Umanistica Digitale - ISSN:2532-8816 - n.8, 2020
M. Materni – Complessità della codifica ed ergonomia strumentale nel contesto XML-TEI: dove siamo?
DOI: http://doi.org/10.6092/issn.2532-8816/9976

Complessità della codifica ed ergonomia strumentale nel contesto XML-TEI: dove siamo?

(Bilancio a partire da un nuovo progetto di edizione digitale medievale)

Marta Materni

Université Grenoble Alpes, France
marta.materni@univ.grenoble-alpes.fr

Abstract

Il contributo, nato dall'esperienza di un progetto di edizione digitale finanziato da una Marie Curie Fellowship (*DigiFlorimont*, grant n° 745821), intende fare il punto sulla questione, ancora problematica, della produzione di file XML-TEI. Gli strumenti sono infatti ben lontani dall'essere ergonomici. La questione è spinosa in quanto la lacuna strumentale incide sia sulla diffusione della pratica editoriale digitale, sia sulla qualità della codifica proposta. Il rischio è quello di continuare a produrre "codifiche semplici con strumenti complessi" quando ormai la quotidianità dovrebbe essere la produzione di "codifiche complesse con strumenti semplici". Il progetto *DigiFlorimont* ha implicato la creazione di un editor-prototipo XML-TEI capace di generare il codice in modo automatico a partire da una sintassi simbolica (sul modello Markdown) applicata a un file .txt associato con un .csv. È stato così possibile realizzare una codifica complessa secondo un modello di testo concepito come un database di parole, pronto a essere utilizzato per altre analisi al di là degli obiettivi specifici del contesto di produzione. Si precisa in questo modo il ruolo del "filologo digitale", responsabile della diffusione all'interno della comunità accademica di un testo filologicamente e informaticamente corretto, "pronto all'uso", secondo un modello di capitalizzazione progressiva del lavoro di analisi.

The aim of this paper is, first of all, to offer on overview of the problems related to the production of XML-TEI files and it summarizes the experience gained in the realization of a digital edition project financed by a Marie Curie Fellowship (*DigiFlorimont*, grant n° 745821). The tools are in fact still far from being ergonomic. The issue is crucial as the instrumental gap affects both the spread of digital publishing and the quality of the proposed textual encoding. The risk is to continue producing a "simplified encoding with complex tools" when, after decades of practice, we should rather produce a "complex encoding with simplified tools". The *DigiFlorimont* project involved the creation of an XML-TEI editor-prototype that generates the code automatically starting from a symbolic syntax (inspired by the Markdown model) applied to a .txt file associated with a .csv file. Thanks to this tool, it was possible to create a very

complex encoding XML-TEI according to a textual model conceived as a words database, ready to be used for other analyzes beyond the specific objectives of the production context. These technical reflections also aim to clarify the role of the "digital philologist": he is a philologist charged of disseminating within the academic community a philologically and informatically correct text, "ready for use", according to a scholarly model of progressive capitalization of the analysis work.

Introduzione

Nel 2005, in un articolo ancora (purtroppo) di attualità e che offre numerosi spunti di riflessione, Peter Robinson lanciava un grido di allarme: "We need some things we do not yet have: software that does not exist and established online publication systems that have yet to be created. Let us not wait too long for these" ([7]: § 31). La letteratura grigia lascia spesso trasparire del disagio al riguardo, e nemmeno tanto velato. Tara Andrews, autrice di una soluzione parzialmente analoga a quella qui proposta, Text::TEI::Markup, è piuttosto efficace nella descrizione:

> TEI XML is a wonderful thing. The elements defined therein allow a transcriber to record and represent just about any feature of a text that he or she encounters. *The problem is the transcription itself*. When I am transcribing a manuscript [...] I do not want to be switching back and forth between keyboard layouts in order to type <tag attr="attr">arrow-arrow-arrow-arrow-arrow</tag> every six seconds. It's prone to typo, it's astonishingly slow, and it makes my wrists hurt just to think about it. (*corsivi miei*) ([1])

Ma le stesse *Guidelines* TEI, seppure indirettamente, alludono alla complessità di produzione dei file .xml:

> XML can appear distressingly verbose, particularly when (as in these Guidelines) the names of tags and attributes are chosen for clarity and not for brevity. Editor macros and keyboard shortcuts can allow a typist to enter frequently used tags with single keystrokes. It is often possible to transform word-processed or scanned text automatically. Markup-aware software can help with maintaining the hierarchical structure of the document, and display the document with visual formatting rather than raw tags. ([4] § *Use in Text Capture and Text Creation*)

Recentemente Roberto Rosselli del Turco ha ripreso la questione in un articolo di sintesi del 2016 ([8]) che prende in considerazione i due poli della problematica: produzione e visualizzazione di un'edizione digitale. L'argomento è talmente vasto (e bisognerebbe includere anche un terzo polo, quello del processamento dei testi formalizzati in termini informatici, vale a dire l'elemento che più marca la differenza tra un'edizione cartacea e un'edizione digitale) che in questo contesto vorrei soffermarmi solo sul polo produzione, che si rivela strettamente connesso alla questione della complessità del modello di codifica proposto: "complessità di codifica di una codifica complessa" potrebbe essere la definizione sintetica del problema. La

conclusione di Rosselli del Turco un decennio dopo la denuncia di Robinson costituisce la base di partenza per la riflessione:

> Robinson's remark about the lack of easy-to-use production tools is unfortunately still valid: there is no software tool nor suite of tools that allows a scholar to produce a full digital edition, be it image-based with a diplomatic text or a critical edition, in a way comparable to how printed editions are prepared. ([8]: § 16)

Il problema strumentale è ovviamente scientificamente piuttosto grave in quanto rischia di condizionare in modo *inaccettabile* le nostre proposizioni teoriche di modellizzazione della codifica testuale, il che significa in ultima analisi una prospettiva in cui lo strumento tecnico ha il potere di influenzare le nostre decisioni intellettuali riguardanti la rappresentazione del pensiero. Ecco ancora una frase di Robinson a farci da guida:

> A well-made electronic scholarly edition will be built on encoding of great complexity and richness. As well as free text searching, efficient search systems can make use of this encoding to enable sophisticated search, going considerably beyond the standard word and phrase search ([6]).

