Many witnesses, many layers: the digital scholarly edition of the *Iudicium coci et pistoris* (*Anth. Lat.* 199 Riese)

Paolo Monella[1]

[1] Centro Linceo Interdisciplinare "B. Segre", Accademia dei Lincei, Rome, Italy
paolo.monella@gmx.net

Abstract. This article will describe the rationale of the digital scholarly edition on which I am currently working (the *Iudicium coci et pistoris* by Vespa, *Anth. Lat.* 199 Riese), an ancient Latin text with a multi-testimonial textual tradition. My edition aims to provide a proof-of-concept application of the ideas of Tito Orlandi and Raul Mordenti while using a customised XML/TEI markup.
In my edition, each witness will be encoded at three layers:

1. the graphic layer (graphemes and other graphic signs);
2. the alphabetic layer (alphabetic letters);
3. the linguistic layer (inflected words).

The main proposed innovations are as follows:

1. the distinction of different textual layers within each witness;
2. the declaration of 'tables of signs' for the graphical and the alphabetical layers of each witness;
3. each layer of a witness will be collated with the corresponding layers of other witnesses.

Keywords: textual layers; encoding levels; table of signs; collation; characters; graphemes; alphabemes; manuscripts; XML/TEI.

1 The conceptual framework

1 The text edited

I am currently working on a scholarly digital edition of the *Iudicium coci et pistoris iudice Vulcano* by Vespa, a Latin mock court debate in verse between a cook and a baker, written in the Roman late imperial age and included by Riese in his *Anthologia Latina* as poem number 199[1]. The edition will contain a number of experimental features. While its realisation is still in progress, the aim of this article is to reveal and discuss the rationale of these innovations, as well as the open issues which arise from them[2].

1 Its oldest witnesses are the *Codex Salmasianus* (Parisinus 10318, VII or VIII Century) and the *Codex Thuaneus* (Parisinus 8071, IX/X Century). Modern editions, after Bücheler and Riese 1894-1926, include Pini 1958, Shackleton Bailey 1980, Baumgartner 1981, Shackleton Bailey 1982 (where the poem is number 190), Barry 1987 and Lespect 2005. Omont 1903 is a photographic reproduction of *Codex Salmasianus*, but the codex is now fully digitised and openly accessible in Gallica at http://gallica.bnf.fr/ark:/12148/btv1b8479004f.

2 I wish to thank James Pearson-Jadwat for his precious advice, going well beyond a mere linguistic revision of my English.

2 A proof of concept

This edition aims to be a proof of concept. It is my aim to ascertain whether the sophisticated theoretical and methodological reflections of Tito Orlandi and Raul Mordenti on digital scholarly editions may be implemented in a prototype through a sustainable work-flow[3]. My experiment addresses two issues central to the current debate in digital philology: manuscript encoding and manuscript collation. The main ideas by Tito Orlandi that I endeavour to apply are[4]:

1. the encoding of each witness's text *at different textual layers*, including the graphical, alphabetic and linguistic ones;
2. the explicit declaration of a *table of signs* for each of the 'lowest' layers (graphical and alphabetic) of each witness;
3. *collation between witnesses by layer*: e.g. the linguistic layer of MS A should be collated with the linguistic layer of MS B, and the alphabetic layer of the former with the alphabetic layer of the latter.

As my aim is to produce a prototype, the edition model came before (and was appropriate to) the choice of text to edit. I chose the *Iudicium coci et pistoris* because

1. It is an ancient Latin work with a multi-testimonial textual tradition – as we, quite remarkably, have no digital scholarly edition of 'canonical' texts of classical antiquity[5].
2. It has a (small) multi-testimonial handwritten textual transmission – so I can devise mechanisms for collation between witnesses with different writing systems.

3 Textual layers

In my edition, I have decided to encode the text of each manuscript at three different textual layers:

1. the 'graphical' layer, whose minimal units of encoding are graphemes and paragraphematic signs (like punctuation);
2. the 'alphabetic' layer, whose minimal units are alphabetic letter ('alphabemes', from now on[6]);
3. the 'linguistic' layer, whose minimal units are inflected words.

The choice of these three layers among the many others possible (lemmatical, allographic, idiographic, etc.[7]) is arbitrary, and responds to the scientific purposes of a specific edition[8]. Ideally, a digital edition should be a modular and 'open source' digital object open to integrations by other scholars: another editor should be able, for example, to add an 'allographic' layer to my edition, if he so chooses[9].

---

3 See (at least) Orlandi 1999, Orlandi 2010 and his *Edizione digitale sperimentale di Niccolò Machiavelli, De principatibus* (`http://rmcisadu.let.uniroma1.it/~orlandi/principe`, last retrieved 10.03.2013); Mordenti 2001, Mordenti 2012 and his *Edizione Critica Ipertestuale dello Zibaldone Laurenziano (Pluteo XXIX.8) autografo di Giovanni Boccaccio* (`http://rmcisadu.let.uniroma1.it/boccaccio`, last retrieved 10.03.2013).

4 See particularly Orlandi 2010.

5 I propose an epistemological explanation for this in Monella, forthcoming.

6 The term 'alphabeme' was suggested to me by Raul Mordenti in an email in December 2012.

7 I am here referring to the terminology fixed by Peter Stokes for the DigiPal Project (`http://www.digipal.eu/blogs/blog/describing-handwriting-part-iv`, last retrieved 10.03.2013).

8 For instance, the 'allographic' and 'idiographic' layers are relevant for Mordenti's edition of the *Zibaldone Laurenziano* by Boccaccio (see footnote 2 above).

9 On the concept of an open source critical edition, see Bodard and Garcés 2009.

4 Graphemes and alphabemes

It should firstly be pointed out that graphemes and alphabemes (alphabetic letters) are not the same thing[10]. In the *Glossary of Unicode Terms*[11], a "letter" (here called an 'alphabeme') is defined with no reference to a graphical representation as "An element of an alphabet", while a grapheme is defined as "A minimally distinctive unit of writing in the context of a particular writing system". In my terminology, both grapheme 'j' and the Morse code / · – – – / *represent* alphabeme 'j'[12]. Likewise, allographs like 'capital J' and 'lowercase j' *represent* grapheme 'j', and idiographs like my individual handwritten sign for 'lowercase j' *represent* allograph 'lowercase j'.

5 Five Gutenbergian assumptions

Our approach to digital textual encoding tends to be influenced by the standardisation of graphic systems brought about by print in the last centuries, as well as by a number of related assumptions that are simply not valid for ancient manuscripts:

1. Standard alphabet: all witnesses of a text share a standard alphabet (a set of alphabemes).

- A counterexample might be a digital edition whose witness base includes a medieval manuscript with no distinction between alphabemes 'u' and 'v', and a modern print edition which makes that distinction. Likewise, some Middle English manuscripts include the 'thorn' alphabeme ('þ') and some do not.

2. Standard graphic system: all witnesses of a text share a standard graphic system (a set of graphemes, punctuation, capitalisation conventions etc.).

- On the basis of the Unicode definition of a grapheme mentioned above, I consider handwritten systematic abbreviations (like 'ē' for 'em') to be graphemes. Such abbreviation conventions vary greatly between manuscripts and normally do not exist in modern print editions. Other aspects of graphic systems which were not standardised until the invention of print include punctuation and comparable forms of 'graphic markup'.

3. Standard spelling: a specific sequence of alphabemes (e.g. 'w', 'i', 'f' and 'e') can be taken as a standard representation of an inflected word (e.g. the singular of 'wife'). Put simply, there exists a standard spelling for each word.

- Spelling is notoriously variable at all stages of a language's historical development. Even in modern European languages, spelling was (almost) completely standardised only recently. Also, old variant spellings of English still exist ('color'/'colour') and new ones

10 See Emiliano 2011, 154-157, where a clear distinction is drawn between "letter" (my 'alphabeme'), "character" and "glyph".

