A sustainable work flow for a multi-layer edition of the *Chronicon* by Romualdus Salernitanus

Paolo Monella[1]
[1] Università di Palermo, Italy, paolo.monella@gmx.net

ABSTRACT (300 words)

My experimental scholarly digital edition of the *De nomine* by Ursus Beneventanus tested the feasibility of the edition model theorized by Orlandi 2010, based on three different layers of text representation (graphematic, alphabetic and linguistic) and on a table of all graphemes having distinctive value in the graphic system of a specific manuscript ("graphematic table of signs"). Its work flow, however, proved to be very time-consuming. This talk analyzes the specific practices of the Ursus edition that mostly slowed down the work flow and outlines possible solutions to be applied in my ongoing digital edition of the *Chronicon* by Romualdus Salernitanus. Those aspects include (a) the markup of abbreviations, which can be expedited by taking advantage of the systematic nature of ancient abbreviations, (b) the markup of ancient punctuation, which will be omitted altogether, and (c) the representation of the linguistic layer. The minimal units of the latter are inflected words regardless of any specific spelling. In the Ursus edition, each word was represented at this layer by a combination of lemma and morphological information (e.g.: ablative plural of lemma "praepositio, -onis"), by means of `@lemma` and `@ana` attributes of `<w>`. The values of those attributes were populated by the lemmatizer/PoS tagger *TreeTagger*, but needed to be reviewed manually. The Romualdus edition will initially include no representation of the Linguistic Layer. If time will suffice, a simplified representation of it will be added, falling back on the common practice of representing words at the linguistic layer by means of their "normalized" spelling. Lastly, while in the Ursus project the TEI-to-HTML transformation was performed dynamically by JavaScript, in the Romualdus edition a Python script will perform this task statically.

KEYWORDS

Digital scholarly edition, layers, Romualdus Salernitanus, Ursus Beneventanus, manuscript, TEI XML, lemmatization, PoS tagging, Python

1. GOAL OF THE TALK

This talk will first shortly analyze the main issue of my edition of the the *De nomine* by Ursus Beneventanus (http://www.unipa.it/paolo.monella/ursus/; see Monella 2006): the excessive time-intensiveness of the work flow. It will then provide strategies to expedite the production of the edition of the *Chronicon* by Romualdus Salernitanus (XII Century; see Garufi 1914, Matthew 1981, Zabbia 2004), which I have recently started to work on, in the framework the ALIM project (http://alim.unisi.it/).

2. METHODOLOGY

My experimental digital scholarly edition of the *De nomine* tested the feasibility of the edition model theorized by Orlandi 2010. The Romualdus edition will be based on the same methodological principles, including

1. the representation of the text on three different layers (graphematic, alphabetic and linguistic) to integrate "diplomatic" and "interpretive" edition (see Haugen 2004, Driscoll 2006, Huitfeldt 2006, Orlandi 2006 and 2010, Brüning *et al.* 2013, Monella 2014, Pierazzo 2015) and
2. a table of all graphemes having distinctive value in the graphic system of a specific manuscript ("graphematic table of signs").

Further details on the edition model are in Monella 2016 and in the project documentation on the edition website.

3. RESULTS EXPECTED

The Ursus project aimed to be a proof of concept, an experiment to test those theoretical and practical issues that would only arise from a real-world application of Orlandi's ideas. The Romualdus edition sets out the specific goal of finding a balance between methodological sophistication and work flow sustainability. In other words, it aims at implementing as many features of the Ursus edition model as possible within the project's three years time frame.

4. ISSUES OF THE URSUS EDITION

Two specific issues of the Ursus project were (a) the extent of the text actually published, compared with the time needed, and (b) the fact that I could not fully review the encoding of the linguistic layer.

4.1 Extent of the edition

In two years I produced an edition of 11 folios of manuscript *Casanatensis* 1086, the *codex unicus* of the grammatical works by Ursus Beneventanus, on three layers (graphemes, alphabetic letters and inflected words): not much. Possible causes of such a long elaboration time include the following:

- I was the only contributor on the project while also working full-time as a school teacher;
- This edition model had never been applied before (except for Orlandi 2006);
- I wrote all the software without any formal training as a professional programmer;
- There was no OCR-ed base text to start from, because the work was unpublished – though it consisted in a paraphrased summarization of known grammatical sources;
- The manuscript was often very much faded out.

