



## Livelli di rappresentazione del testo nell'edizione del *De nomine* di Orso Beneventano

Paolo Monella

Dipartimento Culture e Società, Università degli Studi di Palermo, Palermo, Italy  
[paolo.monella@gmx.net](mailto:paolo.monella@gmx.net)

**Abstract.** This paper presents the methodological and technological rationale of the scholarly digital edition of the *De nomine* by Ursus Beneventanus, a proof of concept of methodological innovations introduced by Tito Orlandi. The proposed model goes beyond the dichotomy between diplomatic and interpretative edition. The transcription layers are defined on the basis of semiotic considerations. This edition includes three layers: (1) A *graphematic layer*, whose minimal units are graphemes, including paragraphematic signs (such as punctuation) and systematic one-glyph abbreviations (such as “*p*” for “per”). All graphemes identified by the editor are listed in a “Graphematic Table of Signs” (GToS), a functional part of the edition. (2) An *alphabetic layer*, whose minimal units are alphabetic letters. The GToS provides the standard alphabetic meaning of graphemes. The editor encodes the alphabetic transcription explicitly (within <expan>) only when the software cannot generate it based on the graphematic transcription through the grapheme/alphabetic letter mapping in the GToS. (3) A *linguistic layer*, whose minimal units are inflected words (<w>), identified through a lemma/morphology combination: the lemma (e.g. “lupus, -i”) is encoded with the attribute @lemma; morphological information (e.g. “genitive singular”), with the attribute @ana. This layer is for “interpretative” visualization, textual analysis, interoperability and collation. A final section of the paper analyzes the strong and weak points of the Ursus edition and discusses strategies to expedite the workflow, in view of the future edition of the *Chronicon* by Romualdus Salernitanus.

**Abstract.** Viene presentato l'impianto metodologico e tecnologico dell'edizione critica digitale del *De nomine* di Orso Beneventano, una *proof of concept* delle innovazioni metodologiche proposte da Tito Orlandi. Il modello qui sperimentato intende superare la dicotomia tra edizione diplomatica e critica grazie ad un ripensamento profondo, semioticamente fondato, dei livelli di trascrizione e di edizione di un testo. I livelli scelti per questa edizione sono: (1) *Livello grafematico*, le cui unità minime sono i grafemi, inclusi i segni paragrafematici e le brachilogie sistematiche. Tutti i grafemi individuati dall'editore sono elencati nella “Graphemic Table of Signs” (GToS), che costituisce parte integrante dell'edizione. (2) *Livello alfabetico*, le cui unità minime sono le lettere alfabetiche. La GToS riporta il significato alfabetico standard dei grafemi. La codifica a livello alfabetico è fornita esplicitamente dall'editore all'interno di un elemento <expan> solo quando

essa non sia desumibile dalla codifica grafematica tramite la tabella di corrispondenze fornita nella GToS. (3) *Livello linguistico*, le cui unità minime sono le parole flesse (<w>), identificate tramite la combinazione di un lemma (“lupus, -i”: attributo @lemma) e di informazioni morfologiche (“genitivo singolare”: attributo @ana). Questo livello di codifica è utile per la visualizzazione (edizione “critica”), l’analisi testuale, l’interoperabilità e la collazione. L’ultima parte dell’articolo traccia un bilancio dell’edizione di Orso e delinea alcune strategie per velocizzare il flusso del lavoro in vista della futura edizione del *Chronicon* di Romualdo Salernitano.

**Keywords:** Digital Humanities, Digital Philology, Filologia digitale, Informatica umanistica, Scholarly digital edition, TEI, Ursus Beneventanus, XML

## 1 Introduzione

Il saggio discute l’impianto metodologico e tecnologico dell’edizione critica digitale del *De nomine* di Orso Beneventano (<http://www.unipa.it/paolo.monella/ursus>), un trattatello grammaticale latino inedito del IX secolo che compendia grammatici tardo-antichi come Prisciano e Donato, tramandato da un *codex unicus* in scrittura beneventana del IX secolo, il *Casanatensis* 1086.<sup>1</sup>

L’edizione è un prototipo che intende mettere alla prova ed approfondire le innovazioni metodologiche proposte da Tito Orlandi nel campo dell’ecdotica digitale.

L’articolo si conclude con un bilancio conclusivo dell’esperimento. Per una trattazione più completa degli aspetti informatico-umanistici si veda la documentazione in inglese del progetto sul repository GitHub “paolomonella/ursus”.<sup>2</sup>

## 2 Il modello dell’edizione

### 2.1 Il modello orlandiano di edizione critica digitale

La prima innovazione del modello di edizione critica digitale delineato da Orlandi è la rappresentazione del testo del documento a più livelli testuali, *semioticamente fondati*.<sup>3</sup>

Il punto di partenza teorico dello studioso è la concezione del manoscritto come “diasistema”, elaborata da Segre ed applicata da Orlandi all’ambito informatico-umanistico:

<sup>1</sup> Vd. [17], [18] e [7].

<sup>2</sup> <https://github.com/paolomonella/ursus/blob/master/documentation.md>.

<sup>3</sup> [20] costituisce una sistematizzazione organica delle riflessioni dello studioso, presentate in numerosi contributi precedenti (ricordo qui solo [21] e [22], che si fonda su [28]). Lo stesso Orlandi ha creato un piccolo prototipo del suo modello con l’“Edizione digitale sperimentale” parziale di Niccolò Machiavelli, *De principatibus* [19], che riporta l’esergo “Lasciate ogni speranza, voi ch’entrate. Esperimento raccomandato ad un pubblico di soli studiosi adulti”. Pare che le ammonizioni abbiano effettivamente sortito l’effetto di scoraggiare i visitatori, perché l’esperimento, a mio parere molto interessante, è completamente ignorato nella comunità dell’Informatica umanistica all’estero, ed anche in Italia. Col risultato che le idee di Orlandi vengono tacciate di astrattezza e si argomenta spesso che non ne esista una realizzazione – e che quindi siano probabilmente irrealizzabili.

il testo di un manoscritto è un sistema complesso, composto dall'interazione tra sotto-sistemi, tra cui quello glifico, quello grafematico, quello linguistico, e potenzialmente anche molti altri.<sup>4</sup>

Per esempio, l'insieme dei grafemi costituisce un *sistema* in quanto ognuno di essi si definisce in opposizione a tutti gli altri – come fa sempre un elemento in un sistema di segni.<sup>5</sup> Lo stesso vale per il sistema dei glifi, per quello linguistico – i cui elementi minimi sono le parole flesse – e così via.

I due (sotto-)sistemi su cui particolarmente insiste Orlandi [20] sono quello che qui chiamerò “grafematico” (Orlandi spesso lo chiama più genericamente “grafico”) e quello “linguistico”.

Il *sistema grafematico* è composto da grafemi aventi valore distintivo, ciascuno rappresentato graficamente da vari glifi, che appartengono al “sottostante” *sistema glifico*: “Intendiamo per glifo il disegno reale, materiale, di un grafema; il grafema è il modello astratto dei glifi, l'invariante rispetto alle variazioni dei singoli glifi”.<sup>6</sup>

All'estremo opposto, nella visione orlandiana, sta il *sistema linguistico*, costituito da parole concepite sinteticamente, non come sequenza di lettere ma come entità uniche, codificate univocamente a prescindere dalle variazioni glifiche ed ortografiche.

La seconda innovazione metodologica proposta da Orlandi è costituita dalla “tabella dei segni”, ovvero una lista dei glifi del manoscritto codificati dall'editore.

## 2.2 Il modello di edizione di Orso Beneventano

I sotto-sistemi, o, come li chiamerò d'ora in poi, i *livelli di rappresentazione* del testo sono potenzialmente infiniti, e la scelta di quelli da rappresentare nell'edizione critica digitale dipende unicamente dagli interessi scientifici dell'editore: si potrebbero codificare gli allografi dei grafemi, incluse le legature e gli allografi di posizione, i grafemi, le lettere alfabetiche, i fonemi, le parole flesse, i lemmi, i concetti, con tutte le infinite sfumature e i problemi di definizione che si aprono a ciascuno di questi livelli.<sup>7</sup> Per il mio lavoro su Orso Beneventano ho scelto di rappresentare tre sotto-sistemi del testo, cui corrispondono tre livelli di trascrizione/edizione:

1. Livello grafematico;
2. Livello alfabetico (non presente nel modello di Orlandi);
3. Livello linguistico.

I grafemi individuati in quanto aventi valore distintivo nel manoscritto vengono elencati in una tabella, la Graphemic Table of Signs (GToS), di cui pure parlerò diffusamente nel corso dell'articolo.

<sup>4</sup> Sui “sistemi” grafico e linguistico nel manoscritto, vedi il già citato [20], ma resta interessante, a proposito del manoscritto come “diasistema” (sistema di sistemi), anche il precedente [22], che si fonda su [28].

<sup>5</sup> Vd. [13].

<sup>6</sup> [22], paragrafo *I supporti dell'informazione*.

<sup>7</sup> La questione del *cosa* stiamo codificando si pone necessariamente a chi lavora effettivamente sui manoscritti: il *Canterbury Tales Project* se l'era posto già con [25], e poi ancora in [26].

### 3 Caratteri generali del markup

#### 3.1 TEI XML e interoperabilità

Prima di illustrare nel dettaglio come ognuno dei tre livelli venga rappresentato nel codice sorgente dell'edizione digitale, varrà la pena di esporre brevemente le scelte generali relative alla codifica dell'edizione.

Ho scelto il linguaggio di markup TEI XML, versione P5,<sup>8</sup> in quanto esso è considerato oggi lo standard *de facto* per la filologia digitale e perché il codice dell'edizione, messo a disposizione liberamente sul web e su GitHub,<sup>9</sup> possa essere più facilmente studiato o riutilizzato in vista di ulteriori ricerche o applicazioni.