11 `http://www.unicode.org/glossary`, last retrieved 10.03.2013.

12 An example might clarify my conceptual distinction. Latin 'j' is an *alphabeme* in that it belongs to the modern English *alphabet* (but not to the traditional Italian alphabet that I learnt in my first grade). It also belongs to the Spanish and to the French alphabet, but in each of them it corresponds to a different *phoneme*. An alphabeme is an abstract cultural concept. Latin alphabeme 'a' is different to Greek 'alpha'. If I want to convey the message 'alphabeme j', either as a part of a spelled word or as the name of a specific section of a tax refund form, I can use a range of signs that all *represent* that alphabeme (but obviously *are not* that alphabeme): I can write my idiograph for the corresponding *grapheme* 'j' on a piece of paper, I can (rather unknowingly) enter in a computer the Unicode codes U+006A (small) or U+004A (capital), I can use the corresponding fingerspelling handshape of American Sign Language, I can pronounce the phonetic chain /dʒeɪ/ or, if I am a sophisticated student in search of ingenious ways to cheat on exams, I can use the Morse code / · – – – /.

keep emerging ('night'/'nite'). Latin, to name another Western *interlingua* that at some point was fixed in a 'standard' form, features 'deviant' spellings in archaic inscriptions and in the treatment of diphthongs like 'ae'.

4. Standard sequentiality: graphemes in a written text form are in an ordered sequence flowing in one direction only.

- A counterexample is provided by the Devanagari Indic script. It normally flows from left to right, but a vowel that phonetically follows a consonant may be written 'before' (to the left of) that consonant[13]. A case more familiar to Western scholars is that of Greek iota subscript ('ᾠ'), but many more examples could be taken from the Arabic script conventions for vowels and from the medieval European custom of writing some letters above others.

5. One grapheme, one alphabeme: there exists a one-to-one correspondence between alphabemes and graphemes in writing.

- This is not only contradicted by the use of ideographs and logographs in Western scripts (like '&' or the Tyronian note), but also by the existence of alphabetic writing systems that systematically do not write a grapheme for each alphabeme, like Arabic and Hebrew, or that make a extensive use of systematic abbreviations, like ancient and medieval Latin and Greek writing systems before the invention of print. In the latter case, as noted above, I consider the final abbreviation in 'regē' as one grapheme ('ē') corresponding to two alphabemes ('e' and 'm'). I should emphasise further that in such cases the correspondence of *one* grapheme to *many* alphabemes was *systematic*: this is how a medieval scribe learned to write, and he did so even when writing or copying an important or sacred text on a luxury codex. It is also worth noting that assumption no. 5 is probably the reason why contemporary Western literates find it hard to distinguish between the concepts of alphabeme and grapheme.

2 TEI issues and solutions

6 Graphic/alphabetic layers in TEI?

Does TEI already provide mechanisms to distinguish formally between the graphic and alphabetic layers when encoding a text to which the assumptions above do not apply? Let us consider the case of abbreviations. It is possible in principle that a theoretically conscious and consistent use of elements like `<abbr>` and `<ex>` might provide means to encode a witness's grapheme/alphabeme distinction in a concise and 'economic' though formally unambiguous way. However, due to the (intentionally) loose definition of those elements in the *Guidelines* and to the different encoding conventions allowed, the current TEI encoding of abbreviations – and above all its practical application in extant editions – does not appear to provide a reliable mechanism for that task[14]. Indeed, the issue is more general: simply, the TEI P5 Guidelines do not postulate any distinction between graphemes and alphabemes. The general concept of 'character' in TEI P5 assumes a triple correspondence: one grapheme in the document, one alphabetic letter in the abstract text, one digital code point in the XML file[15].

13 See Fiormonte 2012a, 68 (corresponding to page 9 in the PDF file in `http://www.cceh.uni-koeln.de/files/Fiormonte_final.pdf`) and Perri 2009, 736.

14 Methodological reflections on digital 'diplomatic' and 'normalised' editions like Driscoll 2006 and Pierazzo 2011 are not based on the grapheme/alphabeme distinction.

7 A Saussurean issue

This brings us to a second, more complex, issue. The TEI Guidelines strongly recommend that in Digital Humanities projects, 'characters' are encoded *according to the Unicode standard*[16]. The Unicode system, in turn, is based on the fact that a 'LATIN SMALL LETTER U' is a 'u' (i. e., despite the name 'letter', the same grapheme) in all written documents, from Late Antiquity parchment codices to contemporary websites, and as such it should be encoded with the same code point (U+0075).

This 'Unicode-compliance' principle works fairly well for the digitisation of those post-Gutenberg print documents that feature a standardised alphabet, graphic system, spelling, sequentiality and a one-to-one correspondence between grapheme and alphabeme. However, this principle does not suffice *per se* as a general principle for the encoding of medieval manuscripts or other documents that do not follow our 'standard' graphic system. Even worse does this principle serve us when we try to collate (or search through) a set of those documents[17]. The reason for this is that a 'u' *is not* always just a 'u': Ferdinand de Saussure taught us that signs have a relational nature *within* a specific semiotic system. This is echoed by the Unicode definition of a grapheme as "A minimally distinctive unit of writing *in the context of a particular writing system*"[18].

Imagine two medieval English manuscripts, A and B: both are copies of the same text, and a philologist is building a digital scholarly edition upon them. The graphic system of MS A has a distinction between grapheme 'u' and grapheme 'v', while MS B does not have that distinction and uses one grapheme ('u') to represent both alphabeme 'u' and alphabeme 'v'[19]. MS A reads: "The loue of love" ('the hill of love'). MS B reads: "The loue of loue". At the linguistic layer, this is the same text, just encoded with different graphemes.

---

15 The two key sections of the TEI P5 Guidelines on the encoding of 'characters' are *vi. Languages and Character Sets* (`http://www.tei-c.org/release/doc/tei-p5-doc/en/html/CH.html`) and *5. Representation of Non-standard Characters and Glyphs* (`http://www.tei-c.org/release/doc/tei-p5-doc/en/html/WD.html`). Section *vi* (#D4-42) makes a useful distinction between "abstract characters" (which seem to coincide with my 'graphemes') and "glyphs" (seemingly 'allographs' in Stoke's and my own terminology, i.e. graphic variants of the same 'abstract character', with no distinctive value). However, I still can neither find any conceptual distinction between grapheme and alphabetic letter (my 'alphabeme') in that section, nor elsewhere in the TEI P5 Guidelines. Compare Orlandi 2010, 8-9 and 48; Emiliano 2011, 154-157.

16 I am now using 'character' and 'glyph' according to the TEI P5 (`http://www.tei-c.org/release/doc/tei-p5-doc/en/html/CH.html#D4-42`) and Unicode terminology (`http://unicode.org/glossary`); both links were last retrieved on 19.03.2013. The 'Unicode-compliance principle' is described in sections *vi. Languages and Character Sets* and *5. Representation of Non-standard Characters and Glyphs* of the TEI Guidelines (mentioned in footnote 5) and discussed in Wittern 2006.

17 See Orlandi 2010, 9 and 48.

18 The Unicode definition of "grapheme" is in the *Glossary of Unicode Terms* (`http://unicode.org/glossary`). See Orlandi 2010, 9 and Emiliano 2011, 157-158: "Now consider a similar proposition: '*A* is a grapheme'. It is neither true nor false, it is simply meaningless. Because the grapheme (like the phoneme) is a linguistic relational concept, this proposition can only have truth or falsehood content in relation to a specific language. The definition of 'emic' units, contrary to 'etic' units, is a function of their status in a given symbolic system […] The grapheme, like the phoneme, is an abstract unit, whose value is defined in terms of the relation between elements of the same type in a system".