4.2 Review of the linguistic layer

In addition, by the end of the research project I did not get to fully review the "linguistic layer" of the edition.

At this layer each word was represented by a combination of lemma and morphological information. For example, inflected word "prepositionibus" was represented as `<w ana="11C---O2---" lemma="praepositio" n="praepositionibus" xml:id="w20673">`, where `11C---O2---` means "Nominal, Positive, III decl, Plural Ablative, Feminine". The values of the `@lemma` and `@ana` attributes were generated by lemmatizer/PoS tagger *TreeTagger* (http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/) and were correct for about 93% of words. I had planned to review those values to correct misanalyzed words, but as of fall 2017 I could not complete this final stage of the work, which is not likely to be very short.

In any case, a simpler representation of each inflected word at the linguistic layer is in the `@n` attribute, providing its "normalized" transcription.

5. WHAT TOOK YOU SO LONG? AN ANALYSIS AND SOME POSSIBLE SOLUTIONS

I shall now analyze in detail the specific aspects of the Ursus edition that mostly slowed down the work flow, and discuss the solutions that I am planning to apply in the Romualdus edition:

5.1 Graphematic layer: punctuation

Issue. In the Ursus edition, I encoded the original grapheme of each punctuation sign and marked its value (strong or weak separation, highlighting of special words, pause in the intonation etc.) with an `@n` attribute. In the following example

```
<pc n="0">.</pc>
```

the "`0`" value of `@n` means that the punctuation sign does not mark any syntactical separation or pause in the sentence.

This was, like many aspects of the Ursus edition model, an experiment to explore and question our notions of graphic system, grapheme, immediate "meaning" of a grapheme. However, it was quite laborious.

Solution. Having explored the feasibility and the methodological issues of such a detailed markup, I will simply skip the transcription of such paragraphematic signs in the Romualdus edition. Those signs can be ascribed to a separate "sub-system", in Orlandi's terminology, within the whole graphic system of the manuscript. Choosing which objects to encode and the manner with which to encode them is subjective and always legitimate, as long as the editor declares and discusses his or her choices in the project documentation.

5.2 Graphematic layer: abbreviations

Issue. For Ursus, at the graphematic layer I encoded each of the graphemes composing an abbreviation (e.g. the brevigraph "ꝑ" for "per", or "p" plus macron for "pre") and marked each component of that abbreviation with the specific markup (`<choice>` / `<abbr>` / `<am>` / `<expan>`) as follows:

```
<w ana="11C---O2---" lemma="praepositionibus" n="praepositionibus" xml:id="w20673">
        <choice>
                <abbr type="superscription">p<am>¯</am></abbr>
                <expan>pre</expan>
        </choice>
        positioni
        <choice>
                <abbr type="after">b<am>;</am></abbr>
                <expan>bus</expan>
        </choice>
```

```
</w>
```

I created custom “markdown-like” conventions to speed up the typing. For example, to generate the above code I would simply type

```
,p,-,prae,positioni,b,;,bus,
```

where commas separate the elements of the abbreviation (base grapheme, abbreviation mark, alphabetic expansion). A Python script would parse the string above and generate the TEI XML code.

Nevertheless, the encoding of each abbreviation in the manuscript slowed down the transcription, especially since I was not able to start from an OCR text, but I was typing the transcription anew without the help of any previous print edition.

I should note that this issue is not specific to my edition model, but affects all projects transcribing the fairly ubiquitous abbreviations of ancient handwritten primary sources.

Solution. I am not encoding all paragraphs of the text at both the graphematic and the alphabetic layers. For some paragraphs, I am only providing an alphabetic transcription, which is not very difficult to create starting from the OCR of the Garufi 1914 edition.

The `@decls` attribute of `<p>` marks the paragraphs encoded at the ***a***lphabetic layer only: `<p decls="#a">`. The reference `#a` points to an `<editorialDecl>` element in the `<encodingDesc>` of the TEI header:

```
    <encodingDesc>
        <editorialDecl xml:id="ag" default="true">
            <p>Paragraph encoded at the Graphematic and Alphabetic Layers.</p>
        </editorialDecl>
        <editorialDecl xml:id="a">
            <p>Paragraph encoded at the Alphabetic Layer only.</p>
        </editorialDecl>
    </encodingDesc>
```

Those elements encoded at both the graphematic and alphabetic layers might be marked with `<p decls="#ag">`, but this is not necessary because of the `@default="true"` attribute above: this means that all paragraphs having no `@decls` attribute at all, default to `@decls="#ag"` (encoded at both layers).