Tuttavia tale riusabilità del codice, o “interoperabilità”, a mio parere non va intesa in senso semplicistico: succede assai raramente che il codice sorgente proveniente da un progetto A di edizione digitale, anche se codificato in TEI XML, possa essere processato dal software di un progetto B senza bisogno di alcun intervento umano mantenendo tutte le funzionalità previste dal progetto A e tutte quelle richieste del progetto B.

Con ciò non intendo dire che l'interoperabilità sia impossibile, ma solo che essa richiede che il progetto A abbia documentato le proprie scelte di codifica e che il team del progetto B, letta la documentazione e visto il codice del progetto A, adatti i propri algoritmi alle specifiche del codice sorgente ereditato. Se la documentazione del primo progetto sarà stata esaustiva e il suo codice (ad esempio TEI XML) concepito ed applicato in modo coerente, il processo sarà senz'altro fattibile.

La codifica TEI XML dell'edizione di Orso Beneventano è strutturata intorno ad elementi <w>, all'interno dei quali i tag relativi alla trascrizione di fonti primarie (in particolare elementi come <abbr> e <expan>) subiscono una specificazione semantica fondata sulla semiotica, in particolare la grafologia,<sup>10</sup> e sui recenti sviluppi della paleografia digitale.<sup>11</sup>

Il linguaggio TEI XML permette allo studioso di personalizzare lo schema di codifica a seconda delle esigenze del proprio progetto. Io ho scelto invece, proprio in vista dell'interoperabilità, di non modificare in nulla lo schema TEI: le novità sono tutte in termini di semantica dei tag e di modello di edizione, non di schema di codifica. Per questo la documentazione costituisce parte integrante – anzi fondamentale – dell'edizione stessa.

#### 3.2 Il protocollo CTS per la citazione delle fonti

Un altro aspetto notevole della codifica, oltre ai livelli di rappresentazione del testo di cui dirò in dettaglio qui di seguito, è costituito dal riferimento alle fonti di Orso Beneventano.

L'opera di Orso è infatti un'epitome/parafrasi di testi grammaticali tardo-antichi come la *Institutio de arte grammatica* di Prisciano, la *Ars minor* di Elio Donato, il *Commentum Artis Donati* di Pompeo, il *Commentarius in Artem Donati* di Servio e la *Explanatio in Artem Donati* di Sergio.

<sup>8</sup> Si vedano le Guidelines di TEI XML, versione P5, in <http://www.tei-c.org/>.

<sup>9</sup> Vedi il repository <https://github.com/paolomonella/ursus>.

<sup>10</sup> Cfr. [27], [1] e [16].

<sup>11</sup> Vd. [29].

Per marcare la fonte di una porzione del testo ho usato l'elemento TEI <ref>, come segue:

```
<ref cRef="urn:cts:latinLit:stoa0234a.stoa001:2.53.8-2.53.12" type="source">
```

Il valore dell'attributo @cRef è un URN<sup>12</sup> che segue il protocollo CTS-CITE.<sup>13</sup> Questo mira a creare codici identificativi per passaggi testuali che siano univoci e processabili da un computer. Nell'esempio sopra, il riferimento è a Prisciano di Cesarea (stoa0234a), *Institutio de arte grammatica* (stoa001), da 2.53.8 (volume 2, pagina 53, rigo 8) a 2.53.12 (stessa pagina, rigo 12).<sup>14</sup>

## 4 Livello grafemico

### 4.1 Unità minime

Le sue unità minime sono i grafemi (glifi aventi valore distintivo: vedi sotto, *Section 12*) specifici del sistema grafico del testimone, inclusi i segni paragrafematici (come la punteggiatura e gli spazi tra le parole grafiche) e le brachilogie sistematiche (come il brevi-grafo “p”, che sistematicamente significa le tre lettere alfabetiche “p”, “e” e “r”).<sup>15</sup>

<sup>12</sup> Un URN (Universal Resource Name) è un identificativo univoco per un certo oggetto (in questo caso una porzione del testo-fonte) secondo un protocollo formale (in questo caso l'architettura CTS-CITE).

<sup>13</sup> CTS sta per *Canonical Text Service*, CITE per *Collections, Indexes, Extensions*. Si vedano *The CITE architecture* nel sito del progetto *Homer Multitext* (<http://www.homermultitext.org/hmt-doc/cite/index.html>) e [30], con ulteriore bibliografia. Il protocollo CTS sta subendo attualmente una revisione importante, col fine di farlo evolvere nel nuovo protocollo DTS (*Distributed Text Services*): i materiali di lavoro relativi sono in <https://github.com/distributed-text-services>. Alcune mie riflessioni sulle implicazioni dell'utilizzo del protocollo sono in [14], 151-158 (paragrafo *A forma di corpus: CTS*).

<sup>14</sup> I testi di questi grammatici tardo-antichi non sono inclusi nel *corpus* PHI 5.3 del Packard Humanities Institute, che è il punto di riferimento per la letteratura latina, da cui vengono definiti i CTS URN per i testi latini classici. Sono dovuto quindi ricorrere, per ottenere i CTS URN questi autori, al canone contenuto nel documento *CTS Schemes Latin* disponibile in [https://docs.google.com/spreadsheets/d/1VPdW\\_upQtP9voPq-fZW\\_McfyPTfR5Lr-jFGmSubffLA/edit#gid=1284785699](https://docs.google.com/spreadsheets/d/1VPdW_upQtP9voPq-fZW_McfyPTfR5Lr-jFGmSubffLA/edit#gid=1284785699). Questo costituisce uno strumento di lavoro del gruppo che sta sviluppando *The Perseus Catalog* (<http://catalog.perseus.org/>). Infatti molti (ma non tutti) degli autori e delle opere catalogate in *CTS Schemes Latin* sono già stati riversati nel *Perseus Catalog*.

<sup>15</sup> Sulla grafologia, intesa come studio dei sistemi grafici come sistemi semiotici autonomi, si vedano [27], [1], [16] e [29]. Quello di Stokes, all'interno del progetto DigiPal <http://www.digipal.eu/>, è un lavoro importante per la fondazione di una paleografia digitale consapevole delle questioni semiotiche di fondo.

## 4.2 Graphemic Table of Signs (GToS) e valore distintivo

Tutti i grafemi che ho individuato nel *Casanatensis* 1086 sono elencati e commentati nella “Graphemic Table of Signs” (GToS), un file CSV distinto dalla trascrizione TEI XML ma che costituisce parte integrante dell’edizione (file `GtoS.csv`).<sup>16</sup>

Vale la pena di sottolineare che la tabella non intende includere tutti i *glifi*, il che sarebbe impossibile, ma tutti i *grafemi*, ovvero i glifi aventi valore distintivo: perché un glifo (o un insieme di allografi) sia considerato un grafema bisogna che esista almeno una coppia minima di parole che differiscano per quel solo grafema.

Riporto per chiarezza alcune righe e colonne del file `GtoS.csv`:<sup>17</sup>

**Table 1.** Estratto della Graphemic Table of Signs

Grapheme	Alphabeme(s)	Grapheme visualization	Type	Notes	Image(s)
e	e	e	Alphabetic		e1.png e2.png
æ	ae	ę	Alphabetic	E caudatum.	ecaudatum1.png ecaudatum2.png
ł		;	Punctuation	Punctuation sign looking like a modern semicolon (;).	punct_semicolon.png
7		-	Abbreviation mark	Written over another grapheme, mostly for a missing “m”.	abbr_7.png
þ	per	p	Brevigraph	“p” with a horizontal stroke crossing the descending trait.	brev_per.png
÷	est	≈	Logograph	A dash or tilde with one dot above and a comma below, meaning the whole linguistic word “est”.	log_est.png

Qualche chiarimento sulle colonne:

- La colonna “Grapheme” include il carattere Unicode usato nel sorgente TEI XML della trascrizione per rappresentare univocamente il grafema: non importa la somiglianza con il glifo (o con gli allografi) manoscritti, ma l’univocità, la coerenza nell’uso e la facilità di digitazione tramite combinazione di tasti.

<sup>16</sup> Il file `GToS.csv`, come tutti quelli che costituiscono l’edizione digitale, è disponibile nel repository GitHub del progetto. Il suo URL è <https://github.com/paolomonella/ursus/blob/master/GToS.csv>.

<sup>17</sup> Ho scelto e riportato qui solo alcune righe con l’obiettivo di esemplificare i diversi tipi di grafemi, ho omesso alcune colonne e ho abbreviato per motivi di spazio il contenuto del campo “Notes” per il grafema codificato col carattere Unicode “f” (U0142+).

- La colonna “Grapheme visualization” indica al software che fa girare l’edizione quale carattere Unicode utilizzare per visualizzare il grafema in HTML nel browser.<sup>18</sup>
- Invece le immagini degli allografi, ritagliate dalle riproduzioni fotografiche digitali del manoscritto, sono file PNG richiamati nella colonna “Image(s)”. Ad esempio, il grafema “æ” (“e caudatum”) è rappresentato nel manoscritto da due allografi, mostrati nei file `ecaudatum1.png` e `ecaudatum2.png`:



Fig. 1. File `ecaudatum1.png`



Fig. 2. File `ecaudatum2.png`

Poche precisazioni sui tipi di grafema (colonna “Type”):

- “Abbreviation mark” indica un grafema che si combina con un grafema-base di tipo “alphabetic” per creare abbreviazioni complesse (come “c̄fer” per “confer”);
- “Brevigraph”: un grafema-abbreviazione che da solo significa una sequenza di alfabetici (“*p*” ha come significato alfabetico immediato “per”);
- “Logograph”: un grafema-abbreviazione che da solo significa un’intera parola linguistica.

Della colonna “Alphabeme(s)” dirò più sotto, nel paragrafo Section 5.3.

### 4.3 Codifica TEI XML

Nei file TEI XML contenenti la trascrizione/edizione del manoscritto (`casanatensis.xml` e `lemmatized_casanatensis.xml`),<sup>19</sup> tutti i caratteri Unicode contenuti in `<w>` (tranne quelli contenuti in `<expan>`) rappresentano grafemi.