19 In their turn, 'u' and 'v' are distinct alphabemes because in the Middle English *linguistic* system there exists at least a pair of words that are distinct by the 'u/v' difference: an example is the pair 'loue' ('hill', in modern English) *vs.* 'love' ('love'), which I am using here as an example.

If we encoded both manuscripts following the TEI Unicode-compliance principle, the result would be as shown in Fig. 1:

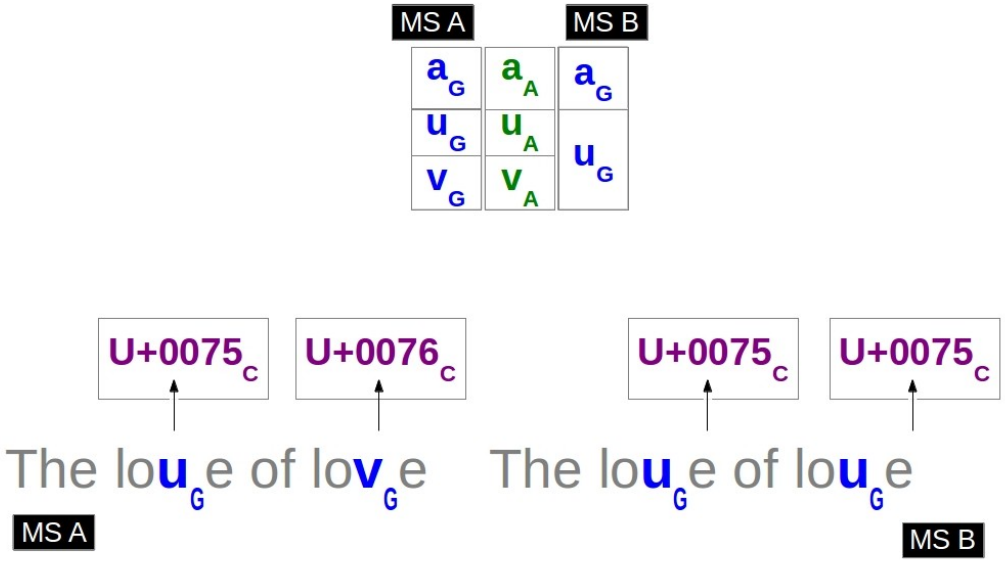

Fig. 1. Encoding of graphemes from documents with different graphic systems by use of the TEI 'Unicode-compliance' principle. 'G' stands for grapheme, 'A' for alphabeme and 'C' for code point.

This entails that a piece of software for collation or cross-corpus searching could not avoid being deceived by the apparent identity of the 'u' grapheme in MS A and the 'u' grapheme in MS B (as both are encoded with the Unicode code point U+0075). Similarly, it may appear to it that the 'v' grapheme in MS A is *not the same as* the 'u' grapheme in MS B (as they are encoded with different Unicode code points, U+0075 and U+0075). This is false, as the 'u' grapheme in MS A, defined contrastively by its opposition to the 'v' grapheme, *is not the same* grapheme as the 'u' grapheme in MS A, which has no such opposition and represents both alphabeme 'u' and alphabeme 'v'.

8 My proposed solutions: the 'musical score' model and tables of signs

As discussed in the previous two paragraphs, if I want to use TEI P5 to encode my manuscripts at different layers with a formal distinction between graphemes and alphabemes, I have two main issues to solve:

1. *The layers distinction issue* (see *6. Graphic/alphabetic layers in TEI?* above). My proposed solution is a model of digital scholarly edition resembling a *musical score*, with three parallel sequences of aligned tokens (graphemes, alphabemes and inflected words). The general model is discussed in *9. The abstract data model* below. Possible linearisations of this data model will be expanded upon in *12. Separate files linearisation model* and *13. Menota linearisation model*.
2. *The 'Saussure' issue* (see *7. A Saussurean issue* above). My proposed solution is an

implementation of Tito Orlandi's concept of the '*table of signs*'[20]. Namely, I am pairing the encoding of each witness with a specific table of graphemes and a specific table of alphabemes. This will be discussed in paragraph *14. A Saussurean solution: the tables of signs*.

3 The 'musical score' model

9 The abstract data model

As mentioned above, my edition model resembles a musical score, with three parallel transcriptions of the text (graphic, alphabetic and linguistic) mapped to one another at the granularity level of single graphemes.

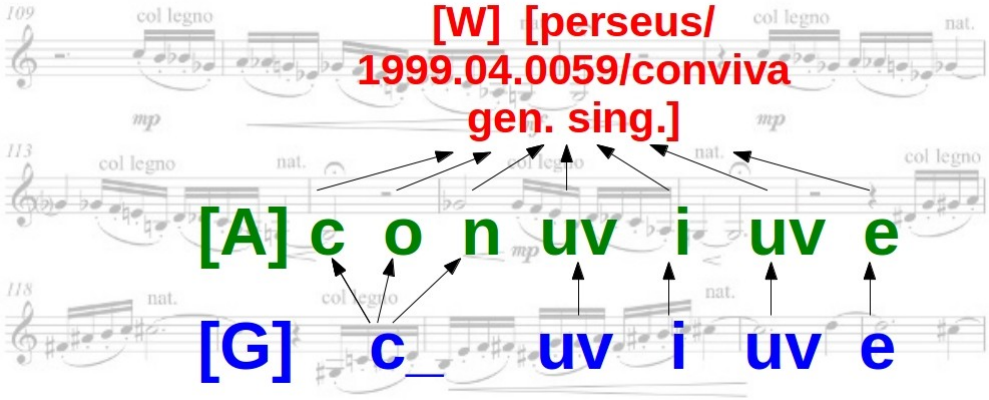

Fig. 2. The 'musical score' model. [G] stands for graphical layer, [A] for alphabetic layer and [W] for linguistic layer (a sequence of inflected **w**ords)

This edition model is approximated in Fig. 2[21]. In this example, grapheme 'c_' at the graphic layer (identifying a systematic abbreviation for 'con') is mapped to three alphabemes ('c', 'o' and, 'n') at the alphabetic layer. The manuscript hypothetically encoded here does not distinguish between the 'u' and 'v' alphabemes: it has a unique alphabeme with digital ID 'uv' (and a unique corresponding grapheme, whose ID is also 'uv').
The word is the genitive singular of 'conviva' ("table companion"). Note that today this inflected word would be spelled 'convivae'. Its alphabetic transcription in Fig. 2 does not constitute a 'regularised spelling'. It is simply a sequence of alphabemes 'as they are' in the manuscript[22], reflecting the medieval spelling 'convivae'.

20 See Orlandi 2010, 38-43.
21 The approximation lies in the fact that the elements of the graphic layer [G] should also be mapped to the linguistic layer [W].
22 i.e. as the philologist interpretively extracts them from the graphic encoding of the manuscript.

10 The linguistic layer: inflected words as atoms

The encoding units of the linguistic layer ([W] in Fig. 2) are inflected words. In Orlandi's view, each inflected word should not be represented as a sequence of letters, a simplistic solution that poses the issues of homographs and variant spellings, but by means of a unique digital identifier. This corresponds conceptually to a 'table of signs' for the linguistic layer, but clearly the attribution of an identifier to many thousands of words poses two kinds of issues. These are the conceptual and the practical:

Conceptual issues. If we do not identify inflected words by means of a sequence of digital characters (i.e. on the basis of alphabetic writing), how else can we identify them? A word might be identified by a combination of a lemma (with a unique ID retrieved from a standard digital vocabulary) and standardised morphological information. Therefore, Latin 'deum' will no longer be identified simply by a sequence of four Unicode code points, which would not tell us whether it is the accusative singular of 'deus' ('god') or the archaic form of its genitive plural (classical Latin 'deorum'). It would instead be encoded essentially as a combination of the following two pieces of information:

- The identifier of lemma 'deus' ('god') in Perseus' Lewis-Short dictionary: perseus/1999.04.0059/deus;
- A standardised encoding of: 'genitive, plural'.