Nel momento in cui si sceglie la via della codifica testuale, la complessità di quest'ultima diventa una premessa necessaria per poter mettere in atto un'analisi testuale altrettanto complessa. E per codifica complessa intendo una codifica che trasformi la sequenza informe di spazi, caratteri e punteggiatura in una sequenza di oggetti identificabili e manipolabili. Altrimenti, a che pro ricorrere alla dimensione digitale quando quella stampata è già così rodata ed efficiente rispetto a una certa ben radicata visione dell'edizione? E qui è d'obbligo citare le parole del padre nobile dell'informatica umanistica. In un intervento del 1997, in apertura dei lavori del convegno trentino *Umanesimo e Informatica*, padre Roberto Busa ammoniva:

> Mi preme fare ai giovani una raccomandazione: non mettete vino vecchio in otri nuovi, e tenete conto che l'informatica non è per fare le stesse ricerche di prima con gli stessi metodi di prima ma solo più velocemente e magari con meno lavoro umano. L'informatica obbliga a due cose: primo, all'invenzione di nuove strategie di ricerca, proporzionate alla possibilità di questo strumento; e, secondo, impegna a un lavoro umano più intenso, più condensato, a livelli umani superiori. […] le banche dei dati buttano su supporto elettronico i testi così come sono: secondo la peculiarità del linguaggio anglosassone in informatica, cioè come *full text*, espressione che io critico ferocemente. Mettono cioè su supporto magnetico solo le parole e le interpunzioni, e basta. Per cui le ricerche vengono basate su stringhe di caratteri, su combinazioni o distribuzioni di parole. Nell'informatica ermeneutica occorre molto di più: un lungo lavoro di *pre-editing*, preparazione del testo. E siccome nel computer ci sono soltanto i segni e non i significati, per usare la consueta terminologia metaforica e impropria, il computer "capisce, ricorda, elabora, moltiplica, divide, indovina", lavora e opera solo sui segni che in esso ci sono. ([3])

In questo senso il criterio talvolta sostenuto della leggibilità (da parte umana) del file XML non può a mio avviso rappresentare un principio guida della codifica, a meno di rinunciare, in nome di questa leggibilità, alla possibilità di analisi radicalmente alternative al paradigma editoriale cartaceo. D'altronde sono le stesse *Guidelines* TEI a indicarci esplicitamente, nella sezione *About*, che "They make recommendations about suitable ways of representing those features of textual resources which need to be identified explicitly in order to facilitate *processing by computer programs*" (corsivo mio) ([4]). E ancora, poco più oltre, in apertura del paragrafo *Use for Local Processing*, leggiamo: "*Machine-readable text* can be manipulated in many ways" (corsivo mio) ([4]: § iv.1). Infine, l'*Historical Background* ci ricorda che "[the] first draft version (P1 […]) of the Guidelines was distributed in July 1990 under the title *Guidelines for the Encoding and Interchange of Machine-Readable Texts*" ([4]: § iv.2). Un titolo, aggiungerei, estremamente significativo.

L'obiettivo dovrebbe cioè essere quello di formalizzare al massimo il testo in modo da renderlo un oggetto documentario processabile dalla macchina (*machine-readable*) affinché l'umano ne possa trarre il massimo delle informazioni. In questa prospettiva, XML è un linguaggio leggibile dagli umani nel senso che la sua sintassi ispirata al linguaggio naturale permette effettivamente una eventuale possibile "lettura visuale" del codice da parte dell'occhio umano, ma, almeno dal mio punto di vista, esso non è destinato, quanto a finalità intrinseca, a essere oggetto di una "lettura contenutistica" da parte dell'utente umano. XML costituisce cioè lo strumento attraverso cui si costruisce una formalizzazione dei dati tale da permettere la simulazione di un'interazione intelligente con quel meccanismo *free-context* che è il computer: più il file XML è contenutisticamente leggibile (e non solo visualmente) dall'umano, più, paradossalmente, è la macchina che vince sull'umano perché l'umano non la sfrutta al massimo del suo potenziale.

A distanza di dieci anni dalla precedente citazione di Robinson, Tara Andrews, nel Read.me del suo Text::TEI:Markup, continua a sottolineare l'esistenza di questo gap, molto banalmente pratico e profondamente frustrante, tra il modello concettuale TEI, ricco di potenzialità, e la quotidianità concreta rappresentata dalle dita del codificatore alle prese con i tag:

> The TEI P5 guidelines include a module called "analysis", which allows the user to tag sentences, clauses, words, morphemes, or any other sort of semantic segment of a text. This is really good for programmatic applications, but very boring and repetitive to have to tag ([1]).

Gli strumenti proposti ufficialmente, come *Oxygen*, con la loro farraginosità di produzione, ci inchiodano di fatto a codifiche al limite talvolta della banalità. Di fronte alla ricchezza di informazioni estrapolabili da un verso in antico francese tramandato da un manoscritto medievale – sistema abbreviativo, stadio evolutivo della lingua, grafie regionali, segmentazione della scrittura, forme paleografiche ecc. –

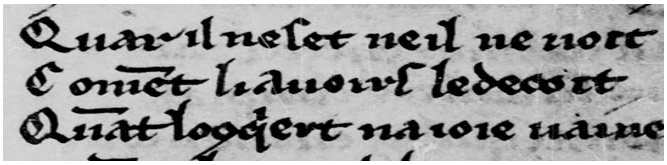

Figura 1: Testo esemplificativo del *Roman de Florimont* in antico francese (Tours, Bibliothèque Municipale Ms. 941)

quanti dati perdiamo proponendo una codifica di questo tipo, purtroppo piuttosto diffusa?

```
<l>Quar il ne set ne il ne voit</l>
<l>Come<ex>n</ex>t li avoirs le deçoit</l>
<l>Qua<ex>n</ex>t lo <ex>con</ex>q<ex>ui</ex>ert n'a joie vaine</l>
```

Quanto possiamo invece guadagnare in potenziale di analisi con una codifica di quest'altro tipo, che trasforma il testo in un database di parole, ciascuna identificabile e quindi analizzabile e quindi manipolabile? Limitiamoci al terzo verso:

```
<l n="3" xml:id="msl3">
   <w xml:id="msl3w1">Qua<expan corresp="#abb-tild-
nas"><ex>n</ex></expan>t</w>
   <w xml:id="msl3w2" ana="#aggl-s">lo</w>
   <w xml:id="msl3w3"><expan corresp="#abb-tir-9"><ex>con</ex></expan>
     <expan corresp="#lsup-q">q<ex>u</ex>i</expan>ert</w>
   <w xml:id="msl3w4" ana="#elis">n</w>
   <w xml:id="msl3w5" ana="aggl-s-unc">a</w>
   <w xml:id="msl3w6"><c ana="#lram-cons">i</c>oie</w>
   <w xml:id="msl3w7"><c ana="#lram-cons">u</c>aine</w>
</l>
```

Possiamo in questo modo, per es., ottenere una visualizzazione diplomatica in modalità "trascrizione":

> Qua*n*t lo*con*q*u*iert na-ioie uaine

e in modalità "interpretativa":

> Quant lo conquiert n'a joie vaine

Si tratta del modello di codifica proposto per un recente progetto di edizione diplomatica digitale dei testimoni di un romanzo francese del XII sec., il *Roman de Florimont*, concepito come una sorta di modello di codifica proiettato verso il futuro, al di là degli obiettivi contingenti legati al progetto specifico: il testo, cristallizzato nella forma documentaria che si offre ai nostri occhi, viene scomposto nei suoi elementi minimali portatori di significato – la parola lessicale (un paleografo, interessato all'immagine del testo, probabilmente considererà

come unità minima il carattere; aggiungo inoltre che la nozione di "parola" non potrebbe applicarsi *tout court* a realtà linguistiche differenti da quelle delle lingue romanze, si pensi ad es. alle lingue agglutinanti come il turco,) – e ricostruito sotto forma di un file testuale codificato che vuole rappresentare un avatar digitale, cioè *formalizzato*, della realtà manoscritta.