Si prenda ad esempio questo frammento di codice del file `lemmatized_casanatensis.xml`:

```
<w ana="3-KA1--6---" lemma="pertineo" n="pertinet" xml:id="w81">
  <choice>
    <abbr type="brevigraph">p</abbr>
    <expan>per</expan>
  </choice>tinet</w>
```

<sup>18</sup> Vedi il paragrafo *Section 4 / Section 4.4* sotto.

<sup>19</sup> Per la differenza tra i due file, vedi sotto il paragrafo Section 6.

In esso, i caratteri che rappresentano grafemi sono quelli qui evidenziati in grassetto, ovvero “**p**tin<sup>et</sup>”.<sup>20</sup>

Tutti i caratteri utilizzati per indicare grafemi nel sorgente TEI XML devono rientrare tra quelli elencati nella colonna “Grapheme” della GToS.

#### 4.4 Visualizzazione HTML

Lo script JavaScript `jsparser.js`<sup>21</sup> prende come input i file `lemmatized_casana_tensis.xml` e `GtoS.csv`, e per ogni grafema nel file TEI XML cerca nella colonna *Grapheme visualization* nel file CSV il carattere Unicode da usare per la sua visualizzazione.

Nel caso del frammento di codice TEI XML riportato nel paragrafo precedente, quando lo script `jsparser.js` legge il carattere Unicode “**p**”, esso individua la riga corrispondente nel file `GtoS.csv`, vi legge che il carattere per la visualizzazione è “*p*” ed usa quest’ultimo per visualizzare il grafema nel file HTML `transcription.html` che l’utente visualizzerà nel browser.<sup>22</sup>

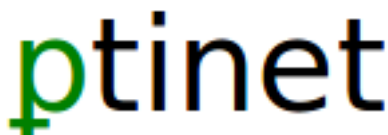


Fig. 3. Visualizzazione HTML del grafema

Il livello grafematico dell’edizione è visualizzato in nero, ma le abbreviazioni solo colorate dallo script JavaScript in colori diversi a seconda del tipo: il verde, ad esempio, rappresenta i brevisgrafi (vedi la sezione *Section 12*), e il rosso le abbreviazioni costituite dalla combinazione di un grafema-base e di un segno abbreviativo tracciato sopra di esso (come in “*ēfer*”).<sup>23</sup>

<sup>20</sup> Si tratta, come si è detto sopra, di tutto il contenuto testuale di `<w>` tranne il contenuto di `<expan>`. Sull’uso di quest’ultimo tag tornerò sotto, nel paragrafo *Section 5*.

<sup>21</sup> Anche il file `jsparser.js` è consultabile nel repository del progetto, all’URL <https://github.com/paolomonella/ursus/blob/master/jsparser.js>. Per una discussione delle scelte software all’interno del progetto, vedi sotto il paragrafo *Section 8*.

<sup>22</sup> Il sorgente del file HTML è consultabile nella repository GitHub, all’indirizzo <https://github.com/paolomonella/ursus/blob/master/transcription.html>, ed è visualizzabile da browser all’URL <http://www1.unipa.it/paolo.monella/ursus/transcription.html>.

<sup>23</sup> Tutte le convenzioni di visualizzazione sono spiegate per il lettore da una legenda in cima alla pagina `transcription.html`.



#### 4.5 Legature

Le legature, come quella tra i grafemi “t” e “i”, o tra “e” e “t” nella congiunzione “et”, non sono qui considerate grafemi, in quanto non hanno valore distintivo: non si trova nel manoscritto neanche una coppia di parole distinte dal fatto di avere o meno una determinata legatura (piuttosto che gli stessi grafemi scritti senza legatura).

Le legature dunque vengono qui considerate un caso particolare di allografi di posizione, e gli allografi non vengono codificati: si tratterebbe infatti di un altro livello di edizione.

Tuttavia, come ogni decisione specifica su cosa abbia valore distintivo – e sia dunque un grafema – e cosa no, anche la codifica delle legature è soggettiva e dipende dalle scelte dell’editore.

#### 4.6 Razionale della GToS

Tramite la GToS l’editore fornisce una sua descrizione del sistema grafematico del manoscritto. Essa rappresenta un’altra innovazione metodologica, realizzazione ed evoluzione dell’idea di “tabella dei segni” teorizzata da Orlandi 2010 e giustificata dal noto principio saussuriano per cui un segno si definisce solo all’interno di un sistema semiotico, in contrasto con ogni altro segno di quel sistema.<sup>24</sup>

Se un manoscritto M non ha una distinzione tra “u” e “v” (dunque ha un solo glifo per entrambi, “a forma di u”) ed un manoscritto N *ha* tale distinzione (dunque ha due glifi separati), il glifo “u” in M non è lo stesso grafema del glifo “u” in N, anche se “corrisponde” ad esso, né il glifo “v” in M è del tutto un altro grafema, in quanto anch’esso gli “corrisponde”.

Le pratiche attuali, ed anche le linee guida TEI XML, si accontentano di usare lo stesso carattere Unicode “u” (U0075+) nella trascrizione grafematica del glifo “a forma di u” di M e di quello di N.<sup>25</sup> Ciò è sufficiente per la visualizzazione a schermo e per l’intuito del lettore, ma per ogni ulteriore elaborazione dei dati da parte del computer ciò implica fallacemente che “u” di M e “u” di N siano lo stesso grafema, e che “u” di M e “v” di N siano grafemi diversi. La GToS serve dunque a “de-finire” ogni grafema all’interno del sistema grafico di uno specifico manoscritto.

Appare però evidente che la collazione tra manoscritti diversi debba avvenire ad un livello di trascrizione più alto, cioè al livello alfabetico (se si vogliono confrontare le ortografie) o al livello linguistico (se interessano le sole varianti “significative”).

### 5 Livello alfabetico

#### 5.1 Unità minime

Le sue unità minime sono lettere alfabetiche, qui chiamate “alfabemi” e distinte dai grafemi: i grafemi “significano” alfabemi.

---

<sup>24</sup> Vd. [13].

<sup>25</sup> Si vedano la sezione 5 *Characters, Glyphs, and Writing Modes* delle *Guidelines* TEI versione P5 in <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/WD.html> e [6].

## 5.2 Razionale del livello alfabetico

La standardizzazione dei sistemi scrittori introdotta dalla stampa gutenberghiana ci ha abituato a considerare grafemi ed alfabemi come un'unica cosa: nella stampa moderna, infatti, ad un grafema corrisponde una sola lettera alfabetica. Non era così nella scrittura manoscritta prima di Gutenberg, in cui un grafema, ad esempio un brevigrafo (“p”), poteva significare tre alfabemi (“per”); o due grafemi, ad esempio un grafema alfabetico ed un segno abbreviativo (“c̄”) potevano significare, combinati, tre alfabemi (“con”).

Per distinguere concettualmente grafemi ed alfabemi, basta ricordare che un alfabema, ad esempio la “a” dell’alfabeto latino, può essere rappresentato – “significato” – oltre che dal grafema “a”, anche dalla sequenza breve-lunga (nel codice Morse), dal fonema /a/ (nel parlato italiano – ma non nel parlato di altre lingue), o dalla parola “alfa” (nell’alfabeto fonetico radiotelegrafico, o NATO). Esso è però diverso dall’alfabema “α”, che appartiene all’alfabeto greco.

L’intero livello alfabetico di trascrizione è un mio ampliamento del modello proposto da Orlandi.<sup>26</sup> Come già detto, la scelta dei livelli a cui codificare un testo è soggettiva e dipende dagli interessi di ricerca dell’editore. In ambito digitale, però, ricade sull’editore la responsabilità di esplicitare e documentare il modello di edizione scelto.

## 5.3 Gli alfabemi nella GToS

La GToS riporta, per ogni grafema del manoscritto, il suo significato alfabetico standard nella colonna “Alphabeme(s)”.<sup>27</sup> Ad esempio, facendo riferimento ai grafemi del frammento della GToS riportato sopra (paragrafo *Section 4.2*):

- Il significato standard del grafema codificato col carattere Unicode “e” è l’alfabema codificato (e dunque identificato) col carattere Unicode “e”.
- Il significato standard del grafema-brevigrafo codificato con “p” (il cui glifo appare simile a una “p”) è una sequenza di alfabemi codificati rispettivamente con “p”, “e”, “r”.
- Il significato standard del grafema codificato con “æ” è una sequenza di alfabemi codificati coi caratteri Unicode “a” ed “e”. Si tratta dell’*e caudatum*, il cui glifo nel manoscritto è una “e” con una coda in basso a sinistra.<sup>28</sup>

<sup>26</sup> Orlandi invece, quando distingue grafemi e lettere alfabetiche, intende queste ultime sostanzialmente come fonemi – un livello, quello fonemico, da cui io, studiando testi latini medievali, mi tengo ben lontano. Scrive [21], paragrafo *Schema operativo: via in su e via in giù*: “A sua volta, si ricava da diverse unità grafematiche con un medesimo «significato» una ulteriore costante astratta, le unità alfabetiche, che si riconoscono dalla coincidenza di diverse grafie, come «ch» e «k», «x» e «sc(i)», «s» e «z», etc. Si deve peraltro tener presente che alcune unità grafematiche non hanno una corrispondenza nelle unità alfabetiche: esse sono i cosiddetti segni di punteggiatura, o segni diacritici, accenti, etc., il cui significato si aggiunge in modo diverso in casi diversi, non tanto a singole unità grafematiche, quanto a sequenze di unità grafematiche (p.e. l’accento in certi casi riguarda l’insieme della parola, e non la singola lettera)”.

<sup>27</sup> Solo i grafemi di tipo “Alphabetic”, “Brevigraph” e “Logograph” hanno un valore alfabetico, cioè significano lettere alfabetiche, non quelli di tipo “Abbreviation mark” o “Punctuation”: si vedano le considerazioni di Orlandi nella nota precedente.

