
- http://exist-db.org/exist/credits.xml
- http://it.wikipedia.org/wiki/Text_Encoding_Initiative#Le_linee_guida_del_progetto
- http://it.wikipedia.org/wiki/XML
- http://piccolabdr.humnet.unipi.it/
- https://sites.google.com/site/philologic3/
- https://sites.google.com/site/philologic3/encoding/ate-artfl-text-encoding
- http://www.lib.uchicago.edu/efts/ARTFL/philologic/
- http://www.tei-c.org/Guidelines/Customization/Lite/teiu5_it.xml

Ringraziamenti

Se sono arrivata fino a questo punto il merito non è esclusivamente mio. Ringrazio quindi tutte le persone che mi sono state vicine per il sostegno, l'incoraggiamento e i consigli attraverso i quali ho potuto conseguire questo traguardo.

Il primo pensiero va, ovviamente, ai miei genitori, senza i quali non sarei mai potuta giungere a questo punto, non solo per il sostegno economico, che sicuramente è stato fondamentale, ma per quell'aiuto indispensabile per superare gli ostacoli incontrati in questi anni e per aver creduto sempre in me, nonostante tutto, incoraggiandomi in ogni mia scelta.

Ringrazio mio fratello per essere sempre stato presente, per il suo supporto e i numerosi consigli durante questi anni.

Un ringraziamento speciale a Save, che mi ha sopportato nelle mie crisi di ansia e paranoia, che mi è sempre stato vicino e ha reso il tutto molto più leggero, aiutandomi a superare con facilità ogni momento.

Ringrazio tutti ma proprio tutti i miei amici, quelli vicini e quelli lontani (ma solo fisicamente): un grazie infinito alle mie amiche d'infanzia, d'adolescenza e oltre, Giulia, Giuseppina ed Elena, con il quale sono cresciuta e ho condiviso tanti bei momenti e che, nonostante la lontananza, mi stanno sempre vicino e continuano a condividere la loro vita con me; ringrazio Paola e Francesca, compagne di liceo un tempo ma diventate subito un punto fermo nel cerchio delle mie amicizie più strette, per essere state e per essere sempre presenti (mitico skype!); grazie a tutte loro e a Giovanni, Anna e Andrea che, quando torno a casa, mi fanno sentire come se non me ne fossi mai andata.

Grazie agli amici del periodo universitario e non solo, a quelli dei primi anni e a quelli più recenti: grazie ai miei fiuri, Rossana, la prima persona che ho conosciuto arrivata a Pisa, ed Elisabetta, coinquiline e amiche che, con la loro esuberanza e tutto il resto, sono state un grande aiuto per superare i primi momenti di spaesamento

in una città sconosciuta e che, grazie alle mitiche cene, continuano a far parte della mia esperienza universitaria; un enorme grazie agli amici che mi sono stati vicino in questo periodo e non solo, che mi hanno supportato e che mi hanno aiutato a rendere questi anni di università molto più leggeri e meno faticosi, ognuno a modo suo e tutti insieme a modo loro! Grazie quindi ad Alessandro, Gigi, Frankekko, Caterina, Andrea, Luca, Leo, Matteo. Grazie anche agli amici dei primi anni di università, Calos, Dejan, Antonio e Adriano, che mi hanno aiutato ad affrontare con leggerezza il primo periodo pisano e che, nonostante il gruppo si sia un po' sfaldato, in un modo o nell'altro sono sempre presenti.

Grazie a Eleonora, compagna di studi e di paranoie pre-esame, con il quale ho condiviso le gioie e le ansie in questi anni di università.

Ringrazio tutti i parenti, in particolar modo mio padrino Paolo, Zia Candida, Mario e Maria, che sono stati sempre vicini e presenti e su cui so di poter sempre contare.

Infine desidero ringraziare la Prof.ssa Sommovigo, il Prof. Rosselli Del Turco e la Prof.ssa Simi, relatori di questa tesi, per la grande disponibilità e cortesia dimostratemi, per il sostegno e tutto l'aiuto fornito durante il lavoro di tirocinio e la stesura della relazione.