Così preparato, il file XML-TEI ha soddisfatto gli obiettivi immediati del progetto – la realizzazione di un'edizione diplomatica visualizzabile in modalità trascrizione grafematica e in modalità interpretativa modernizzata –, offrendosi al contempo come prodotto già pronto per essere sottoposto a ulteriori analisi: lemmatizzazione ed etichettattura morfo-sintattica, utilizzo del *Double end-point attached method* per la creazione di un'edizione critica, analisi sintattico-retoriche ecc. facilmente codificabili a questo punto attraverso uno *stand-off markup* dato che ciascuna parola è provvista di un @xml:id (allo stato attuale del progetto *DigiFlorimont*, un primo *stand-off markup* è già stato utilizzato per codificare i discorsi diretti e i monologhi, attraverso il tag @from @to). Si tratta cioè di applicare il principio metodologico cumulativo dei "nani sulle spalle dei giganti", senza naturalmente attribuire alcun valore di "nanismo" o "gigantismo" contenutistico ai vari momenti. Fuor di metafora, si tratta di capitalizzare al massimo il lavoro, cioè il tempo della ricerca; di concepire il processo come un percorso di accumulazione progressiva di strati ulteriori di analisi (i nani) a partire da una base fissata una volta per tutte (i giganti).

La posizione pragmatica, direi quasi imprenditoriale, rispetto alla codifica, sostenuta con forza per es. da Elena Pierazzo:

> We all know how important economic considerations are in our decision-making processes; almost all of our research projects are funded for a specific time-span and budget, and so it is fundamental to ensure that the transcription (and encoding) is feasible within this lifetime. *The decision whether or not to encode a specific feature will be heavily determined by the cost of encoding it*, and it would be naive not to admit this. Economic considerations may then be used as the pragmatic limits of the level of transcription we are looking for, in the same way that the limits of the typography worked for print-based publications. [...] How far should we go in considering the needs of the Others? [...] if it is matter of considering the needs of possible future scholars in other disciplines and providing special markup for them, the temptation is to say: 'not far'. There is, in fact, a serious risk of wasting precious project time here: it is very difficult to guess other scholars' needs or principles which are, potentially, infinite – as infinite as the set of 'facts' that can be derived from the document. (corsivi miei) ([5]: 469, 471)

è esattamente il contrario di quanto si vuole qui proporre, e cercare di perseguire. Se tempo e denaro influenzano ovviamente le nostre attività in quanto in assenza di un tempo di lavoro retribuito non si potrà realizzare una ricerca, l'obiettivo scientifico non potrà tuttavia essere calibrato assecondando le 'leggi del mercato'. Invece di parlare di un "cost of encoding" da utilizzare come criterio di inclusione o esclusione di elementi nella concezione del modello di codifica, dovremmo piuttosto impegnarci in un percorso di sperimentazione e ricerca che permetta semmai di 'ottimizzare i tempi di produzione', per continuare a utilizzare questa

metafora mercantile, al fine di ottenere ciò che si reputa scientificamente corretto e necessario e non 'economico'.

Inoltre, la preoccupazione riguardo "how far should we go in considering the needs of the Others" potrebbe essere in realtà una falsa preoccupazione derivante dalla tendenza a cumulare nello stesso file gli strati di codifica descrittiva e gli strati di codifica interpretativa. Piuttosto che prevedere uno *special markup* per ciascuna, possibile disciplina, in quanto filologi creatori di un'edizione digitale dovremmo in realtà proporre una sorta di *markup passe-partout* che descriva ma soprattutto formalizzi il testo, al quale poi le varie discipline applicheranno la loro griglia, e codifica, interpretativa. Lo *stand-off markup* è dal mio punto di vista la risposta per eccellenza, risposta che presuppone, per essere praticata correntemente, la formalizzazione del testo come agglomerato di elementi minimi identificati in modo univoco attraverso un @xml:id e utilizzabili quindi come *anchor*.

La questione diventa dunque: come creare in tempi ragionevoli e in una modalità scevra da errori ricorrenti un prodotto complesso e utile a me e agli altri? Non certo attraverso editor XML 'prestati' al mondo TEI, bensì attraverso editor XML-TEI *specificatamente* concepiti per la codifica testuale in una prospettiva informatico-umanistica, e, in più, attraverso la massima automatizzazione della produzione dei tag XML: più il processo è automatizzato, più l'errore della digitazione è ridotto e la complessità della codifica incrementata.

Se, ed è evidente, il processo di produzione con strumenti come Oxygen è farraginoso e lento, parimenti, ed è una posizione personale, non reputo che la soluzione del problema possa passare attraverso piattaforme online come T-Pen[1] o la nuova nata TextualCommunity[2], che spostano il problema dall'ambiente di lavoro in locale a quello in linea, incatenandoci però in questo modo alla schiavitù di un utilizzo online e alla dipendenza da strutture complesse che soffrono di una certa fragilità tecnologica. Né attraverso strumenti che nascondono il codice e che rappresentano un vero e proprio ossimoro teorico rispetto all'idea della codifica testuale: se il primo insegnamento rispetto al senso profondo della TEI insiste sulla differenza fra un sistema WYSIWYG (What You See Is What You Get) e uno WYSIWYM (What You See Is What You Mean), utilizzando uno strumento che cela ai nostri occhi il codice XML-TEI ci si trova a produrre un codice WYSIWYM con uno strumento WYSIWYG, rischiando così di entrare in contraddizione con la stessa filosofia di cui si dovrebbe essere divulgatori.

Quello che si intende illustrare in questo contesto è un prototipo per un processo di produzione che passa attraverso un file testuale 'di mediazione' basato su una sintassi simbolica ispirata al Markdown HTML e alle *entities* XML (&…;). Si tratta della metodologia sperimentata in prima persona nella realizzazione del progetto *DigiFlorimont* e concretizzatasi nella creazione di un editor-prototipo, TEI-Medit (TEI Mediator-editor), realizzato in Python. Il sistema ha permesso di realizzare la trascrizione completa e la codifica a livello di parole (<w>) di sei manoscritti di un testo di circa 13.500 versi (all'inizio del progetto ne erano stati previsti quattro) nello spazio di circa diciotto mesi. Le edizioni sono in corso di pubblicazione

1 http://www.t-pen.org/TPEN/

2 http://www.textualcommunities.usask.ca

sul sito *digiFlorimont. Archive numérique du "Roman de Florimont" d'Aimon de Varennes (Lyon, 1188)*[3].