<sup>28</sup> Questo grafema difficilmente si lascerebbe inquadrare come un’abbreviazione.

#### 5.4 Codifica TEI XML

La trascrizione al livello alfabetico non è codificata esplicitamente dall'editore per tutto il testo. Si danno infatti due casi:

1. Se in un certo punto del testo un determinato grafema ha il suo significato alfabetico standard, indicato nella colonna "Alphabeme(s)" della "Graphemic Table of Signs" (GToS), lo script JavaScript `jsparser.js` desume l'alfabema (o gli alfabemi) corrispondenti sulla base della GToS (file `GtoS.csv`: ad esempio, al grafema "æ" corrisponderanno gli alfabemi "a" ed "e");
2. Può però succedere – e nei manoscritti medievali succede sistematicamente – che uno o più grafemi non abbiano un significato alfabetico standard desumibile dalla GToS: ad esempio, se uno scriba ha abbreviato "confer" in "c̄fer", i due grafemi che costituiscono l'abbreviazione iniziale ("c" e il macron) hanno come significato alfabetico la sequenza di alfabemi "c", "o" e "n", che non è desumibile da una corrispondenza uno-ad-uno grafema/alfabema. Lo stesso vale per il grafema-brevigrafo iniziale di "q̄a" per "quia": qui un solo grafema ("q̄") rappresenta tre alfabemi ("q", "u" e "i").<sup>29</sup> In questo secondo caso l'editore, in fase di codifica, marca l'abbreviazione tramite un elemento `<choice>` con all'interno un `<abbr>` che contiene una sequenza di grafemi, e un `<expan>` che contiene una sequenza di alfabemi (il significato alfabetico di quei grafemi). Nel caso in esame, dunque, il software (`jsparser.js`) trova i due livelli (grafematico e alfabetico) già codificati esplicitamente come segue:<sup>30</sup>

```
<w ana="4-0-----" lemma="quia" n="quia" xml:id="w129">
  <choice>
    <abbr type="brevigraph">ϥ</abbr>
    <expan>qui</expan>
  </choice> a
</w>
```

In sintesi, la codifica a livello alfabetico è fornita esplicitamente dall'editore all'interno di un elemento `<expan>` solo quando (caso *b*) essa non sia desumibile dal software (`jsparser.js`) a partire dalla codifica grafematica tramite la tabella di corrispondenze fornita nella GToS, colonna "Alphabeme(s)".<sup>31</sup>

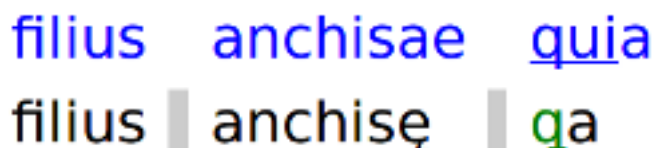
<sup>29</sup> La questione dello scioglimento delle abbreviazioni è, dal punto di vista semiotico, tanto complessa da mettere in crisi anche la lucidissima riflessione di [11] (vd. p. 302): "When a transcription expands an abbreviation, the exemplar and the transcript would seem to have different type sequences".

<sup>30</sup> Il brevigrafo iniziale è codificato con "ϥ" nella GToS e nel sorgente TEI XML per facilità di digitazione, e poi visualizzato con "q̄" in HTML dallo script `jsparser.js`.

<sup>31</sup> Essendo l'intero livello alfabetico una mia innovazione al modello orlandiano di edizione, è chiaro che anche l'inserimento della corrispondenza standard grafemi/alfabemi nella tabella dei segni non appartiene al modello di Orlandi. Anzi, quando gliene ho parlato l'ho trovato in disaccordo su tale inserimento. Il suo senso però è di rendere la tabella dei segni (GToS) parte integrante del sistema-edizione, nel senso che il software JavaScript la usa come input e ne ha bisogno per generare una rappresentazione completa del testo. La mia idea originaria era stata quella di inserire la GToS nel `<teiHeader>` dei file sorgente TEI XML (vd. [13] e

## 5.5 Visualizzazione HTML

Come risultato, lo script `jsparser.js` visualizza la trascrizione alfabetica in una riga di colore blu, parallela alla riga nera della trascrizione grafematica, direttamente nel file `transcription.html`:



filius anchisae quia  
filius anchisae qa

Fig. 4. Visualizzazione HTML della trascrizione alfabetica

La sequenza di alfabetici “qui” viene sottolineata dallo script JavaScript per evidenziarne l’allineamento con il grafema-brevigrafo “*q*”, a sua volta evidenziato in verde.

## 6 Livello linguistico

### 6.1 Unità minime

Le sue unità minime sono parole flesse, ma intese sinteticamente, non come sequenza di lettere alfabetiche. La parola flessa (es.: “*praepositionibus*”) viene così identificata nell’edizione tramite la combinazione di un lemma (“*praepositio*, -onis”) e di informazioni morfologiche (“*ablativo plurale*”).

La parola flessa viene così codificata univocamente al di là del sistema grafico e persino dell’alfabeto utilizzati nei singoli testimoni, offrendo così un livello di codifica utile per la visualizzazione “normalizzata” (la tradizionale edizione “critica” o interpretativa), per la ricerca e l’analisi testuale, per l’interoperabilità dell’edizione al di là delle specificità paleografiche del manoscritto e, potenzialmente, per la generazione di apparati critici dinamici in tradizioni pluritestimoniali.

### 6.2 Codifica TEI XML

Nel codice TEI XML, si sono usati a questo fine gli attributi dell’elemento `<w>` `@lemma` e `@ana` (quest’ultimo per l’analisi morfologica). Nell’esempio di “*praepositionibus*”, dunque:

```
<w ana="11C---02---" lemma="praepositio" n="praepositionibus" xml:id="w20673">
```

[15], paragrafo 10.4.4), ma questo si è rivelato successivamente un vicolo cieco a causa di una caratteristica di fondo di XML (non della TEI): non esistono in XML meccanismi formali per *ridefinire* i caratteri Unicode utilizzati per il contenuto testuale. Tanto valeva, dunque, utilizzare un file esterno nel formato più semplice e dunque più interoperabile, il CSV: non è apparso infatti necessario, data la quantità limitata di dati, creare un database.

Dove la stringa 11C---02---, significa “nominale, grado positivo, III declinazione, ablativo plurale femminile”.

L’attributo @n rappresenta un’ulteriore, più semplice, modalità per rappresentare una parola al livello linguistico: esso contiene un’ortografia “normalizzata” per la parola.

### 6.3 Visualizzazione HTML

Visualizzare il livello linguistico dell’edizione in un’ulteriore riga parallela, secondo il modello “a pentagramma” dell’edizione di Orso, pone un problema: se il browser dovesse visualizzare l’intera stringa “[praepositio] Nominal, Positive, III decl, Plural Ablative, Feminine” in una riga parallela in corrispondenza della trascrizione alfabetica “prepositionibus”, ogni riga di testo non potrebbe contenere più di un paio di parole.

In generale, i problemi di visualizzazione sono un buon segno: vogliono dire che il modello di edizione è veramente nuovo e che c’è più informazione da visualizzare di quanta un’edizione cartacea o un suo surrogato digitale potrebbero contenerne.

Una soluzione, però, andava trovata. Quella che ho adottato finora è di far visualizzare al browser soltanto il livello grafematico (riga nera in basso) e quello alfabetico (riga blu in alto): solo quando l’utente va col mouse sopra una parola (“hover”), il browser fa apparire un box nero “pop up” con lemma e morfologia della parola (il livello linguistico):

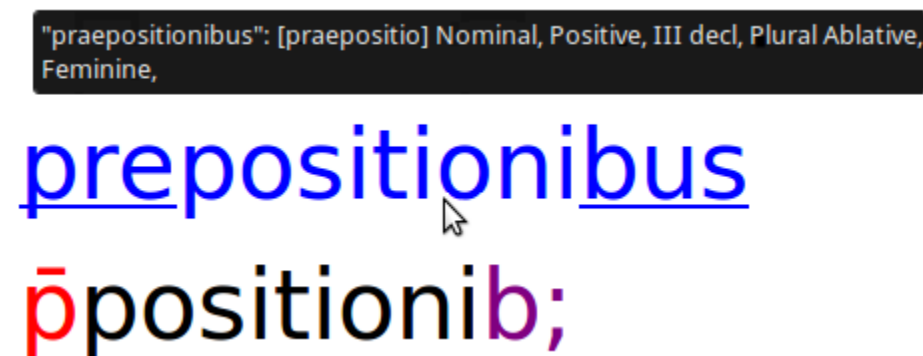


Fig. 5. Visualizzazione HTML del lemma e della morfologia della parola

### 6.4 Fasi di lavoro

Attraverso la distinzione, in fase di codifica, del livello linguistico dai livelli grafematico ed alfabetico, la linguistica computazionale può tornare ad essere, come è sempre stata la linguistica in ambito tradizionale, supporto fondamentale dell’attività ecdotica.

Nella realizzazione pratica dell’edizione ho proceduto come segue:

1. In fase di trascrizione del manoscritto, ho inserito in ogni elemento `<w>` un semplice attributo `@n` avente come valore una trascrizione “normalizzata” di ogni parola (ad esempio `<wn="praepositionibus">`): ho prodotto così il file `casanatensis.xml`.<sup>32</sup>
2. Successivamente, sul file TEI XML `casanatensis.xml` codificato in questo modo, ho fatto girare tramite uno script Python<sup>33</sup> il lemmatizzatore/PoS tagger *TreeTagger*<sup>34</sup> col parameter file *Latin-ITTB UD treebank*,<sup>35</sup> basato sull'IT-TB Index Thomisticus Treebank.<sup>36</sup> Viene così prodotto un file `lemmatized_casanatensis.xml`, che è quello che lo script JavaScript `jsparser.js` prende come input per generare la visualizzazione HTML dell'edizione (`transcription.html`).<sup>37</sup>
3. La fase finale prevedeva che io rivedessi l'output di *TreeTagger* (lemmatizzazione e analisi morfologica, col conseguente inserimento degli attributi `@lemma` e `@ana`) e correggessi gli errori, che ad una prima analisi sembrano riguardare un 20% delle parole.<sup>38</sup> Ma ad oggi (febbraio 2018) non ho potuto effettuare questa revisione – che non sarebbe breve.