It might be necessary, however, to add a third piece of information: the spelling of the word, i.e. the sequence of the four Unicode code points for 'deum'. This would allow us to distinguish between 'deum' and 'deorum', which otherwise would erroneously be encoded in the same way (lemma: 'deus'; morphology: genitive plural)[23].

Practical issues. It is obviously easier to identify inflected words by a sequence of 'characters', either keyed in by hand or obtained by OCR. Encoding every single word of every manuscript as a combination of a lemma and some morphological information implies that the philologist has to perform lemmatisation and morphological parsing on every word. The only solution is to make both operations semi-automatic with the help of dedicated software that makes the work-flow sustainable, though undoubtedly more burdensome for the encoder. This will be discussed in paragraph *19. Current work-flow* below.

11 Two possible XML linearisations of the 'musical score' data model

Let us go back to the 'musical score' data model and its possible linearisation. XML appears to be better suited to represent trees than three sequences of tokens flowing in parallel and aligned with each other at such a level of granularity. Therefore, I am still open to the option of turning to other text-encoding data models, including the range-based model elaborated by Gregor Middel and

23 This approach may look like a simple addition of lemmatisation and morphological parsing to a regular Unicode string ('deus'), and therefore an unnecessary complication. A key question in this respect is whether there exists such a thing as a homographic variant, i.e. whether two witnesses can bear two sequences of signs which resolve to the same alphabetic sequence but can be reasonably interpreted as different readings, due to some contextual information. There is no room for such a discussion here: see my talk Monella 2012, slides 42-48.

others within the Faust Project[24], Desmond Schmidt's multi-version documents[25] and Manfred's Thaller's extended strings[26].
At the moment, however, I am still experimenting on TEI/XML to see whether I can adapt it to serve the purposes of my edition model. My work on this is still in progress, so what follows is a mere working hypothesis. I shall now describe the two XML linearisation models I am currently testing:

1. Separate files linearisation model
2. Menota linearisation model

12 Separate files linearisation model

This model requires that, for every witness, three different TEI P5 'XML transcription files' are created[27]:

1. `salm_graphic.xml`
2. `salm_alphabetic.xml`
3. `salm_linguistic.xml`

`salm_graphic.xml`. This is the transcription of the text of the *Codex Salmasianus* at the graphic layer. It includes a sequence of TEI P5 `<g>` elements representing graphemes, paragraphematic signs (like punctuation) and other graphic signs (like spaces between words).
The following code is taken from the current graphic XML transcription file of the *Codex Salmasianus* and represents the first two words of the first line of the poem ('Ter ternae'):

```
<g id="10.1" ref="#T" />
<g id="10.2" ref="#e" />
<g id="10.3" ref="#r" />
<g id="11.1" ref="#t" />
<g id="11.2" ref="#e" />
<g id="11.3" ref="#r" />
<g id="11.4" ref="#n" />
<g id="11.5" ref="#ae" />
```

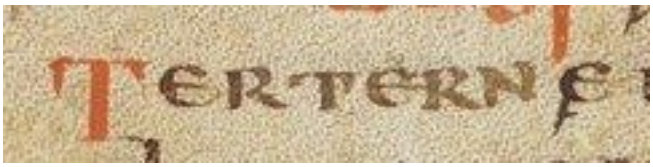

Fig. 3. The first two words of the first line of the *Iudicium coci et pistoris* in the *Codex Salmasianus*: 'Ter ternae'

24 The Faust Project is also elaborating a data model in which documents are encoded at more than one level, namely two: "dokumentarische" and "textuelle Transkript" (the latter is similar to my linguistic layer). See Bohnenkamp et al. 2011, especially section *III. Umsetzung* (pp. 38-65) and Brüning, Henzel, Pravida 2012.
25 Schmidt & Colomb 2009; Schmidt & Fiormonte 2010.
26 Thaller 2006.
27 The 'salm_' prefix of filenames denotes that they refer to the encoding of *Codex Salmasianus*, the first witness I am encoding.

The @id attributes have the usual function of marking each grapheme unambiguously and sequentially. For readability's sake, the value of @id is the number of the word ('Ter' is the 10th word of the transcription, as it is preceded by the words of the title line), followed by a dot and the number of the grapheme within the word. The manuscript has no spaces between words (*scriptio continua*), so the delimitation of a sequence of graphemes as a 'word' is an interpretive act of the philologist. Also note that, as there are no graphic signs (spaces) to mark the distinction between words, in the graphic transcription above there is no element marking the distinction between the two words. Line breaks between verses, instead, are reported in this file because although they are not not ink marks, they are still graphic signs found in the document.
Grapheme 10.1 above is an uppercase Latin 'T': as it is a grapheme (not an alphabeme), it is distinct from a lowercase Latin 't'. Grapheme 11.5 (the last sign on the right in Fig. 3) has the two alphabemes 'a' and 'e' as alphabetic content, and 'ae' as digital identifier. I shall turn back to the semantics of the @ref attributes and on the use of <g> elements in paragraph *15. The table of graphemes* below.

salm_alphabetic.xml. This is the transcription of the same witness at the alphabetic layer. It includes a sequence of TEI P5 <g> elements which, in this file, represent alphabemes. What follows is a sampling of the current salm_alphabetic.xml file:

```
<g id="10.1.1" ref="#t" />
<g id="10.2.1" ref="#e" />
<g id="10.3.1" ref="#r" />
<g id="11.1.1" ref="#t" />
<g id="11.2.1" ref="#e" />
<g id="11.3.1" ref="#r" />
<g id="11.4.1" ref="#n" />
<g id="11.5.1" ref="#a" />
<g id="11.5.2" ref="#e" />
```

In fact, the introduction of a new <alphabeme> element would theoretically be a much better solution. This is still an option on the table. However, for the time being I am using the existing <g> element while giving it different semantics than in file salm_graphic.xml. Also the value of @ref attributes in the code above has different semantics, as will be discussed in paragraph *16. The table of alphabemes* below.
In the value of the @id element of alphabeme 11.5.1 above, the three numbers mean that this is the 1st *alphabeme* ('a') represented by the 5th *grapheme* (the abbreviation for diphthong 'ae') of the 11th *word* ('ternae').

salm_linguistic.xml. This is the transcription of the same witness at the linguistic layer. It includes a sequence of TEI P5 <w> ('word') empty elements. They are empty because words, in this model, are not represented by a sequence of alphabemes. This is the part of the model that still requires most work. The following code simply aims to give an idea of the general concept:

```
<w id="10" ana="adv,[ter],ter" />
<w id="11" ana="adj,[ternus],n,p,f,ternae" />
```

At the current stage of development of the model, a generic @ana attribute includes the digital identifier of the inflected word: in the example above, word 11 is the *n*ominative *p*lural *f*eminine of lemma 'ternus' (which is an *adj*ective), whose contemporary standard spelling is 'ternae'. At this

stage, this notation is nothing more than a placeholder. At a later stage in the development of the prototype, this information will be distributed into the relevant analytical attributes, like `@type`, `@lemma`, `@lemmaRef` etc.[28]
The elements of the three files above (graphemes, alphabemes and inflected words) must be aligned with each other. This task is performed by three more 'alignment files':

1. `salm_align_alph_graph.xml`
2. `salm_align_alph_ling.xml`
3. `salm_align_graph_ling.xml`

These files include `<link>` and `<ptr>` elements only. Their function is to perform the mapping described in Fig. 4. I shall now briefly describe how the alignment information is encoded in each file.