La soluzione in sé non è originale e, una volta messo a punto l'editor – inizialmente ispirato dalla lettura della documentazione di lavoro dei progetti *Charrette Project*[4] (che utilizza delle entities SGML per rappresentare le abbreviazioni e le iniziali) e *BFM Base du Français Médiéval*[5] (che utilizza una serie di simboli ASCII per generare automaticamente alcuni tag) – ho scoperto l'esistenza di almeno altre due iniziative, Text::TEI::Markup e TEIdown, che si basano sui medesimi principi. Tuttavia, la constatazione della poligenesi in questo caso è stata confortante piuttosto che frustrante in quanto ha indirettamente fornito una serie di conferme: in primo luogo la conferma che il problema sussiste; e in secondo luogo la conferma che la tipologia di soluzione proposta è condivisa (e potrebbe quindi rappresentare, forse, una buona soluzione).

Le soluzioni Text::TEI::Markup e TEIdown

Fra i vari strumenti messi liberamente a disposizione da Tara Andrews[6] (Università di Vienna) e relativi sopratutto a questioni di collazione e stemmatologia, spicca, per il discorso che si sta qui conducendo, il pacchetto Text::TEI::Markup, definito come "a transcription markup syntax for TEI XML" ([1]). Nelle parole della stessa autrice (si consideri la citazione in apertura di questo contributo, nella quale si evidenziava il problema della difficoltà di produzione di un file XML-TEI): "Text::TEI::Markup is my solution to that problem. It defines a bunch of single- or double-character sigils that represent tags. These are a lot faster and easier to type; I don't have to worry about typos; and I can do it all with a plain text editor, thus minimizing use of the mouse" ([1]).

Il funzionamento di Text::TEI::Markup, realizzato in Perl, si articola intorno a tre elementi:

1. Il <teiHeader> è prodotto automaticamente a partire da un template in cui sono di volta in volta inseriti i valori espressi nel file di input:

```
=HEAD
title:My Summer Vacation: a novel
author:John Smith
myinitials:tla
myname:Tara L Andrews
=BODY
The ^real^ text b\e\gins +(above)t+here.
...
```

Figura 2: Il file di input del <teiHeader> secondo la sintassi Text::TEI::Markup

3 http://digiflorimont.huma-num.fr/

4 http://www.princeton.edu/~lancelot/ss/

5 http://bfm.ens-lyon.fr

6 https://metacpan.org/author/AURUM

```
<?xml version="1.0" encoding="UTF-8">
<TEI>
  <teiHeader>
        <fileDesc>
          <titleStmt>
                <title>__TITLE__</title>
                <author__AUTHOR__</author>
                <respStmt xml:id="#__MYINITIALS__">
                  <resp>Transcription by</resp>
                  <name>__MYNAME__</name>
                </respStmt>
          </titleStmt>
        </fileDesc>
  </teiHeader>
  <text>
        <body>
        __MAIN__
        </body>
  </text>
</TEI>
```

Figura 3: Il template Text::TEI:Markup per il teiHeader

2. Il cuore della sintassi Text::TEI::Markup è rappresentato dall'elenco dei segni utilizzati per produrre una serie di tag all'interno del <body>:

```
%SIGILS = (
        'comment' => '##',
        'add' => '+',
        'del' => '-',
        'subst' => "\x{b1}",    #
        'div' => "\x{a7}",
        'p' => "\x{b6}",
        'ex' => '\\',
        'expan' => '^',
        'supplied' => '@',
        'abbr' => [ '{', '}' ],
        'num' => '%',
        'pb' => [ '[', ']' ],
        'cb' => '|',
        'hi' => '*',
        'unclear' => '?',
        'q' => "\x{2020}",
        );
```

Figura 4: I simboli della sintassi Text::TEI::Markup

In alcuni casi è possibile passare ai tag un argomento per generare un attributo:

+	+esempio+	<add>**esempio**</add>
+( )	+(marg)esempio+	<add place="margin">**esempio**</add>

3. Infine, la funzione *word_tag_wrap* "takes an XML string as input, looks for words (defined by whitespace separation) and returns an XML strings with each of these words wrapped in an appropriate tag" ([1]).

Nel 2017, Alejandro Bia Platas e Ramón P. Ñeco García (Universidad Miguel Hernández de Elche), hanno presentato il prototipo TEIdown, fondato su un "uso de Makdown extendido para el marcado automático de documentos TEI" ([2]), ma limitato ai tag più basilari, per es. quelli riguardanti la struttura in prosa o in versi, oltre a quelli più comunemente usati all'interno del teiHeader. Si tratta quindi di un'automatizzazione parziale della produzione, in quanto dopo la conversione è necessario un re-intervento a mano sul file .xml per poter introdurre il resto della codifica. Rapidità e semplicità nella creazione del file, riduzione al minimo dei problemi di validazione e facilità di correzione hanno rappresentato gli obiettivi alla base della creazione del prototipo:

> el proyecto TEIdown […] consiste en una ampliación del Markdown para obtener una sintaxis abreviada que sirva para la *creación rápida* de documentos XML-TEI y en la creación de los programas de procesamiento correspondientes para llevar a cabo dicha conversión. Con este enfoque, resulta más sencilla la obtención de un documento TEI válido *en un tiempo reducido*, evitando pasar por una larga lista de errores de validación. […] Esta sintaxis ha sido cuidadosamente diseñada con el objetivo de que sea *fácil de leer y de corregir* y, sobre todo, *fácil de codificar*. (*corsivi miei*) ([2]: 59, 69)

Il prototipo Markdown prevede un trattamento e una sintassi differenti per il teiHeader e per il corpo del testo: nel primo caso ci si è ispirati all'header dell'AsciiDoc[7], nel secondo alla sintassi del Multimarkdown (mmd)[8]. I due elementi del futuro file TEI sono quindi processati in modo indipendente, rispettivamente attraverso un ascii doc-like parser e un mmd parser; a partire dal file di partenza in formato .mmd si arriva quindi a un file .xml definito "proto-TEI", successivamente trasformato in file XML-TEI grazie a XSLT. Ecco in sintesi:

1. l'intero processo:

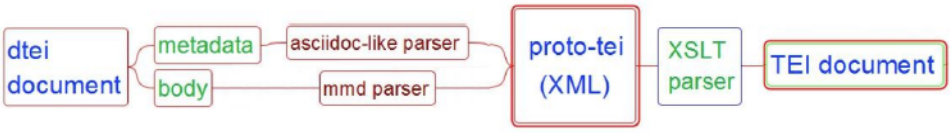

Figura 5: il processo di produzione TEIdown

2. la sintassi AsciiDoc per il teiHeader:

```
:title: Carta abierta
:author: Guiraldes, Ricardo
:publisher: Biblioteca Virtual Miguel de Cervantes Saavedra
:pubPlace: Universidad de Alicante
:idno: 004229
:availability: Copyright (c) Universidad de Alicante, Banco Santander Central Hispano. Accesible desde http://cervantesvirtual.com
:sourceDesc: Martín Fierro, segunda época, año II, núm. 14 y 15, Buenos Aires, enero 24 de 1925. Obra cedida por la Biblioteca de la Academia Argentina de las Letras. Digitalización realizada por Verónica Zumárraga.
```