## 7 Distinzione formale dei livelli nel markup

I livelli di trascrizione sono dunque distinti formalmente, ma non, come nel progetto Menota, tramite una codifica parallela, ripetuta, dello stesso testo sui tre livelli, detti `fac(simile)`, `dipl(omatic)` e `norm(alized)`:<sup>39</sup>

```
<w>
<choice>
  <me:fac>&drot;<am>&osup;</am>ttn<am>&bar;</am></me:fac>
```

<sup>32</sup> Per velocizzare la scrittura del codice sorgente TEI XML, in particolare l'inserimento del tag `<w>` col citato attributo `@n` per ogni parola e la codifica delle abbreviazioni, ho utilizzato delle convenzioni di marcatura, una sorta di “markdown” *ad hoc*. Uno script Python `ursusMarkdown.py` (<https://github.com/paolomonella/ursus/blob/master/ursusMarkdown.py>) trasformava questi codici semplificati e veloci da digitare nel vero e proprio markup TEI XML. Ad esempio, se io avevo digitato “ordin,e,2,em,” (si notino le virgole), lo script `ursusMarkdown.py` sostituiva questa stringa col codice `<w n="ordinem" xml:id="w66">ordin<choice><abbr type="superscription">e<am>2</am></abbr><expan>em</expan></choice></w>`. Tra le funzioni dello script, come si vede, c'era anche quello di inserire `xml:id` univoci.

<sup>33</sup> Si tratta dello script `paoloUrsusTagger.py`, basato sul modulo Python `ElementTree` per il parsing del file TEI XML `casanatensis.xml`: vd. [https://github.com/paolomonella/ursus/blob/master/lemma/my\\_tree\\_tagger\\_scripts/paoloUrsusTagger.py](https://github.com/paolomonella/ursus/blob/master/lemma/my_tree_tagger_scripts/paoloUrsusTagger.py)

<sup>34</sup> <http://www.cis.uni-muenchen.de/~char~schmid/tools/TreeTagger/>

<sup>35</sup> Scaricato il 15/01/2016 dal sito Universal Dependencies (<http://universaldependencies.org/>).

<sup>36</sup> Dal sito <http://itreebank.marginalia.it/>.

<sup>37</sup> Tutti i file citati sono visibili nel citato repository GitHub <https://github.com/paolomonella/ursus>.

<sup>38</sup> Vedi sotto il paragrafo *Section 10.2* per dati più precisi al riguardo sulla precisione dell'output di *TreeTagger*.

<sup>39</sup> L'esempio di codice sorgente è tratto dal paragrafo 3.4 *Multi-level transcriptions* della documentazione del *Menota Handbook 2.0*, [http://www.menota.org/HB2\\_ch3.xml#N1055B](http://www.menota.org/HB2_ch3.xml#N1055B).

```
<me:dipl>d<ex>ro</ex>ttin<ex>n</ex></me:dipl>
<me:norm>Dróttinn</me:norm>
</choice>
</w>
```

Nell'edizione di Orso, invece, la distinzione è determinata da una risemantizzazione del markup TEI XML e dall'integrazione, nel sistema-edizione, dei file sorgente TEI XML (*casanatensis.xml* e *lemmatized\_casanatensis.xml*) con la tabella dei segni GToS (*GtoS.csv*) e con il software JavaScript (*jsparser.js*), come segue:

1. Tutto il contenuto testuale<sup>40</sup> degli elementi `<w>` e dei suoi discendenti tranne `<expan>` costituisce la rappresentazione del livello *grafematico*.
2. Il contenuto testuale degli elementi `<expan>` appartiene invece al livello *alfabetico*. Buona parte della rappresentazione di tale livello, però, viene generata automaticamente dallo script *jsparser.js* a partire dalla GToS (*GtoS.csv*) e visualizzata direttamente sul file *transcription.html*.
3. Il livello *linguistico* viene codificato tramite gli attributi `@lemma` e `@ana` degli elementi `<w>`.

Si ottiene in questo modo una codifica più compatta dei file sorgente TEI XML – non dovendosi trascrivere tre volte la stessa porzione di testo – ed una rappresentazione più precisa del livello linguistico:<sup>41</sup>

## 8 Software

Come si è visto, ho scelto JavaScript come linguaggio principale per il processamento dei file sorgente (TEI XML e CSV), e Python per gli script che hanno facilitato la stesura del codice TEI e per l'integrazione con *TreeTagger*.

Ho preferito JavaScript e Python a XSLT, oltre che per la mia personale familiarità con questi linguaggi, per la loro maggiore flessibilità (processamento delle tabelle CSV, integrazione con *TreeTagger*). A causa della specificità del mio modello di edizione, utilizzare EVT per la visualizzazione avrebbe richiesto eccessive modifiche a tale piattaforma. L'estensione non eccessiva del testo, poi, non ha reso necessario gestire i dati tramite un database come eXist DB. Infine, ho scelto *TreeTagger* tra i vari lemmatizzatori/PoS tagger disponibili per una sua caratteristica, quella di prendere in considerazione anche il contesto della parola che analizza.

<sup>40</sup> Nel senso tecnico di “textual content” in XML.

<sup>41</sup> Devo l'idea di compattare i tre livelli di trascrizione in un unico sorgente TEI XML ad una conversazione con Desmond Schmidt al workshop DiXiT *The scholarly digital edition and the Humanities. Theoretical approaches and alternative tools* (Roma, 4 dicembre 2014). In quell'occasione Schmidt obiettava che se il livello alfabetico era desumibile dal livello grafematico tramite software, i due livelli erano equivalenti dal punto di vista dell'informazione contenuta e non valeva la pena di distinguerli nel markup (né concettualmente). Ho replicato che in alcuni casi (ad esempio nelle abbreviazioni) la trascrizione alfabetica non era desumibile da quella grafematica. Ne ho concluso che era più economico, ma ugualmente rigoroso dal punto di vista formale, codificare il livello alfabetico solo quando questo non era desumibile automaticamente da quello grafematico.

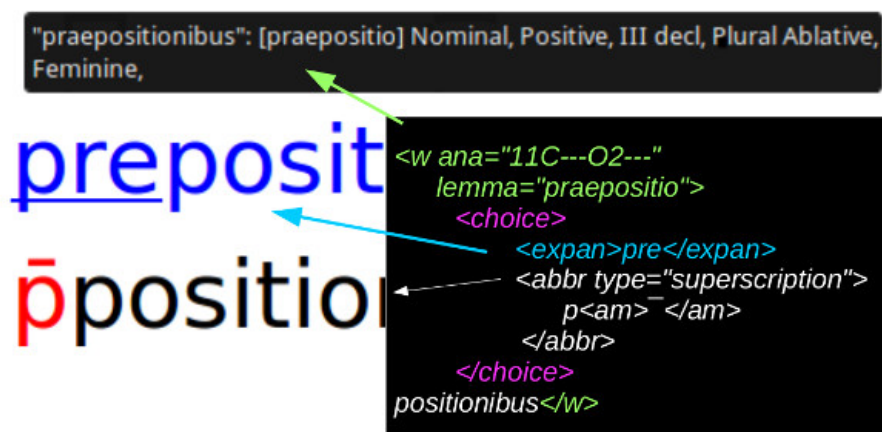


Fig. 6. Codifica XML/TEI del livello linguistico

Lo script JavaScript (`jsparser.js`) ad ogni visualizzazione processa dinamicamente, nel browser, sorgente TEI XML e GToS e genera la visualizzazione modificando direttamente il DOM del file `transcription.html`. Tale scelta, però, si è rivelata problematica, in quanto lo script impiega non meno di otto-dieci secondi per visualizzare la pagina, peggiorando l'esperienza del fruitore dell'edizione: uno script Python che elaborasse i sorgenti XML e CSV e generasse una visualizzazione statica in HTML sarebbe risultato più adatto agli scopi dell'edizione.

## 9 Discussione

### 9.1 Innovazioni

Come già detto, le due caratteristiche principali del modello di edizione proposto da Tito Orlandi sono i sotto-sistemi testuali ("livelli", nella mia terminologia) e la tabella dei segni.

Nel panorama della filologia digitale non mancano però progetti di ottima qualità che codificano il testo a più livelli. Tra le sperimentazioni più significative in questo campo si possono citare il *Wittgenstein's Nachlass. The Bergen Electronic Edition* su CD-ROM (BEE),<sup>42</sup> il *Vercelli Book Digitale*,<sup>43</sup> l'archivio testuale *Menota - Medieval Nordic Text Archive*,<sup>44</sup> la *Digitale Faust-Edition*.<sup>45</sup>

Rispetto a questi lavori, l'innovazione sostanziale del modello orlandiano consiste nel definire i livelli di rappresentazione del testo *non pragmaticamente ma semioticamente*.

<sup>42</sup> Vd. [11] e [http://wab.uib.no/wab\\_BEE.page](http://wab.uib.no/wab_BEE.page) con ulteriore bibliografia.

<sup>43</sup> Disponibile in <http://vbd.humnet.unipi.it/beta/>.

<sup>44</sup> Si vedano <http://menota.org/> e [9].

<sup>45</sup> Vd. <http://beta.faustedition.net/>; [3]; [4].



Nei progetti citati si distinguono normalmente un livello di trascrizione diplomatico, più vicino alle specificità grafiche della fonte primaria e definito in relazione ad alcuni fenomeni testuali specifici (abbreviazioni, allografi etc.), e un livello interpretativo, definito da un'ortografia normalizzata del testo secondo le convenzioni ortografiche e tipografiche attuali.<sup>46</sup>

Nel modello di Orlandi e nella mia edizione di Orso Beneventano, invece, i livelli sono definiti dalla teoria del testo e della sua rappresentazione digitale riassunta per sommi capi sopra (paragrafo *Section 2*), e sulla base di una modellizzazione dei concetti di glifo, grafema, lettera alfabetica, parola, fondata sulla semiotica.