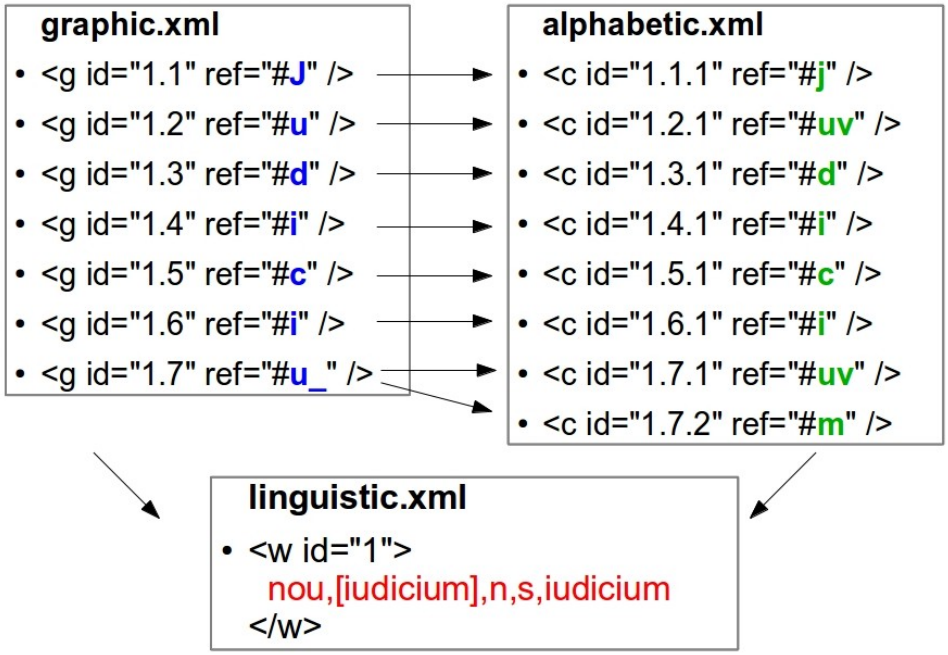

Fig. 4. Alignment between the three separate XML transcription files

`salm_align_alph_graph.xml`. There must be a formal way to encode the information so that, for example, grapheme 11.5 in `salm_graphic.xml` (abbreviation 'ae') corresponds to alphabemes 11.5.1 ('a') and 11.5.2 ('e') in `salm_alphabetic.xml`. This function is performed by a number of TEI P5 `<link>` elements stored in a separate file. The XML file aligning `salm_alphabetic.xml` and `salm_graphic.xml` is currently named `salm_align_alph_graph.xml`. This is a portion of its content:

```
<link targets="salm_graphic.xml#11.5 #11.5" />
```

28 Section *17 Simple Analytic Mechanisms* of the TEI P5 Guidelines (`http://www.tei-c.org/release/doc/tei-p5-doc/en/html/AI.html`, last retrieved 18/03/2013).

```
<ptr id="11.5" targets=
        "salm_alphabetic.xml#11.5.1
        salm_alphabetic.xml#11.5.2" />
```

The <ptr> element is needed to encode a one-to-many link in TEI P5 (one grapheme must here point to two alphabemes)[29].

salm_align_alph_ling.xml. This file aligns the alphabetic transcription with the linguistic one, i.e. an inflected word with a sequence of alphabemes that represent it. This is the snippet of code that aligns word 11 ('ternae') with the corresponding alphabemes:

```
<link targets="salm_linguistic.xml#11 #11" />
<ptr id="11" targets=
        "salm_alphabetic.xml#11.1.1
        salm_alphabetic.xml#11.2.1
        salm_alphabetic.xml#11.3.1
        salm_alphabetic.xml#11.4.1
        salm_alphabetic.xml#11.5.1
        salm_alphabetic.xml#11.5.2" />
```

salm_align_graph_ling.xml. This file aligns the graphic transcription with the linguistic one. The following code aligns the same word ('ternae') with the *graphemes* that encode it. Note that, while the word has 6 alphabemes (see code above), it is here linked to *only 5* graphemes, as the final diphthong 'ae' is one grapheme in the manuscript:

```
<link targets="salm_linguistic.xml#11 #11" />
<ptr id="11" targets=
        "salm_graphic.xml#11.1
        salm_graphic.xml#11.2
        salm_graphic.xml#11.3
        salm_graphic.xml#11.4
        salm_graphic.xml#11.5" />
```

The source code above does look complex at first sight, but of course none of it is written directly by the philologist. The writing of the actual XML files, the attribution of @id numbers and the simultaneous creation of the relevant links between graphemes, alphabemes and inflected words are all tasks performed by a small piece of software (currently a Python 3.3 script, input.py). What the philologist actually writes is a simple 'input transcription file' in CSV format (salmasianus.csv, for Codex Salmasianus) . The script inputs this file and outputs the three XML transcription files (salm_graphic.xml, salm_alphabetic.xml, salm_linguistic.xml) and the three 'alignment files' (which pair each transcription with the other two). This will be discussed in detail in paragraph *19 Current work-flow* below.
All XML, CSV and Python files described in this article are openly available in the GitHub repository https://github.com/paolomonella/vespa.git.

13 Menota linearisation model

While I was in search of an already existing customisation of TEI that could encode the text at

29 Section *16.1.4 Intermediate Pointers* of the TEI P5 Guidelines (http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SA.html#SAPTIP, last retrieved 18/03/2013).

different parallel layers, my attention was drawn by Roberto Rosselli Del Turco to The Medieval Nordic Text Archive (Menota)[30]. The project's customisation of TEI provides a mechanism to encode a text at three layers, named "facsimile", "diplomatic" and "normalised" (which, respectively, correspond roughly to my graphical, alphabetic and linguistic layers). To encode the text in this way, the Menota Project added three elements to the TEI/XML schema, namely `<me:facs>`, `<me:dipl>` and `<me:norm>`[31]. The resulting code looks like this:

```
<w>
   <choice>
      <me:facs>&drot;<am>&osup;</am>ttin<am>&bar;</am></me:facs>
      <me:dipl>d<ex>ro</ex>ttin<ex>n</ex></me:dipl>
      <me:norm>Dróttinn</me:norm>
   </choice>
</w>
```

The Menota encoding practice for primary sources is certainly a good basis to experiment upon, but it differs from my envisioned model in four ways:

1. *Granularity*. The finest granularity of the current Menota markup is at word level, while I need alignment at grapheme-level granularity.
2. *Grapheme/alphabeme distinction*. All three Menota transcription layers share the same set of 'characters', while my graphic and alphabetic transcriptions are based on different sets of elements (graphemes and alphabemes respectively).
3. *Unicode-compliance principle*. Menota relies on Unicode for the definition of each sign encoded, while I want to have a formal and explicit definition of each element (grapheme or alphabeme) used in the transcription (see paragraph *7. A Saussurean issue* above).
4. *Linguistic transcription*. My linguistic transcription layer should not encode inflected words as a sequence of letters, as Menota does, but by means of unique digital identifiers, as described in paragraph *10.The linguistic layer: inflected words as atoms* above.