Figura 6: il teiHeader in TEIdown

7 http://asciidoc.org

8 https://fletcherpenney.net/multimarkdown/

3. La sintassi Multimarkdown per il corpo del testo (per es. la presenza di una linea bianca genera il tag <p>):

```
# Carta abierta #
docAuthor: Ricardo Güiraldes

Amigos:

He leído hoy el último número de MARTíN FIERRO. Patria chica en el papel, grande en el anhelo.
Vuestra juventud sube hacia mi rostro, como un aliento de pampa, cuando sobre la gramilla iluminada
de rocío (emoción de la madrugada, que vuelve a encontrar su mundo) me aferro al optimismo
ascendente de los nuevos crecimientos. El hombre se siente pequeño ante la infinita transmutación
que anuncia lo porvenir, pero crece con sentirse capaz de comprenderla.
```

```
<text lang="es">
  <front>
    <titlePage>
      <docTitle>
        <titlePart>
          <title type="main">Carta abierta</title>
        </titlePart>
      </docTitle>
      <docAuthor>Ricardo Güiraldes</docAuthor>
    </titlePage>
  </front>
  <body>
    <div>
      <p>Amigos:</p>
      <p>He leído hoy el último
número de MARTíN FIERRO. Patria chica en el papel, grande en el
anhelo. Vuestra juventud sube hacia mi rostro, como un aliento de pampa, cuando
sobre la gramilla iluminada de rocío (emoción de la madrugada,
que vuelve a encontrar su mundo) me aferro al optimismo ascendente de los
nuevos crecimientos. El hombre se siente pequeño ante la infinita
transmutación que anuncia lo porvenir, pero crece con sentirse capaz de
comprenderla.</p>
```

Figura 7: il body in TEIdown e la sua corrispettiva conversione XML-TEI

Nelle parole degli stessi autori, per il momento:

> Este método sirve para realizar un marcado sencillo con elementos de prosa, poesía y algo de teatro, pero no está pensado para casos más especializados o complejos, como par ejemplo el marcado de manuscritos, ediciones críticas o análisis genético. Este marcado especializado siempre se puede agregar a posterior, una vez convertido el texto con marcado ligero a TEI. Incluso en estos casos, se ahorraría bastante tiempo en el marcado básico inicial. ([2]: 74)

I limiti sono evidenti, benché l'idea rappresenti già un passo in avanti e il prototipo sia destinato a futuri sviluppi:

> Las pruebas presentadas aquí se realizaron con un prototipo quick-and-dirty, a modo de

> prueba de concepto. Como trabajo futuro, tenemos planeado hacer una implementación completa y robusta de este método y hacerla disponible como servicio Web. ([2]: 75)

Sottolineo, perché mi sembra importante, la parola "método": l'obiettivo di queste esperienze, e dell'esperienza di TEI-Med, non è infatti la proposta di un *nuovo linguaggio* di markup testuale (come per es. nel caso di TAGML *Text-as-Graph Markup Language*[9]) bensì la proposta di una *nuova metodologia* per ottenere un markup XML-TEI corretto, e potenzialmente più ricco, in un modo più rapido e più semplice.

Benché innegabilmente affetto da numerosi difetti, fra cui spicca per gravità il problema concettuale dell'overlapping, l'XML resta malgrado tutto un linguaggio robusto, ben rodato, e per il quale esistono un'infinità di strumenti. Ma soprattutto, dal 1998, l'identificazione fra la codifica testuale, principalmente incarnata dalla TEI, e questo linguaggio è un dato di fatto: la proposta di un nuovo linguaggio di markup dovrebbe innanzitutto porsi il problema del rapporto con la massa documentaria prodotta durante questo ventennio. L'XML non ha certo rappresentato di per sé la migliore delle scelte possibili (verbosità e *overlapping* non sono certo peccati veniali), e in realtà le stesse *Guidelines* restano aperte al cambiamento (sottintendo in questo modo che l'XML rappresenta una delle incarnazioni tecniche possibili della filosofia TEI ma non la sua essenza):

> The rules and recommendations made in these Guidelines are expressed in terms of what is currently the most widely-used markup language for digital resources of all kinds: the Extensible Markup Language (XML) […] However, the TEI encoding scheme itself does not depend on this language; it was originally formulated in terms of SGML […] and may in future years be re-expressed in other ways as the field of markup develops and matures. ([4])

ma, forse, il risultato di un'analisi costi-benefici in caso di migrazione a un altro linguaggio potrebbe essere un'incognita piuttosto inquietante.

Il progetto TEI-Med

L'obiettivo del progetto TEI-Med è stato triplice: (1) creare una codifica particolarmente complessa (una codifica a livello delle parole, ciascuna identificata da un @xml:id), (2) in tempi rapidi, (3) attraverso un sistema che al tempo stesso semplificasse al massimo (nei limiti del possibile) il processo di scrittura della codifica e riducesse al minimo (nei limiti del possibile) l'errore. Il che ha significato, per conseguenza, dover innalzare al massimo il livello di automatizzazione del processo: minore è l'intervento delle dita del codificatore, maggiore è il grado di correttezza del codice.

Il processo di produzione TEI-Med si può semplificare in questo modo: TXT > parser TEI-Med, gestito attraverso l'interfaccia dell'editor TEI-Medit – XML-TEI.

9 https://huygensing.github.io/TAG/

Per essere più precisi, al momento sono previste due fasi di correzione, per cui il percorso dal file testo al file TEI si può articolare ancor meglio in: TXT > parser TEI-Med – TXT preTEI – parser TEI-Med – XML-TEI.

Il punto di partenza è dunque un 'semplice' file .txt realizzabile attraverso un 'semplice' Text Editor: dal mio punto di vista il codificatore è prima di tutto, e prima di essere codificatore, un individuo che, per svariati motivi, ha bisogno di concentrare la sua attenzione su un testo per sviscerarne forma e senso: si è cercato perciò di riportare l'ambiente di lavoro alle condizioni più 'umanistiche' possibili. In fondo, come ci ricorda il W3, XML "is a simple text-based format for representing structured information" ([11]: § *What is XML?*). Rapidità nella scrittura, maggiore leggibilità del testo rispetto alla sua versione XML esplicita, maggiore facilità negli interventi di correzione: la scrittura del testo codificato sotto forma di file di testo puro ha permesso di raggiungere questi obiettivi e di gestire, da parte di una sola persona, l'edizione di sei manoscritti ciascuno di circa 13.500 versi.