In this way, while the edition model framework remains the multi-layered one of the Ursus edition, each section of each manuscript transcription can flexibly implement the model in full (all layers) or in part, based on specific needs, as well as on the project scope and time frame.

However, those paragraphs that are encoded also at the graphematic layer still need a faster and more efficient encoding strategy for abbreviations.

A simple solution comes from the fact that the abbreviation system of pre-modern handwriting… is systematic. Most of the times, a specific abbreviation (e.g. “ꝑ” or “p” plus macron) means a specific sequence of alphabetic letters (“per” or “pre” respectively). The practice that I am following in the Romualdus edition (and will document in detail in the documentation) takes advantage of the systematic nature of ancient abbreviations:

- I created a CSV table for each manuscript of Romualdus’ work (“table of standard abbreviation combinations”) mapping common abbreviation combinations (e.g. “p” plus macron) to their standard alphabetic value (“pre”). The alphabetic meaning of one-glyph brevigraphs such as “ꝑ” (“pro”) is already provided in the “graphematic table of signs”.
- In the source TEI XML of the transcription, when an abbreviation in a specific point of the text has its standard meaning/expansion I only encode its graphemes (e.g. “`p0`”, where “`0`” is the Unicode character chosen for the *encoding* – not the visualization – of the macron). In this case, I do not mark the abbreviation with `<choice>` / `<abbr>` / `<am>` / `<expan>` in the source code because the software (a Python script), based on the documentation, can easily check the CSV “abbreviation combinations” file and identify the string “`p0`” as a standard abbreviation to be expanded to “`pre`” at the alphabetic layer.
- If, instead, an abbreviation does not have a standard alphabetic value as mapped in the CSV “table of standard abbreviation combinations”, I encode the abbreviation in full with `<choice>` / `<abbr>` / `<am>` / `<expan>`. Those cases, however, represent a minority of the abbreviations actually found in a manuscript.

I consider this faster practice equivalent, in terms of information content, to the practice (followed for Ursus) of encoding all abbreviations with `<abbr>`, since the CSV table combined with regular expression (regex) matching software ensures that abbreviations are formally identified and represented at the graphematic and alphabetic layers.

5.3 Linguistic layer: lemma/PoS tagging or normalized spelling?

Issue. The new approach tested in the Ursus edition to identify inflected words at the linguistic layer (through `@lemma` and `@ana` attributes of `<w>`) had a shortcoming: the amount of work needed to review the output of *TreeTagger*, the lemmatizator/PoS tagger.

Falling back on the usual approach to representing of this textual layer, one might simply provide a “normalized” spelling of each word (the function of the `@n` attribute in `<w>` in the Ursus project), as in `<w n="usque">usq9</w>`. This certainly is a much more straightforward encoding process, to the point that it may be considered complete for the Ursus edition itself. But what would be the value of `@n` – the “normalized” transcription – of “pr*e*positionibus” (written in the manuscript with the medieval spelling “-e-” instead of classical “-ae-”)? Choosing classical “pr*ae*positionibus” is disputable from the point of view of cultural history, while choosing “pr*e*positionibus” would bring about those issues connected with alternative spelling, which affect all computational methods of textual analysis (search, indexing, collation, lemmatization etc.).

Solution. A Draconian shortcut might consist in disposing of the linguistic layer altogether. With medieval Latin texts, the alphabetic layer is probably sufficient enough to provide scholars with a readable text. But for textual searches, indexing, automatic collation, lemmatization and any other form of textual analysis, it proves inadequate. For the time being, I am applying the “Draconian” solution of omitting the linguistic layer. If time will suffice, I will apply the “normalized spelling” solution (`<w n="usque">usq9</w>`) at a later stage of the project.