Still, the Menota customisation of TEI P5 has the clear advantage of allowing the coexistence of three different layers of transcription in the same XML file. The following code illustrates a hypothetical combination of the Menota 'three-layers' innovation and my own proposed encoding of graphemes, alphabemes and inflected words:

```
<w id="11">
 <me:facs>
       <g id="11.1" ref="#grapheme_t" />
       <g id="11.2" ref="#grapheme_e" />
       <g id="11.3" ref="#grapheme_r" />
       <g id="11.4" ref="#grapheme_n" />
       <g id="11.5" ref="#grapheme_ae" />
 </me:facs>
 <me:dipl>
       <g id="11.1.1" ref="#alphabeme_t" />
       <g id="11.2.1" ref="#alphabeme_e" />
       <g id="11.3.1" ref="#alphabeme_r" />
       <g id="11.4.1" ref="#alphabeme_n" />
       <g id="11.5.1" ref="#alphabeme_a" />
```

30 Home page: `http://www.menota.org/EN_forside.xhtml` (last retrieved 18.03.2013). See Haugen 2004.

31 Paragraph *3.2 Levels of text representation* in *The Menota Handbook v 2.0*: `http://www.menot-a.org/HB2_ch3.xml#d481e406` (last retrieved 18.03.2013).

```
        <g id="11.5.2" ref="#alphabeme_e" />
  </me:dipl>
  <me:norm>
        adj,[ternus],n,p,f,ternae
  </me:norm>
  <link targets="#11.1 #ptr11.1" />
    <ptr id="ptr11.1" targets=
        "#11.1.1" />
  <link targets="#11.2 #ptr11.2" />
    <ptr id="ptr11.2" targets=
        "#11.2.1" />
  <link targets="#11.3 #ptr11.3" />
    <ptr id="ptr11.3" targets=
        "#11.3.1" />
  <link targets="#11.4 #ptr11.4" />
    <ptr id="ptr11.4" targets=
        "#11.4.1" />
  <link targets="#11.5 #ptr11.5" />
    <ptr id="ptr11.5" targets=
        "#11.5.1
        #11.5.2" />
</w>
```

Of course, this depends on the possibility of allowing, under the Menota scheme, a hierarchy like <w> / <choice> / <me:facs> (or <me:dipl> , or <me:norm>) / <g>[32].
As word/graphemes and word/alphabemes alignment is granted by the inclusion of Menota <me:facs> and <me:dipl> elements in the same <w> element, <link> elements are only required for graphemes/alphabemes alignment. <link> elements are now included in the same salm_menota.xml file.
In this hypothesis, issues 1 and 2 above (granularity and grapheme/alphabeme distinction) are overcome by the pervasive use of <g> elements, aligned through <link> elements. Issue 4 above (linguistic transcription) may be overcome by the introduction of non-alphabetic representation of inflected words in the <me:norm> element. My proposed strategy for the solution of issue 3 above (Unicode-compliance principle) is centred on the use of the <g> element and can be applied either to the 'Separate files linearisation model' or to the 'Menota linearisation model' equally. This strategy will be described in detail in the following paragraphs.

4 Tables of signs

14 A Saussurean solution: the tables of signs

To solve the 'Unicode-compliance' issue discussed in paragraph *7. A Saussurean issue* above, I propose a TEI/XML implementation of Tito Orlandi's idea of "tabella dei segni" (henceforth 'table of signs')[33].
In Orlandi's view, when a philologist digitally encodes the 'glyphs' of a manuscript or any other set of textual phenomena in a document, he or she must create an 'ideal' table (that can, however, become an actual table as part of an edition's documentation), the 'left column' of which consists

32 Some of the proposed modifications to the Menota encoding practice might require a further customisation of the Menota scheme.
33 Orlandi 2010, 38-43.

of the digital signs that they use to represent the textual phenomena, while the 'right column' "includes the list of single phenomena of the object of the encoding which one wants to encode". Orlandi's formulation is conceptually wide, and intentionally does not specify the nature of the textual phenomena of the 'right column'[34].
In this respect, I am proposing two innovations:

1. I am creating two different tables of signs: the table of graphemes and the table of alphabemes;
2. I am looking for a suitable format to include the two tables in my digital edition as a formal, computable and stable component of the XML source code, possibly in the `<charDecl>` section of the TEI header.

15 The table of graphemes

In the current stage of development of my edition, this is a simple CSV file (`salm_table_graphemes.csv`) with three columns, which my Python script (`input.py`) transforms into TEI/XML code and writes in the TEI header of the `salm_graphic.xml` and `salm_menota.xml` transcription files. Tab. 1 below shows several rows from my current table of graphemes for the *Codex Salmasianus*:

Tab. 1. A portion of the table of graphemes for the *Codex Salmasianus*.

Digital ID	Content of the grapheme (=alphabeme[s])	Expression of the grapheme (=glyph)	Visualisation
u	uv	u/v Latin minuscule uncial (u-shaped, not v-shaped)	u
z	z	Latin minuscule uncial z	z
ae	a,e	Latin minuscule uncial e with a tail bottom left, `img/ae.jpg`	ę

Again, note that this file and the others mentioned in this article (and available in `https://github.com/paolomonella/vespa.git`) are still at a prototype stage and are merely reported to give a practical idea of the path that my experimentation is currently following.
The 'Digital ID' column includes strings of Unicode characters, but they might just as well be ASCII characters or numerals. They unambiguously identify every sign (in this case, every grapheme) and correspond to Orlandi's 'left column'[35].
For the other two columns ('Grapheme: content' and 'Grapheme: expression'), a premise must be made: all digital elements upon which our digital textual encoding is based (Unicode characters, XML entities, `<g>` elements etc.) are signs. As such, they have an expression (e.g. a Unicode code point) and content. A key concept of my edition is that the *content* of such digital signs *is another sign* (e.g. a grapheme), which, in its turn, is constituted and defined by another expression/content

34 The quotation is from Orlandi 2010, 38 (the translation is mine). For his edition of the *Zibaldone Laurenziano* (see footnote 3 above), Raul Mordenti is creating a table of signs describing the set of glyphs identified in the manuscript ('right column', textual phenomena), mapped to a set of digital signs, i.e. ASCII characters or XML entities in the form `&abc;` ('left column'). This is an actual table meant be published with the documentation of the digital edition (Mordenti 2012).

35 Orlandi 2010, 38.

pair. For example, in the third row of the table above, the digital sign identified by the two Unicode characters ‘ae’ has these two digital characters as its expression, and the corresponding grapheme as its content. In its turn, this grapheme consists of the pairing of its expression (the ‘e with a tail’ glyph described in the third column and visible in the JPEG image `img/ae.jpg`) with its content (the two alphabemes ‘a’ and ‘e’, listed in the second column). Thus Orlandi’s ‘left column’, describing the textual phenomenon, is here represented by *two* columns (the second and third in table 1 above).
The forth column (‘Visualisation’) has an eminently practical function: it instructs visualisation software (XSLT or other technology) on what Unicode character(s) it should use to display that grapheme, if it is required to give a screen or print visualisation of the graphic layer of the text.

16 The table of alphabemes

This too is currently a simple CSV table (`salm_table_alphabemes.csv`), which the `input.py` Python script transforms into XML code. The table for the *Codex Salmasianus* has 20 rows, as this is the number of alphabemes that I identified in the codex[36]. This is a part of the table:

Tab. 2. A portion of the table of alphabemes for the *Codex Salmasianus*.

Digital ID	Description of the Alphabeme	Visualisation
t	Latin t	t
uv	Latin u/v	u
z	Latin z	z

There is a number of issues connected with this table which are still open. Alphabet and alphabemes are conceptual objects much more complex than they appear at first sight[37].
In the tentative Tab. 2 above, I did not describe alphabemes by means of an expression/content pair, simply because I am not sure of what that would be. One might think that the expression of an alphabeme is its corresponding grapheme, and its content is a phoneme. The very invention of the alphabet had the goal of making things just this simple. However, things are not so simple, because alphabets are cultural objects in themselves[38].