La trascrizione del documento è dunque codificata attraverso una sintassi di marcatura composta da:

1. simboli ASCII, soprattutto associati a un carattere: per es. * e ^ sono stati associati rispettivamente alle lettere ramiste con valore consonantico e a quelle con valore vocalico

```
*u = <c ana="#lram-cons">u</c> (o <choice><orig>u</orig><reg>v</reg></choice>)
^V = <c ana="#lram-voy">V</c> (o <choice><orig>V</orig><reg>u</reg></choice>)
```

2. *entities* fisse con struttura &*...;* (molto usate nel caso delle abbreviazioni): per es.

```
&n; = <expan corresp="#abb-tild-nas"><ex>n</ex></expan>
```

o

```
    <choice>
      <abbr><am><g ref="#tild-nas"/></am></abbr>
      <expan><ex>n</ex></expan>
  </choice>
```

o qualsiasi altra formulazione si scelga per la codifica delle abbreviazioni

3. *entities* con argomenti variabili con struttura &*…;($)*; per es.:

&*sub-ex-int;(parla,parola)* (dove la prima parola è stata espunta e la seconda scritta al di sopra dell'espunzione in interlinea) =

```
<sub>
   <del rend="expunged">
      <w xml:id="msl1w1">parla</w>
   </del>
   <add place="interline">
      <w xml:id="msl1w2">parola</w>
   </add>
</sub>
```

Simboli ASCII ed *entities* sono liberamente definiti dall'utente in un file .csv in cui ciascun elemento della sintassi deve essere associato all'espansione XML-TEI desiderata:

```
cb-a | <cb n="a"/>
pb | <pb n="$"/>
add-m | <add place="margin">$</add>
sub-ex-int | <sub><del rend="exp">$</del><add place="interl">$</add></sub>
```

Rispetto al modello proposto da Text::TEI::Markup l'utente ha quindi molta più libertà: TEI-Med fornisce il meccanismo di conversione, o mediazione, ma è l'utente a indicare sia l'input (secondo i propri gusti e abitudini mentali, mnemoniche e di scrittura; personalmente ho semplicemente ridotto i tag TEI a *entities*) che l'output (secondo la propria interpretazione della codifica TEI). La mediazione di TEI-Med è infatti sia una mediazione tra due formati informatici differenti, sia una mediazione tra l'umano e la macchina, nel tentativo di semplificare le cose per gli standard umani in modo da produrre qualcosa di più efficiente per gli standard della macchina. Inoltre, grazie alla esternalizzazione e personalizzabilità della sintassi è possibile associare eventualmente più formule di codifica XML-TEI allo stesso simbolo/*entity* (una classica alternanza <choice><reg> o una codifica alternativa basata su <c> come nell'esempio proposto per le lettere ramiste), semplicemente associando al .txt un differente .csv.

TEI-Med produce poi in automatico:

1. il tag <l> (lo strumento è nato nel contesto dell'edizione di un testo in versi, il trattamento della prosa richiederà alcune implementazioni);

2. il tag <w>;

3. @xml:id per tutti gli elementi e @n per alcuni elementi definiti.

Per quel che riguarda gli @xml:id ci si è posti il problema della alterazione del sistema di puntatori in caso di alterazioni degli @xml:id stessi ad es. per inserimento di una parola. Il problema non si pone nel caso della produzione degli @from @to (v. oltre) in quanto i valori degli attributi @from @to sono generati contestualmente alla produzione degli @xml:id: al variare del testo e quindi degli @xml:id, varieranno automaticamente anche i valori dei due attributi. Il problema si porrebbe invece nel caso in cui lo *stand-off markup* fosse creato autonomamente e posteriormente alla creazione del file XML-TEI. Si tratta di un

'inconveniente' per il quale non si è ancora trovato una soluzione; quel che si è potuto fare è stato concepire un modello di @xml:id che limitasse al massimo i danni. Al posto di un @xml:id progressivo e continuativo dal primo all'ultimo elemento del testo si è optato, nell'ambito di un testo in versi, per una forma di @xml:id di questo tipo: numero <lg> - numero <l> di ciascun <lg> - numero <w> di ciascun <l>:

```
<lg n="1" xml:id="lg1">
   <l n="1" xml:id="lg1l1">
        <w xml:id="lg1l1w1">
        <w xml:id="lg1l1w2">
        <w xml:id="lg1l1w3">
   <l n="2" xml:id="lg1l2">
        <w xml:id="lg1l2w1">
        <w xml:id="lg1l2w2">
        <w xml:id="lg1l2w3"> ecc.
```

In questo modo l'alterazione non è eliminata ma, perlomeno, ha un impatto perburbatore su una porzione di testo molto ridotta e non sull'intero sitema.

Riprendendo il verso di esempio proposto in apertura, questo codice XML:

```
<l n="3" xml:id="msl3">
   <w xml:id="msl3w1">Qua<expan corresp="#abb-tild-
nas"><ex>n</ex></expan>t</w>
   <w xml:id="msl3w2" ana="#aggl-s">lo</w>
   <w xml:id="msl3w3"><expan corresp="#abb-tir-9"><ex>con</ex></expan>
      <expan corresp="#lsup-q">q<ex>u</ex>i</expan>ert</w>
   <w xml:id="msl3w4" ana="#elis">n</w>
   <w xml:id="msl3w5" ana="aggl-s-unc">a</w>
   <w xml:id="msl3w6"><c ana="#lram-cons">i</c>oie</w>
   <w xml:id="msl3w7"><c ana="#lram-cons">u</c>aine</w>
</l>
```

è stato ottenuto a partire da questa stringa testuale:

```
Qua&n;t +lo &9;&qi;ert 'n +_a *ioie *uaine
```

Nel dettaglio, il viaggio dal file .txt al file .xml si realizza quindi attraverso queste tappe:

1. produzione del file .txt secondo i criteri della sintassi TEI-Med e caricamento in TEI-Medit, dove viene associato al file .csv;

Prima trasformazione operata dal parser TEI-Med:

2. *teimxml.py* sostituisce simboli ed *entities* con i tag corrispondenti leggendo dal file .csv compilato dall'utente;

3. viene prodotto un file di log che segnala gli eventuali errori a livello di scrittura dei simboli/*entities* e degli argomenti; si può così procedere a una correzione del file .txt e poi rilanciare il processo;

Seconda trasformazione TEI-Med:

4. *teimlineword.py* aggiunge il tag <l> per ogni riga e il tag <w> per ogni parola; numera i tag

per cui si desidera l'attributo @n (è possibile settare il numero iniziale da cui partire); attribuisce gli @xml:id;

5. *teimspan.py* è attivato per uno *stand-off markup*: per codificare il discorso diretto e i monologhi è stato utilizzato infatti uno *stand-off* attraverso il tag @from @to, collocato alla fine di ogni <lg>; a livello di file .txt il punto di apertura e chiusura dello span, e dunque i valori degli attributi @from @to, sono segnalati tramite delle parentesi che racchiudono la porzione di discorso diretto/monologhi;

6. *teimnote.py* aggiunge al file .xml le note (ancorate al testo tramite <ptr>) che, al momento della scrittura del file .txt sono stoccate in un .csv. Dal punto di vista della codifica TEI le note editoriali sono state codificate utilizzando il sistema-pointer <ptr> @ref / <note> @target, che permette di ancorare la nota al testo in modo analogo al sistema tipografico senza 'contaminare' il testo originale. A livello di sintassi semplificata TEI-Med le note vengono scritte in un file .csv esterno, a due colonne: nella prima è indicato il @target identificativo della nota, nella seconda il corpo della nota; nel testo si inserirà invece l'*entity* &ptr($), che avrà come argomento il valore del @target.