In altre parole, se normalmente sono le pratiche e le convenzioni della trascrizione e della visualizzazione a definire la distinzione dei livelli, nel modello qui proposto è la riflessione semiotica ad orientare la pratica ecdotica.

Rispetto al modello di Orlandi, poi, l'edizione di Orso ha integrato quattro aspetti specifici:

- Il livello alfabetico dell'edizione;
- La corrispondenza tra grafema e il suo significato alfabetico standard nella tabella dei segni, ovvero la colonna “Alphabeme(s)” nella GToS;
- Il fatto che la tabella dei segni non sia solo parte della documentazione, rivolta alla lettura da parte del fruitore umano, ma una componente integrante dell'edizione stessa, processata dal software per generare il livello alfabetico delle porzioni di testo non coinvolte in abbreviazioni;
- La rappresentazione della parola flessa al livello linguistico tramite la combinazione di lemma e morfologia.

## 9.2 Quale testo leggere? L'utenza dell'edizione

Quale dei tre livelli (grafematica, alfabetica, linguistica) è dunque il “testo da leggere”? Quale corrisponde all'edizione critica tradizionale?

Uno dei vantaggi dell'edizione critica digitale è, in realtà, proprio quello di integrare in un sistema unico l'edizione cosiddetta “diplomatica” e quella “critica” o interpretativa, superando tale dicotomia.<sup>47</sup>

Nel modello di edizione qui proposto, l'edizione “diplomatica” è facilmente identificabile con il livello grafematico.

Una codifica formale sistematica di fenomeni paleografici come le abbreviazioni può fornire i dati necessari per costruire analisi quantitative e statistiche sui sistemi abbreviativi manoscritti, come ha evidenziato Neven Jovanović, per addestrare software in grado poi di tentare uno scioglimento automatico delle abbreviazioni, come ha suggerito Federico Boschetti.<sup>48</sup>

<sup>46</sup> Nell'archivio Menota esiste anche un livello intermedio: i tre livelli sono chiamati “facsimile”, “diplomatic” e “normalized”.

<sup>47</sup> Tra le molte riflessioni teoriche sulla questione, si vedano [2], [20], [23] e [24].

<sup>48</sup> Sono grato ai due studiosi per gli spunti preziosi, che giungono entrambi da domande che mi hanno posto alla fine di miei interventi a convegni AIUCD: Neven Jovanović, al convegno AIUCD 2016 di Venezia in cui ho presentato il contributo da cui nasce il presente articolo; Federico Boschetti, al convegno AIUCD 2018 di Bari.

Invece le funzioni svolte dall'edizione "critica" o "interpretativa" cartacea vengono distribuite a più livelli. Nel caso di una tradizione testuale unitestimoniale come quella del *De nomine* di Orso Beneventano, la funzione-lettura viene diffratta tra il livello alfabetico e quello linguistico.

Il lettore leggerà il livello alfabetico se vuole ignorare le abbreviazioni e la punteggiatura antica (livello grafemico) ma è interessato all'ortografia originale del testo. Al livello alfabetico non troverà infatti alcuna "normalizzazione" delle monottongazioni o dell'ortografia antica, né correzione dei *lapsus calami*.

Notis Toufexis [31] ha sostenuto che le ortografie "devianti" dei manoscritti medievali, per lo più non conservate nelle edizioni critiche a stampa di testi classici, se fossero preservate da una nuova generazione di edizioni critiche digitali costituirebbero una documentazione preziosa per gli studiosi dello sviluppo storico della lingua greca dal punto di vista fonetico e morfologico.

Il lettore leggerà invece il livello linguistico se è interessato al "testo" astratto: vi troverà l'ortografia e la punteggiatura "normalizzate", moderne, e i *lapsus calami* corretti. A questo livello, infatti, viene rappresentata *la parola latina* che, secondo l'editore, lo scriba *intendeva scrivere* (ad es.: l'ablativo plurale di "praepositio, -onis").

Se poi, a parere dell'editore, lo scriba intendeva effettivamente scrivere una determinata parola flessa (ad es. l'accusativo di un sostantivo), ma l'editore vuole *emendare* il testo trådito (ad es. in un dativo) – come ho fatto più volte col testo di Orso – questo tipo di intervento, pur appartenendo anch'esso concettualmente al livello linguistico, viene indicato tramite markup specifico: nel caso dell'edizione di Orso, ho usato gli elementi `<choice>` / `<sic>` / `<corr>`. Per questa via, di fatto, si aggiunge un altro livello di codifica: il testo critico dell'editore.

### 9.3 Sviluppi futuri: la collazione

Un ulteriore livello di complessità potrà essere aggiunto, in futuri esperimenti su questo modello di edizione, nelle edizioni di testi con più testimoni. In questi casi, ogni manoscritto avrà una sua edizione a più livelli, e la collazione in vista della definizione di un "testo dell'editore" confronterà le parole flesse

- al *livello linguistico*, se si vogliono evidenziare solo le varianti dette "significative",
- o al *livello alfabetico*, se si vogliono evidenziare le varianti ortografiche, ignorando invece le varianti paleografiche (abbreviazioni etc.), appartenenti al livello grafemico.

## 10 Un bilancio

### 10.1 Esperimento concluso

Ad oggi (febbraio 2018), l'edizione di Orso si può considerare conclusa, se non altro perché, perfezionamenti futuri a parte, è tempo di muovere verso altri progetti, ed è già in preparazione un'edizione critica digitale del *Chronicon* di Romualdo Salernitano (XII secolo).<sup>49</sup>

---

<sup>49</sup> Per una introduzione all'opera, edita integralmente da [8], si vedano [12] e [32].

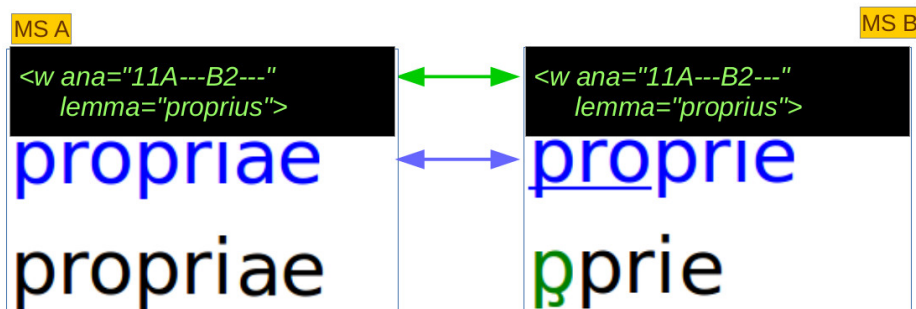


Fig. 7. Esempio di collazione di più testimoni

L'esperimento consisteva nel verificare la realizzabilità del modello di edizione proposto da Tito Orlandi. Ha avuto successo?

## 10.2 Due problemi

Due aspetti sono risultati problematici:

- *Estensione dell'edizione.* L'opera di Orso Beneventano *Abbreuiatio artis grammaticae* occupava circa 63 carte del codice *Casanatensis* 1086. Tra il 2015 e il 2016, in un paio d'anni di lavoro, per dire così, part-time (cioè lavorando anche a scuola come docente), ho pubblicato 11 carte (ff. 1r-11r), cioè la sezione relativa al sostantivo (*De nomine*), per un totale di 14.354 parole. Inoltre ho lavorato su un *codex unicus*, non su più testimoni. Vari fattori hanno concorso a questa lentezza nel processo di produzione:
  1. Molto tempo è stato occupato dalla messa a punto teorica del modello di edizione e dalla scrittura del software che ho dovuto creare da zero – essendo peraltro l'unico a lavorare su questo progetto, e per di più un umanista e non un programmatore per formazione.
  2. Non sono partito da una trascrizione digitale preesistente (risultato dell'OCR di un'edizione a stampa) in quanto il testo è inedito.
  3. La scrittura è in molti punti sbiadita e quasi illeggibile – il che contribuisce forse a spiegare perché il testo è rimasto inedito fino ad oggi.<sup>50</sup>
- *Revisione del livello linguistico.* Come ho già scritto nel paragrafo *Section 5/ Section 6.4*, ad oggi non ho potuto effettuare la revisione e correzione della lemmatizzazione e dell'analisi morfologica effettuate da *TreeTagger* che hanno popolato gli attributi @lemma e @ana degli elementi <w>. Da un saggio effettuato su 397 pa-

<sup>50</sup> In un saggio fondamentale sul manoscritto, [17], 291 n. 1 scriveva: “Può naturalmente essermi sfuggito qualcosa; anche perché la scrittura è spesso così evanida da mettere a dura prova la vista più acuta”.

role,<sup>51</sup> la precisione dell'analisi si attesta intorno all'80% (82% per @lemma e 79% per e @ana), per cui vi sono circa venti parole lemmatizzate o analizzate male ogni cento: un'accuratezza inaccettabile per un'edizione critica. Certo, ogni parola ha comunque una sua rappresentazione al livello linguistico costituita dalla trascrizione normalizzata registrata nell'elemento @n, ma una delle sperimentazioni più interessanti dell'edizione di Orso consisteva appunto nell'uso di @lemma e @ana.

L'edizione è stata dunque realizzata per quelle 14.354 parole, per i livelli grafemico e alfabetico, ed anche per quanto riguarda il livello linguistico – almeno relativamente agli attributi @n e a quel 79-82% di attributi @lemma e @ana corretti. Non si può dunque dire che il modello di edizione a più livelli proposto da Orlandi (ulteriormente complicato dall'aggiunzione del livello alfabetico) non sia realizzabile.

Ma senz'altro si può dire che la sua realizzazione ha richiesto troppo tempo – e altro sarebbe necessario per ultimare la revisione del livello linguistico rappresentato tramite la combinazione di lemma e morfologia.