6. SOFTWARE AND LICENSES

In the Ursus project, a large JS script processed the TEI XML code and the CSV “graphemic table of signs”. The script manipulated the DOM of an HTML file dynamically in the browser and visualized the edition. However, the browsers required 7-10 seconds to load the page. In the Romualdus edition all such tasks will be performed statically by a Python script using the lxml library.

All files and software of the new edition will be, as it was in the Ursus project, open source and designed with interoperability and reuse in mind. As a blind reviewer of this abstract correctly pointed out, the Ursus software as such was “too project oriented to be reused by other scholars in other contexts”. This may be the case for the Romualdus software as well, as its tasks are narrow: all it has to do is generating an HTML visualization of the TEI XML transcription. based on the CSV tables of signs and of standard abbreviation combinations. My aim, in any case, is to allow scholars to read the Ursus and Romualdus software source code, so they can find ideas to write their own code, should they create other multi-layered editions following a similar model.

7. REFERENCES

[1] Brüning, Gerrit, Katrin Henzel, e Dietmar Pravida. 2013. «Multiple encoding in genetic editions: the case of “Faust”». *Journal of the Text Encoding Initiative*, n. 4. doi:10.4000/jtei.697.

[2] Driscoll, Matthew J. 2006. «Levels of Transcription». In *Electronic Textual Editing*, a cura di Lou Burnard, Katherine O’Brien O’Keeffe, e John Unsworth. Modern Language Association of America. http://www.tei-c.org/About/Archive_new/ETE/Preview/driscoll.xml.

[3] Garufi, Carlo Alberto, a c. di. 1914. *Romualdi Salernitani Chronicon (A.m. 130-A.C. 1178)*. Vol. 127. Rerum italicarum scriptores. Città di Castello: S. Lapi.

[4] Haugen, Odd Einar. 2004. «Parallel Views: Multi-level Encoding of Medieval Nordic Primary Sources». *Literary and Linguistic Computing* 19 (1): 73–91. doi:10.1093/llc/19.1.73.

[5] Huitfeldt, Claus. 2006. «Philosophy Case Study». In *Electronic Textual Editing*, a cura di Lou Burnard, Katherine O’Brien O’Keeffe, e John Unsworth. Modern Language Association of America. http://www.tei-c.org/About/Archive_new/ETE/Preview/huitfeldt.xml.

[6] Matthew, Donald J. A. 1981. «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.

[7] Monella, Paolo. 2014. «Many Witnesses, Many Layers: The Digital Scholarly Edition of the Iudicium Coci et Pistoris (Anth. Lat. 199 Riese)». A cura di Fabio Ciotti. *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. doi:10.13133/978-88-98533-27-5. http://digilab2.let.uniroma1.it/ojs/index.php/Quaderni_DigiLab/article/view/190.

[8] Monella, Paolo. 2016. «Livelli di rappresentazione del testo nell'edizione del De nomine di Orso Beneventano». In *AIUCD 2016 Book of Abstracts. Digital editions: representation, interoperability, text analysis and infrastructures*, a cura di Federico Boschetti, 53–56. AIUCD. http://www.himeros.eu/aiucd2016/c20.pdf.

[9] Mordenti, Raul. 2011. *Paradosis. A proposito del testo informatico*. Memorie lincee Scienze morali, storiche, filologiche IX. Roma: Accademia Nazionale dei Lincei.

[10] Orlandi, Tito. 2006. «Edizione digitale sperimentale di Niccolò Machiavelli, De principatibus». http://www.cmcl.it/~orlandi/principe/.

[11] ———. 2010. *Informatica testuale. Teoria e prassi*. Roma: Laterza.

[12] Pierazzo, Elena. 2015. *Digital Scholarly Editing: Theories, Models and Methods*. Farnham, Surrey, UK and Burlington, VT: Ashgate.

[13] Robinson, Peter. 2014. «Exactly what are we transcribing?» *Textual Communities*. http://www.textualcommunities.usask.ca/web/canterbury-tales/blog/-/blogs/exactly-what-are-we-transcribing-.

[14] Sampson, G. 1990. *Writing Systems: A Linguistic Introduction*. Stanford University Press.

[15] Stokes, Peter A. 2011. «Describing Handwriting, Part IV: Recapitulation and Formal Model». *DigiPal Blog*. http://www.digipal.eu/blogs/blog/describing-handwriting-part-iv/.

[16] Zabbia, Marino. 2004. «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».