7. *teimxmllint.py* formatta e controlla il file .xml;

8. viene quindi prodotto un file di log che segnala gli eventuali errori a livello di sintassi XML;

9. la correzione viene realizzata di nuovo nel file .txt (assai più facile da correggere di un file .xml) e il processo rilanciato;

Arrivo:

10. il file XML-TEI è pronto. Di fronte a un file .txt totalmente privo di errori, l'intero processo non dura che qualche secondo.

Per riassumere:

1. trascrizione codificata iniziale:

```
&div-epB;(ep12)
&pb;(28r)
&cb-b;
&lgB;
&pers;(RIS,&hi-f;(R)^Jsus) que*uauque &7; sa gent
Si nen aloit mie lent.
Par_mi le &geog-f1;(VDB,*ual,de,bri) paserent
```

2. dopo la prima trasformazione:

```
<div type="episode" ref="#ep12">
<pb n="28r"/>
<cb n="b"/>
<lg>
<persName ref="#RIS">
<forename><hi rend="init-flour">R</hi><c ana="#lram-v">J</c>sus</forename>
</persName> que<c ana="#lram-c">u</c>auque
<expan corresp="#ab-tir-7"><ex>et</ex></expan> sa gent
```

```
Si nen aloit mie lent<pc resp="#ed">.</pc>
Par_mi le <geogName ref="#VDB"><geogFeat>
<c ana="#lram-c">u</c>al</geogFeat>de<name>bri</name></geogName> paserent
```

3. dopo la seconda trasformazione:

```
<TEI>
  <div type="episode" ref="#ep12">
    <pb xml:id="gpb1" n="28r"/>
    <cb xml:id="gcb1" n="b"/>
    <lg xml:id="glg1">
      <l n="1" xml:id="glg1l1">
        <persName ref="#RIS">
          <forename>
            <w xml:id="glg1l1w1">
              <hi rend="init-flour">R</hi><c ana="#lram-v">J</c>sus
            </w>
          </forename>
        </persName>
        <w xml:id="glg1l1w2">que<c ana="#lram-c">u</c>auque</w>
        <w xml:id="glg1l1w3">
          <expan corresp="#ab-tir-7">
            <ex>et</ex>
          </expan>
        </w>
        <w xml:id="glg1l1w4">sa</w>
        <w xml:id="glg1l1w5">gent</w>
      </l>
      <l n="2" xml:id="glg1l2">
        <w xml:id="glg1l2w1">Si</w>
        <w xml:id="glg1l2w2">nen</w>
        <w xml:id="glg1l2w3">aloit</w>
        <w xml:id="glg1l2w4">mie</w>
        <w xml:id="glg1l2w5">lent</w>
        <pc xml:id="glg1l2pc1" resp="#ed">.</pc>
      </l>
      <l n="3" xml:id="glg1l3">
        <w xml:id="glg1l3w1">Par mi</w>
        <w xml:id="glg1l3w2">le</w>
        <geogName ref="#VDB">
          <geogFeat>
            <w xml:id="glg1l3w3"><c ana="#lram-c">u</c>al</w>
          </geogFeat>
          <w xml:id="glg1l3w4">de</w>
          <name>
            <w xml:id="glg1l3w5">bri</w>
          </name>
        </geogName>
        <w xml:id="glg1l3w6">paserent</w>
      </l>
```

L'esperimento LaTex2TEI

Il principio di mediazione alla base di TEI-Med è stato successivamente messo alla prova con

un'altra tipologia di file di input, un file LaTeX (.tex), ed è stata quindi creata una variante chiamata LaTex2TEI (LaTeX to TEI). L'occasione è stata offerta da una collaborazione informale con la prof.ssa Rosanna Cantavella, direttrice del progetto *Ausiàs March (Edición sinóptica de las poesías de Ausiàs March a partir de todos los testimonios manuscritos e impresos en su contexto literario)* promosso dall'Universitat d'Alacant[10], un progetto che ha visto la digitalizzazione iniziale dei testi in formato LaTeX e che prevede oggi la trasformazione di questi files (nell'ordine delle centinaia) in formato XML-TEI.

Il fatto di doversi confrontare con il passaggio da un sistema di codifica 'orientato alla *mise en page*' a un sistema 'orientato alla semantica' ha rappresentato un'occasione di riflessione intorno alla questione della codifica testuale e delle conseguenze (a volte pesanti) legate alla scelta della tipologia di markup; ma soprattutto ha messo in evidenza i limiti potenziali dei processi di conversione. In questo caso il problema è rappresentato dalla multivalenza semantica dei tag LaTeX, di natura tipografica, multivalenza che rende di fatto impossibile un processo di automatizzazione completa della conversione. Per es. il tag \textit{$} è utilizzato in una molteplicità di situazioni testuali di natura differente (e anche assai distante) che impiegano però l'espediente grafico del corsivo per esprimere i metadati: titoli, lettere restituite nelle abbreviazioni, evidenziazioni ecc. Questa situazione di ambiguità rispetto ad alcuni tag obbliga inevitabilmente a prevedere un successivo intervento a mano sul file XML-TEI per risolvere i conflitti. Data però l'entità della massa documentaria da trasformare, l'obiettivo è stato comunque quello di facilitare il lavoro cercando di ridurre al minimo questo intervento successivo.