### 10.3 Possibili soluzioni

Sarà dunque utile individuare e analizzare i punti critici, cioè gli aspetti specifici del lavoro che hanno richiesto più tempo, per immaginare una versione più snella di questo modello di edizione – da applicare all'edizione del *Chronicon* di Romualdo Salernitano o ad altri esperimenti futuri:

- *Livello grafemico*. La codifica della punteggiatura antica e la marcatura con <choice> / <abbr> / <expan> di ogni singola abbreviazione richiedono molto tempo, anche se velocizzate tramite codici di codifica rapida personalizzati e macro di Vim. Questo problema però non è specifico del mio modello di edizione, ma comune a tutte le trascrizioni digitali a livello grafemico. Nell'edizione di Romualdo Salernitano intendo testare due soluzioni:
  1. *Sfruttare la sistematicità di molte abbreviazioni*. Se nella maggioranza dei casi l'abbreviazione “e2” significa “em” a livello alfabetico, si potrà codificare solo “e2” (il solo livello grafemico), senza il markup <choice> / <abbr> / <expan>, e lasciare che il software, ogni volta che incontra la stringa “e2”, generi il livello alfabetico (“em”) sulla base di una tabella di scioglimenti standard delle abbreviazioni. Solo nei casi specifici in cui i grafemi “e2” significassero una sequenza diversa di alfabemi (ad esempio “er”), userò il markup esplicito <choice> / <abbr> / <expan>.
  2. *Non codificare tutti i livelli per tutto il testo*. Una volta delineato il modello di edizione e la distinzione concettuale dei livelli, si potrà decidere, per una determinata sezione del testo o per un intero testimone, di codificare il testo al solo livello alfabetico – purché lo si dichiari formalmente nel sorgente TEI XML (tramite l'attributo @decls) e si adatti il software di conseguenza.

---

<sup>51</sup> Ho tratto le parole da due punti diversi del manoscritto: dall'inizio del rigo 6r.a.1 (foglio 6 *recto*, colonna a, rigo 1) “comparatiua species...” alla fine del rigo 6r.a.25 “...deterior”, e dall'inizio di 11r.a.9 alla fine di 11r.a.25.

- *Livello linguistico*. L’ambizione di rappresentare le parole flesse tramite una combinazione di lemma e morfologia, invece che con una sequenza di caratteri, si è rivelata non sostenibile per un progetto di edizione portato avanti da una sola persona in tempi limitati. L’uso del lemmatizzatore/PoS tagger *TreeTagger* impone una curva di apprendimento iniziale, ma soprattutto richiede ulteriore tempo per la revisione del risultato. Possibili soluzioni:
  1. *Ortografia normalizzata*. Si potrà sempre ripiegare sulla pratica comune di rappresentare le parole anche al livello linguistico (spesso detto “normalizzato”) tramite una loro trascrizione “standard”, cioè come le scriveremmo oggi: si potrebbe avere, desumendo un esempio dall’edizione di Orso, “spanus” (livello grafematico) / “spanus” (livello alfabetico), “hispanus” (livello linguistico, con ortografia normalizzata, trasparente agli strumenti di ricerca ed analisi testuale). Questa era peraltro una delle strategie adottate già nell’edizione di Orso, tramite l’uso dell’attributo @n di <w>. Resta comunque un aspetto problematico: quale ortografia, ad esempio, di “praepositio” si può considerare “standard” in un testo latino medievale? Scegliere la stringa “praepositio” (secondo l’ortografia del latino classico) faciliterebbe il lavoro del software per la lemmatizzazione, il PoS tagging, la cross-corpus search, la collazione automatica, ma sarebbe discutibile sotto il profilo storico-culturale.<sup>52</sup> Scegliere la grafia medievale “prepositio”, viceversa, sarebbe più giustificabile dal punto di vista umanistico, ma più problematico dal punto di vista informatico, per quegli stessi software. Il problema della rappresentazione del livello linguistico, a mio parere, resta aperto.
  2. *Rinunciare al livello linguistico*. Almeno nella prima fase del lavoro per l’edizione di Romualdo Salernitano, prevedo semplicemente di non codificare il livello alfabetico, ma solo quelli grafematico e alfabetico: quest’ultimo potrebbe bastare per molti usi, a partire dalla semplice lettura. Resteranno però i problemi relativi alla collazione automatica e alla ricerca testuale.
- *Lentezza dello script JavaScript*. Come ricordato sopra, lo script `jsparser.js` richiede molto tempo per processare i sorgenti dell’edizione e produrre dinamicamente la visualizzazione nel browser.
  1. *Python*. Il software necessario per gestire un modello di edizione così complesso non sarà mai troppo semplice: converrà dunque in futuro usare uno script Python che effettui il processamento una volta per tutte in locale e produca un file HTML statico pronto per la visualizzazione.

## 11 Sviluppo, licenze e riuso

L’edizione è *open source*: tutto il codice sorgente (markup e software) è stato scritto con l’editor Vim<sup>53</sup> ed è disponibile sul repository <https://github.com/paolomonella/>

<sup>52</sup> Soprattutto per i testi medievali, per cui la prassi ecdotica è spesso di mantenere l’ortografia dell’epoca, è discutibile normalizzare a livello linguistico “presum” in “praesum”, “spanus” in “hispanus” o “mouilis” in “mobilis”: si può infatti argomentare che esista un lemma del latino medievale “mouilis, -e”, diverso dal lemma del latino classico “mobilis” in quanto appartenente ad un sistema linguistico diverso.

<sup>53</sup> <http://www.vim.org/>.

ursus sotto la GNU General Public License, insieme ad una ampia documentazione in inglese.

La scelta dei linguaggi utilizzati, in particolare di TEI XML, ma anche di JavaScript, Python e del formato CSV, è mirata a consentire il riuso del codice sorgente – nella prospettiva delineata nel paragrafo *Section 3.1*. Allo stesso fine mira la documentazione dettagliata di ogni aspetto, metodologico e tecnologico, del sistema-edizione, e in particolare della semantica degli elementi TEI XML utilizzati.

L'impianto metodologico dell'edizione prevede che le riproduzioni digitali del *codex unicus* che contiene il testo (*Casanatensis* 1086) siano collegate alla trascrizione. Tuttavia, un accordo con la Biblioteca Casanatense di Roma, che ha fornito le riproduzioni, attualmente vincola l'editore a non rendere disponibili le immagini online.

## 12 Glossario

Su suggerimento di uno dei “blind reviewer” di questo articolo (i suggerimenti di entrambi i reviewer sono stati preziosi, e sono loro grato), riporto qui un glossario essenziale di alcuni concetti definiti nell'articolo. L'ordinamento dei termini non è alfabetico ma logico.

*Glifo*. Ogni segno grafico nel documento: non lo sono solo i tratti di penna non intenzionali o puramente decorativi.

*Grafema*. Glifo avente valore distintivo: nel sistema grafico esiste almeno una coppia minima di parole che differiscono per esso. Nell'edizione di Orso sono considerati grafemi anche i segni di punteggiatura (segni paragrafematici, o grafemi “non alfabetici”): in questo caso, il valore distintivo riguarda la funzione del segno come “descrizione metalinguistica della struttura del testo”.<sup>54</sup>

*Brevigrafo*. Grafema costituito da un glifo unico, che significa una sequenza di più lettere alfabetiche (“alfabemi”: vd. più sotto). In “c̄fer” (per “confer”), l'abbreviazione iniziale è costituita dalla combinazione di due grafemi distinti (“c” e macron “̄”). In “(U+A753)sum” (per “prosum”), invece, l'abbreviazione iniziale è un brevigrafo in quanto si tratta di un glifo, e quindi di un grafema, unico.

Graphemic Table of Signs (GToS). Tabella che riporta tutti i grafemi individuati dall'editore come aventi valore distintivo in un manoscritto.

*Allografo*. Modello formale, tipo, di glifo: se è impossibile che due token-glifo (due segni grafici) sulla carta siano identici nella loro effettiva realizzazione, ognuno di essi è però un'istanza specifica di un tipo-forma ideale, l'allografo – ad esempio la “s corta” o la “s lunga”. A differenza dei grafemi, gli allografi non hanno valore distintivo: non esistono infatti due parole distinte dal solo fatto di avere una “s corta” o una “s lunga” nella medesima posizione. Diremo dunque che nel sistema grafico di un manoscritto il grafema “s” viene rappresentato (viene “significato”) dai due allografi “s corta” e “s lunga”.

---

<sup>54</sup> [5], paragrafo 4: “[Il markup] può, come la punteggiatura, o come ogni altro segno diacritico, essere considerato come una descrizione metalinguistica della struttura del testo, o come un'estensione del sistema stesso di scrittura, che permette di esplicitare caratteri altrimenti impliciti del testo”.

*Allografo di posizione.* Allografo la cui forma è determinata dalla sua posizione relativamente agli altri grafemi. Ad esempio, nel codice *Casanatensis* 1086, un allografo del grafema “t” ha la forma bassa più comune nella scrittura beneventana, simile ad un tau ( $\tau$ ), un secondo ha la forma di una “s” molto stretta ed allungata verso l’alto, un terzo ha la forma di un epsilon ( $\epsilon$ ) appoggiato in legatura al grafema successivo. Il secondo allografo si trova solo in fine di parola, e il terzo solo in legatura con determinati glifi, ad esempio con un allografo specifico del grafema “e”.

*Alfabema (o lettera alfabetica).* Segno astratto, non grafico, appartenente ad un “alfabeto”.<sup>55</sup> Ad esempio, la “a” dell’alfabeto latino può essere rappresentata (“significata”) dal grafema “a latina” o dalla sequenza breve-lunga del codice Morse (vedi il paragrafo *Section 5.2* per ulteriori esempi). Un grafema può significare un alfabema (il grafema “a”), più alfabemi (il grafema *e caudatum* “ĕ” significa i due alfabemi “a” ed “e”, mentre il grafema “p” significa i tre alfabemi “p”, “e”, “r”), nessun alfabema (i segni di punteggiatura), o sequenze variabili di alfabemi se usato in combinazione con altri grafemi (un segno abbreviativo come il macron “̄” può combinarsi ad un grafema alfabetico-base come “a” per significare i due alfabemi “a” e “m” oppure “a” e “n”, o altri ancora).

## References

1. Baroni, Antonio. “La grafematica: teorie, problemi e applicazioni.” Tesi di laurea specialistica, università di Padova, 2009.
2. Bodard, Gabriel e Juan Garcés. “Open source critical editions: a rationale.” In *Text editing, print, and the digital world*, a cura di Marilyn Deegan e Kathryn Sutherland, 83–98. Londra: Ashgate Publishing, 2009. [http://www.stoa.org/wordpress/wp-content/uploads/2010/09/Bodard-Garces\\_2009\\_Open-source-digital-editions.pdf](http://www.stoa.org/wordpress/wp-content/uploads/2010/09/Bodard-Garces_2009_Open-source-digital-editions.pdf)
3. Bohnenkamp, Anne, Gerrit Brüning, Silke Henke, Katrin Henzel, Fotis Jannidis, Gregor Middel, Dietmar Pravida e Moritz Wissenbach. “3. Perspektiven auf Goethes ›Faust‹ werkstattbericht der historisch-kritischen hybridedition.” a cura di Anne Bohnenkamp. *Jahrbuch des freien deutschen hochstifts* 2011, 2012, 23–67.
4. Brüning, Gerrit, Katrin Henzel e Dietmar Pravida. “Multiple encoding in genetic editions: the case of ‘Faust.’” *Journal of the Text encoding initiative*, num. 4 (2013). <https://doi.org/10.4000/jtei.697>.
5. Buzzetti, Dino. “Ambiguità diacritica e markup: note sull’edizione critica digitale.” In *soluzioni informatiche e telematiche per la filologia*, a cura di Simone Albonico. Pavia: Università degli Studi di Pavia, 2000. [http://studiumanistici.unipv.it/diplslamm/pubtel/Atti2000/dino\\_buzzetti.htm](http://studiumanistici.unipv.it/diplslamm/pubtel/Atti2000/dino_buzzetti.htm).
6. Driscoll, Matthew James. “Levels of transcription.” In *Electronic textual editing*, a cura di Lou Burnard, Katherine O’Brien O’Keffee e John Unsworth. New York: Modern Language Association of America, 2006. [http://www.tei-c.org/About/Archive\\_new/ETE/Preview/driscoll.xml](http://www.tei-c.org/About/Archive_new/ETE/Preview/driscoll.xml).
7. Fioretti, Paolo. “L’eredità di un maestro. Genesi ed edizione della grammatica di Orso Beneventano.” In *Libri di scuola e pratiche didattiche. Dall’antichità al rinascimento, Atti del convegno internazionale organizzato dall’università degli studi di Cassino (Cassino 7-10 maggio*

---

<sup>55</sup> Sulla complessità e il carattere astratto del concetto di “alfabeto” trovo illuminanti le considerazioni di [16].

- 2008), a cura di Lucio Del Corso e Oronzo Pecere, 293–328. Cassino: Edizioni università di Cassino, 2010.
8. Garufi, Carlo Alberto, a cura di. *Romualdi salernitani cronicon (a.m. 130-A.C. 1178)*. Vol. 127. *Rerum italicarum scriptores*. Città di Castello: S. Lapi, 1914.
  9. Haugen, Odd Einar. “Parallel views: multi-level encoding of medieval nordic primary sources.” *Literary and linguistic computing* 19, num. 1 (2004): 73–91. <https://doi.org/10.1093/llc/19.1.73>
  10. Huitfeldt, Claus e Christopher Michael Sperberg-McQueen. “What is transcription?” *Literary and linguistic computing* 23, num. 3 (Settembre 1, 2008): 295–310. <https://doi.org/10.1093/llc/fqn013>
  11. Huitfeldt, Claus. “Philosophy case study.” In *Electronic textual editing*, a cura di Lou Burnard, Katherine O’Brien O’Keffe e John Unsworth. New York: Modern Language Association of America, 2006. [http://www.tei-c.org/About/Archive\\_new/EET/Preview/huitfeldt.xml](http://www.tei-c.org/About/Archive_new/EET/Preview/huitfeldt.xml)
  12. Matthew, Donald James “The chronicle of romuald of Salerno.” In *The writing of history in the middle ages: essays presented to Richard William Southern*, a cura di Ralph Henry Carless Davis, 239–74. Oxford: Oxford University Press, 1981.
  13. Monella, Paolo. “A saussurean approach to graphemes declaration in charDecl for manuscripts encoding.” presentazione alla conferenza TEI, Roma, 2013. <http://www1.unipa.it/paolo.monella/tei2013/index.html>
  14. Monella, Paolo. “Forme del testo digitale.” In *Filologia digitale: problemi e prospettive*, a cura di Raul Mordenti, 143–61. Roma: Bardi Edizioni, 2017.
  15. Monella, Paolo. “Many witnesses, many layers: the digital scholarly edition of the *Iudicium coci et pistoris* (Anth. Lat. 199 Riese).” In *Digital humanities: progetti italiani ed esperienze di convergenza multidisciplinare, atti del convegno annuale dell’Associazione per l’informatica umanistica e la cultura digitale (AIUCD), Firenze, 13-14 Dicembre 2012*, 173–206. Roma: Sapienza università editrice, 2014. <https://doi.org/10.13133/978-88-98533-27-5>
  16. Mordenti, Raul. *Paradosis. A proposito del testo informatico*. Vol. IX. *Memorie lincee Scienze morali, storiche, filologiche*. Roma: Accademia Nazionale dei Lincei, 2011.
  17. Morelli, Camillo. *I trattati di grammatica e retorica del cod. casanatense 1086. Nota del dott. C. Morelli, presentata dal socio G. Vitelli*. Vol. XIX. *Rendiconti della reale accademia dei Lincei, classe di scienze morali, storiche e filologiche*. Roma: Tipografia della accademia, 1910.
  18. Munzi, Luigi. “Prisciano nell’Italia meridionale: la *adbreuiatio artis grammaticae* di Orso Di Benevento.” In *Priscien*, a cura di Marc Baratin, Bernard Colombatand e Louis Holtz, 21: 463–79. *Studia Artistarum*. Brepols Online Publishers, 2009. <https://doi.org/10.1484/M.SA-EB.3.1228>
  19. Orlandi, Tito. “Edizione digitale sperimentale di Niccolò Machiavelli, *De Principatibus*.” 2006. <http://www.cmcl.it/~orlandi/principe/>
  20. Orlandi, Tito. *Informatica testuale. teoria e prassi*. Roma: Laterza, 2010.
  21. Orlandi, Tito. “Linguistica, sistemi, e modelli.” In *Il ruolo del modello nella scienza e nel sapere (Roma, 27-28 Ottobre 1998)*, 73–90. Roma: Accademia Nazionale dei Lincei, 1999. <http://www.cmcl.it/~orlandi/pubbli/modello.html>
  22. Orlandi, Tito. “Ripartiamo dai diasistemi.” In *I nuovi orizzonti della filologia, ecdotica, critica testuale, editoria scientifica e mezzi informatici elettronici, convegno internazionale 27-29 maggio 1998. Atti dei convegni lincei*, 87–101. Roma: Accademia nazionale dei Lincei, 1999. <http://www.cmcl.it/~orlandi/pubbli/branca.html>
  23. Pierazzo, Elena. “A rationale of digital documentary editions.” *Literary and linguistic computing* 26, num. 4 (2011): 463–77. <https://doi.org/10.1093/llc/fqr033>
  24. Pierazzo, Elena. *Digital scholarly editing: theories, models and methods*. Farnham, Surrey: Ashgate, 2015.



25. Robinson, Peter e Elizabeth Solopova. "Guidelines for the transcription of the manuscripts of the wife of Bath's Prologue." *Canterbury Tales*, 1993. <http://www.textualcommunities.usask.ca/web/canterbury-tales/wiki/-/wiki/Main/Original+Transcription+Guidelines>
26. Robinson, Peter. "Exactly what are we transcribing?" *Canterbury Tales* (blog), 2014. <http://www.textualcommunities.usask.ca/web/canterbury-tales/blog/-/blogs/exactly-what-are-we-transcribing->
27. Sampson, Geoffrey. *Writing systems: a linguistic introduction*. Stanford: Stanford University Press, 1990.
28. Segre, Cesare. "Les transcriptions en tant que diasystèmes." In *La pratique des ordinateurs dans la critique des textes*, a cura di Jean Irigoien e Gian Piero Zarri, 45–49. Parigi: Éditions du CNRS, 1979.
29. Stokes, Peter A. "Describing handwriting, part IV: recapitulation and formal model." *DigiPal* (blog), 2011. <http://www.digipal.eu/blog/describing-handwriting-part-iv-recapitulation-and-formal-model/>
30. Tjepmar, Jochen, Christoph Teichmann, Gerhard Heyer, Monica Berti e Gregory Crane. "A new implementation for canonical text services. In "Proceedings of the 8th workshop on language technology for cultural heritage, social sciences, and humanities (LaTeCH), 2014, 1–8. <https://doi.org/10.3115/v1/W14-0601>
31. Toufexis, Notis. "One era's nonsense, another's norm: diachronic study of greek and the computer." In *Digital research in the study of classical antiquity*, a cura di Simon Mahony e Gabriel Bodard, 105–18. Farnham, Surrey: Ashgate, 2010.
32. Zabbia, Marino, a cura di. "Romualdo Guarna arcivescovo di Salerno e la sua cronaca." In *Salerno nel XII secolo. Istituzioni, società, cultura*, a cura di Paolo Delogu e Paolo Peduto, 380–98. Salerno: provincia di Salerno - Centro studi salernitani «Raffaele Guariglia», 2004.