L'esperimento, un puro prototipo molto legato al progetto specifico e alla casistica offerta da questi testi, ha avuto nel complesso un esito positivo e ha confermato la bontà (potenziale) del metodo. Complessivamente è stato dunque applicato il principio di equivalenza segni/tag XML-TEI alla base di TEI-Med ma in questo caso i segni sono stati sostituiti dai tag LaTeX, per es.:

```
\begin{estrofa}= <lg type="cobla" n="$" xml:id="$">
\end{estrofa} = </lg>
\numero{$} = <l n="$" xml:id="$"> </l>
/ = <seg type="hemistiquio" xml:id="$"> </seg>
\interlin{$} = <add place="interlinear">$</add>
```

La maggior parte degli interventi viene realizzata a livello del file .tex, sicuramente più facile da gestire del file .xml: si tratta di eliminare alcuni elementi e modificare alcuni dettagli che permettono di risolvere alcune situazioni di conflitto. Un file .tex di questo tipo, con alcuni interventi di cancellazione, indicati da ~~###~~:

```
~~#####~~ \documentclass[12pt]{article}
~~#####~~ % ------------------------------------------------------
~~#####~~ \input{../preamble}
~~#####~~\renewcommand{\espaiAbansEtiquetaPoema}{\vspace{0ex}}
~~#####~~ \renewcommand{\espaiAbansSeccio}{\vspace{0ex}}
~~#####~~ % ------------------------------------------------------
```

10 http://www.cervantesvirtual.com/portales/ausias_march/

```
\begin{document}

~~/////~~ \begin{estrofa}
~~/////~~ \espaiAbansEtiquetaPoema
~~/////~~ \poema{(b 31) [1]}\\
~~/////~~ \end{estrofa}

\begin{estrofa}
\numero{1} [25r] Axi com cell / qui'n lo somnis de lita
\numero{2} Del temps p\textit{re}sent / nom trobe amador
```

attraverso la conversione LaTex2TEI ha dato vita automaticamente a questa codifica XML-TEI:

```
<text>
  <body>
    <div type="poem" xml:id="Adiv1">
      <pb n="49r" facs=""/>
         <lg type="cobla" n="1" xml:id="Alg1">
           <l n="1" xml:id="Al1">
               <seg type="hemistiquio" xml:id="Al1s1">AXicom cell:</seg>
               <seg type="hemistiquio" xml:id="Al1s2">quin lo sompnis delita
               </seg>
           </l>
           <l n="2" xml:id="Al2">
               <seg type="hemistiquio" xml:id="Al2s1">Del temps
               <choice><abbr>p˜sent</abbr><expan>present</expan></choice>
               </seg>
               <seg type="hemistiquio" xml:id="Al2s2">nom trobe amador
               </seg>
           </l>
```

Obiettivi futuri

Il progetto TEI-Med deve, naturalmente, evolversi e passare definitivamente dallo stadio di prototipo a quello di strumento solido e stabile. La prospettiva, per convinzione personale, è comunque quella di mettere a punto uno strumento autonomo da consegnare all'utente interessato e non un servizio Web (come nel caso previsto dal progetto TEIDown): dal mio punto di vista l'infrastruttura Internet resta essenzialmente un'infrastruttura che permette lo scambio di dati (a partire dal *Login* rimasto in sospeso esattamente cinquant'anni fa) e non uno spazio di lavoro. L'ergonomia di TEI-Medit deve sicuramente migliorare, sopratutto nella gestione della fase di correzione: si potrebbe ad esempio riflettere sull'utilità di realizzare in una sola volta l'intero processo di conversione senza spezzarlo nelle due fasi di esplosione delle *entities* e successiva generazione del resto dell'XML. E sicuramente alcune trasformazioni dovranno essere gestibili dall'utente come opzioni: la codifica a livello di parole, l'assegnazione di @n e @xml:id. Un testo in prosa porrà infine una serie di problemi che non sono ancora emersi. Ma per quel che riguarda TEI-Med e TEI-Medit si può sicuramente affermare che la quantità di lavoro già effettuata è intorno al settanta per cento e che lo strumento può già funzionare correttamente in un contesto diverso che presenti la medesima tipologia testuale, cioè un testo in versi. Il prototipo LaTex2TEI, invece, deve ancora uscire dalla fase larvale del

primo abbozzo e richiederà quindi molto più lavoro per passare dal funzionamento corretto in un contesto ultra-specifico al funzionamento corretto in un contesto qualsiasi: per il momento il programma è stato tagliato su misura sui testi *Ausiàs March*. Ma, oltre alle messe a punto del già esistente, il progetto TEI-Med mira anche a espandersi. In particolare l'aspirazione è quella di aggiungere un ulteriore passaggio al processo al fine di ottenere un'ulteriore conversione automatica: la conversione del file .xml in un file .html pronto per la pubblicazione, evitando il passaggio attraverso XSLT e sfruttando lo stesso principio di equivalenza fra tag utilizzato nella fase precedente. Lo strumento TEI-Medit e tutta la documentazione e i codici-sorgente saranno messi a disposizione nello spazio GitHub https://github.com/florimont

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowslka-Curie grant agreement N° 745821

References

[1] Andrews, T. 2014. *Text-Tei-Markup-1.9*. https://metacpan.org/pod/Text::TEI::Markup

[2] Bia Platas, A. and R. Ñeco García. 2017. "TEIdown: Uso de Markdown extendido para el marcado automático de documentos TEI." *Revista de Humanidades Digitales (RHD)* 1:57-75. DOI: https://doi.org/10.5944/rhd.vol.1.2017.16683

[3] Busa, R. 1996. "Introduzione ai lavori." In *Umanesimo e informatica. Le nuove frontiere della ricerca e della didattica nel campo degli studi letterari,* conference (Trento, Università degli studi di Trento, 24-25 maggio 1997). http://circe.lett.unitn.it/attivita/eventi/pdf_eventi/busa.pdf

[4] Guidelines TEI. iv. *About These Guidelines* https://www.tei-c.org/release/doc/tei-p5-doc/en/html/AB.html

[5] Pierazzo, E. 2011. "A Rationale of Digital Documentary Editions." *Literary and Linguistic Computing* 26.4:463-477.

[6] Robinson, P. 2004. "Where We Are with Electronic Scholarly Editions, and Where We Want to Be." *Jahrbuch für Computerphilologie Online* 1.1. http://computerphilologie.uni-muenchen.de/jg03/robinson.html

[7] Robinson, P. 2005. "Current Issues in Making Digital Editions of Medieval Texts–Or, Do Electronic Scholarly Editions Have a Future?" *Digital Medievalist* 1.1. http://www.digitalmedievalist.org/journal/1.1/robinson http://doi.org/10.16995/dm.8

[8] Rosselli del Turco, R. 2016. "The Battle We Forgot to Fight: Should We Make a Case for Digital Editions?" In *Digital Scholarly Editing: Theories and Practices*, Cambridge: Open Book Publisher. http://books.openedition.org/obp/3423>.

[9] Sahle, P. 2016. "What is a Scholarly Digital Edition?" In *Digital Scholarly Editing: Theories and Practices.* Cambridge: Open Book Publisher:

http://books.openedition.org/obp/3423>.

[10] Stolz, M. 2017. "Copying, Emergence and Digital Reproduction. Transforming Medieval Manuscript Culture into an Electronic Edition." *Digital Philology* 6.2:257-87.

[11] *W3C. Standards. XML Technology. What is XML?.* Webpage. https://www.w3.org/standards/xml/core