1. Is a phoneme the content of an alphabeme? What is the phonetic content of alphabeme ‘Latin letter c’ in *Codex Salmasianus*, a Latin document written in the VII or VIII Century CE in Vandalic Africa, where ‘letter c’ was probably pronounced in different ways (not fully known to us) according to the vowel that followed, and to the social and cultural status, geographic origin and ethnicity of the reader? Not to mention that the same ‘Latin letter c’ in the same text has been read in many different ways throughout the centuries by the many readers that happened to have that codex in their hands. In other words, ‘dead languages’ have the advantage of keeping alive in us a critical sense of the correspondence between alphabemes and phonemes. But even in contemporary languages, the same English ‘letter’ may be connected to slightly different phonemes in different national, regional, dialectal or socially determined pronunciations. This should already suffice to show that an alphabeme is defined *per se* as a cultural object, not as a

36 The second of the three rows in Tab. 2 shows that the *Codex Salmasianus* does not have two different alphabemes ‘u’ and ‘v’, but one only: this is why there are 20 rows in the table instead of 21, a more familiar number for the Latin alphabet.
37 Mordenti 2011, 640-648 has an interesting reflection on such complexity.
38 Suffice it to mention Sampson 1990.

content/expression (phoneme/grapheme) pair.

2. Is a grapheme the expression of an alphabeme? This hypothesis is conceptually less problematic, but is still rather simplistic. For instance, grapheme ‘b’ may represent alphabeme ‘b’, but grapheme ‘b_’ in Codex Salmasianus (a ‘b’ with a macron top right, serving as abbreviation for ‘bis’) also represents alphabeme ‘b’, as well as alphabemes ‘i’ and ‘s’.

I would still be ready to create a column in the table of alphabemes where grapheme ‘b’ is designated as the *main* expression (though this is a rather loose concept) of alphabeme ‘b’, but I would hesitate to create a column including the contents of alphabemes, as I currently would not know how to populate it.
The last column on the right in the table above, again, instructs the software on what Unicode character may be best used in visualising of the alphabetic layer of the text.
Lastly, it is important to note that the second column of the table of *graphemes* (Tab. 1) points to the IDs (first column) of the table of *alphabemes* (Tab. 2).

17 Inclusion of the tables of signs in the TEI header

As anticipated, it is my intention to make the two tables of signs an integral part of the source code in the XML transcription files. This is, however, very problematic and requires further work. Again, I shall here describe the hypotheses that I am currently working on.
If we adopt the ‘separate files linearisation model’ described in paragraph 12 above, in the *Codex Salmasianus* the table of graphemes should be integrated in file `salm_graphic.xml` and the table of alphabemes in file `salm_alphabetic.xml`. The most obvious place for these tables is the `<charDecl>` element in the TEI header.
TEI P5 features an interesting innovation: the philologist can define ‘non-standard characters’ in the `<charDecl>` element of the header, and then encode instances of them in the body by means of `<g>` elements. This appears to be a more sophisticated mechanism for formally defining a grapheme than XML entities, so I decided to adopt it in my edition.
A major problem, however, is that to my knowledge, there is no way in TEI to formally define simple Unicode characters.
In other words, if I encode something like `ui<g ref="ae" />` (for Latin ‘viae’ written with a final ‘æ’ grapheme), I can formally define the last grapheme (the ‘non-standard’ ‘æ’) in `<charDecl>`, but I have no way to define the first grapheme (the ‘standard’ ‘u’, for which I am using Unicode code point U+0075) anywhere. This is because the TEI Unicode-compliance principle described in *7. A Saussurean issue* above assumes that a ‘u’ is a ‘u’, and requires no further definition than the *absolute* one given by the Unicode standard. However, as my ‘Saussurean’ argument in paragraph 7 above and the very definition of grapheme in the *Glossary of Unicode Terms* postulate, a grapheme must be always defined in a *relative* way, “in the context of a particular writing system”[39]. In the context of a manuscript featuring a ‘u’/‘v’ distinction, a ‘u’ *is not the same thing* as a ‘u’ in the context of a manuscript with no such distinction. This implies that it must always be possible – in fact, that it should be required – to give a description of *all* graphemes identified in a manuscript, both ‘standard’ and ‘non-standard’. In the example above, not only the final `<g ref="ae" />`, but also the initial Unicode character U+0075 (for ‘u’)

39 The quotation here comes from the definition of “grapheme” in the *Glossary of Unicode Terms* (see footnote 18 above). The need for the ‘local’ definition of signs is postulated by Orlandi 2010, 38-43 (but see also pages 9 and 48).

should be matched by a (brief and formal) description in <charDecl>.
Of course, all that is being said about the need to define all graphemes also applies to alphabemes, allographs and the other kinds of signs that the philologist decides to encode.
At the moment, as I said, it is technically not possible to formally define Unicode characters, and I am afraid that such a change would require not only much work on the TEI schema, but also a quite radical modification in the approach towards 'characters' in the Guidelines, in which

1. it should be recommended to define *all* graphemes (and other signs), not ;
2. the very conceptual distinction between 'standard' and 'non-standard' characters should be eliminated.

As this seems to be more work than I am currently willing to undertake, for the time being I am confining myself to using only <g> elements (not Unicode characters) in the body of the XML transcription files, as I know that I can define them in the <charDecl>. I hope that a solution for this issue can be found in the future, allowing the philologist to simply encode ui<g ref="ae" /> rather than

```
<g ref="uv" />
<g ref="i"  />
<g ref="ae" />
```

Needless to say, it would be utterly infeasible to key the 'all <g>' source code above directly. This is another aspect for which the input.py Python script is of use. What I am currently keying is something like this: ui(ae). The script transforms this into the sequence of three <g> elements above. This will be further discussed in paragraph *19. Current work-flow* below.
The following code is taken from file salm_graphic.xml. It exemplifies the initial description of graphemes in <charDecl> and their subsequent use in <body>:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
  <!-- ... -->
  <charDecl>
    <!-- ... -->
    <char xml:id= "t">
      <charProp>
        <localName>Expression</localName>
        <value>Latin minuscule uncial t</value>
      </charProp>
      <charProp>
        <localName>Content</localName>
        <value>t</value>
      </charProp>
      <charProp>
        <localName>Visualisation</localName>
        <value>t</value>
      </charProp>
    </char>
    <!-- ... -->
  </charDecl>
</teiHeader>
<text>
<body>
  <!-- ... -->
  <g id="24.1" ref="#e" />
```

```
  <g id="24.2" ref="#t" />
  </text>
</body>
</TEI>
```

Likewise, alphabemes are described in the <charDecl> section of file salm_alphabetic.xml:

```
<char xml:id= "uv">
  <charName>Latin u/v</charName>
  <desc>Latin alphabeme 'u', with no 'u'/'v' opposition</desc>
  <charProp>
    <localName>Visualisation</localName>
    <value>u</value>
  </charProp>
</char
```

Note that the less structured description method for alphabemes (<desc>) mirrors the simpler current structure of the table of alphabemes (see Tab. 2 above).
If we adopt the 'Menota linearisation model' described in paragraph 13, in addition to the issues already discussed, a further complication arises: both tables (graphemes and alphabemes) should be included in the same salm_menota.xml file. Unfortunately, the TEI header is not designed to accommodate two different <charDecl> elements marked as belonging to two different textual layers. This too would require an *ad hoc* customisation of TEI.

18 Collation

As anticipated at the beginning of this article, my edition is still at an early stage, namely that of encoding witnesses, a stage which runs parallel to the elaboration of experimental encoding standards. The collation phase is yet to come, but I can here mention the two principles that will lead it.

1. Scribes and scholars. I mean to encode the text of modern and contemporary editions, and also scholarly editions, alongside that of ancient medieval manuscripts, thus making no artificial distinction between 'scribes' and 'scholars'. For the same principle, I shall produce 'my' text, by comparing the extant variants and my own *iudicium*, but this text will then be presented to the reader at the same level as any other text from ancient or modern witnesses[40].

2. Collation performed layer by layer. The *linguistic layer* of one witness will be collated with the same layer of others. This should be somewhat easier, as all witnesses' texts will be encoded at this layer by means of references to the same dictionary (for lemmas) and to the same formalised grammar (for morphology)[41].

40 I shall mention only some of the main reading that influenced me in this respect: Pasquali 1971; Reynolds & Wilson 1991 (whose title is "Scribes and scholars"); Cozzo 2006; Benozzo 2011; Fiormonte 2012b.

41 The most obvious candidate for a reference dictionary is the online version of the *Latin Dictionary* by Charlton T. Lewis and Charles Short in the *Perseus Digital Library* (http://www.perseus.tufts.edu/hopper, last retrieved 19.03.2013). A theoretical issue regarding lemmata and morphology is that both the vocabulary and the grammar of Latin evolved over the centuries, and one could argue that Virgil's text in a medieval manuscript should be mapped against the vocabulary and grammar of the manuscript's time – therefore not those of 'classical' Latin. This objection, however, can be partially rejected on the basis that the scribes/philologists normally meant to preserve the

Collating the *alphabetic layer* of a manuscript with the alphabetic layer of a modern print edition, however, will be more complicated, as two witnesses may well not share exactly the same alphabet (e.g. for the 'u' / 'v' distinction issue). Still, this problem can be solved with a formal table of alphabemes for each manuscript describing each reference alphabet. This might allow us to create, by means of RDF/OWL or other standards, a net of correspondences: for example, the 'uv' alphabeme in MS B might be connected to the 'u' and the 'v' alphebemes in MS A. This should suffice to instruct sophisticated collation software to compare two witnesses at this layer.
The very same issue will affect collation at the *graphic layer*, but on a much larger scale: just think of the enormous variance in the nature and use of punctuation in manuscripts. I am considering the option of skipping collation at this layer altogether – or alternatively adopting the same approach that I outlined for the alphabetic layer.

5 Work-flow

19 Current work-flow

I shall lastly describe the current work-flow of my edition. As said before, all working files described here are available in the the GitHub repository `https://github.com/paolomonella/vespa.git`, while further documentation and discussion on the open issues will be published in `http://www.unipa.it/paolo.monella/lincei/edition.html` as the work proceeds.
I am currently encoding the oldest manuscripts. What I actually edit 'by hand' is a CSV file for each manuscript. This is a snippet of the content of file `salmasianus.csv`, the current CSV transcription file for the *Codex Salmasianus* ('§' is the CSV delimiter character):

```
Ter           §     adv,[ter],ter
tern(ae)      §     adj,[ternus],n,p,f,ternae
uarias        §     adj,[varius],ac,p,f,varias
```

The first column includes the graphic transcription. This is what I actually key:

- If the grapheme's ID is composed of one Unicode character (as for uppercase 'T' in 'Ter', above), I simply key that character inline.
- If the grapheme's ID is composed of two or more characters (as for the 'ae' grapheme in 'ternæ'), I still key it inline, but wrapped in parentheses, so the script `input.py` knows how to process it.

Though the *Codex Salmasianus* has no regular graphic word distinction (no spaces between words), I am interpretively distinguishing words: in each row, the left cell has a sequence of graphemes mapped to an inflected word in the right cell. For the time being, this is a sufficiently practical and efficient way to encode the graphemes/word mapping with a simple text or spreadsheet editor.
The script `input.py` does the following (I am using here, as elsewhere in this article, the files relative to *Codex Salmasianus* as an example):

1. *Importing files.* It imports the tables of graphemes and alphabemes (`salm_table_graphemes.csv` and `salm_table_alphabemes.csv`) and the CSV transcription file (`salmasianus.csv`) of the witness.
2. *Tables of signs*. It converts the two tables of signs into a sequence of `<char>` elements as

text in a presumed 'original' linguistic form.

described above and writes them in the <charDecl> section of the <teiHeader> of files salm_graphic.xml, salm_alphabetic.xml and salm_menota.xml.

3. *Graphic transcription*. For each row of the transcription CSV file (i.e. for each word), it processes the 'left column' strings (i.e. the graphic transcription) and writes the 'all <g>' XML code described above to files salm_graphic.xml and salm_menota.xml. In the latter file, the script inserts this sequence of <g> elements in a <me:facs> element.
4. *Alphabetic transcription*. Each 'left column' string is a sequence of grapheme IDs. The script matches these IDs in table_graphemes.csv with the ID(s) of the corresponding alphabeme(s). For instance, grapheme 'ae' in table_graphemes.csv corresponds to the sequence of alphabemes 'a', 'e'. The script writes the relative <g> elements (representing alphabemes) to files salm_alphabetic.xml and salm_menota.xml. In the latter file, the script inserts this sequence of <g> elements in a <me:dipl> element.
5. *Linguistic transcription*. The script processes the 'right column' (linguistic transcription) of the CSV transcription file and writes its content to salm_linguistic.xml (with a <w> element), and to salm_menota.xml (in a <me:norm> element).

As far as the graphic and alphabetic encoding is concerned, this work-flow is very efficient. Apart from the tables of signs, what I actually key in is the graphic transcription alone, in a sort of 'dialect format' for internal use: e.g. tern(ae) for 'ternæ'. The alphabetic transcription is generated automatically.

20 Prospective work-flow

At the present stage, the 'missing link' in this work-flow is the linguistic layer. In fact, I currently populate the 'right column' of the CSV transcription file by hand. I do so simply because I do not want to leave those cells blank. As anticipated, however, I plan to develop an efficient input mechanism for the linguistic transcription (possibly by means of a web interface and JavaScript) that will run as follows:

1. The philologist keys in the graphic transcription in a dynamic HTML page in the current 'dialect format', e.g. tern(ae).
2. JavaScript generates the alphabetic transcription layer semi-automatically (the philologist approves it or rejects it).
3. A 'normalised' string resulting from the processing of the alphabetic transcription[42] is sent to a web service (possibly *Perseus*[43]) that parses and lemmatises it, thus returning the lemmatic and morphological information needed for the linguistic layer encoding. This process will also be semi-automatic, as in most cases *Perseus* will return a number of possible lemmas or morphological analyses, between which the philologist will be asked to choose. The whole process should be performed by JavaScript and other HTML5 technologies.

Undoubtedly, this envisioned work-flow will require more work on the side of the philologist than a simple 'one-layer' transcription, but I think that it may be considered a sustainable work-flow when compared to the complexity of the edition model that it would produce.

---

42 This is one of the phases in which the values of the 'visualisation' field in the table of alphabemes come of use.

43 See footnote 41.

6 Conclusion

I believe that a sophisticated, multi-layered model requiring a complete description of encoded signs is needed to meet the challenge of creating digital scholarly editions which are based on the collation of textual witnesses written with different graphic and alphabetic systems.
I hope that I shall soon be able to develop a complete and sustainable work-flow that will allow me to implement such a model in a working prototype, and to submit the results of this experiment to the attention of the Digital Humanities community.

7 Bibliography

Baroni A. (2009). *La grafematica: teorie, problemi e applicazioni*, Master's thesis, Università di Padova. URL=http://unipd.academia.edu/AntonioBaroni/Papers/455456/La_grafematica_teorie_problemi_e_applicazioni [last retrived 10.03.2013].

Barry B. (1987). *The Iudicium coci et pistoris of Vespa*, in *Filologia e forme letterarie. Studi offerti a Francesco della Corte*, vol. 4, Quattro Venti, pp. 135-149.

Baumgartner A. (1981). *Untersuchungen zur Anthologie des Codex Salmasianus*, Baden Köpfli.

Benozzo F. (2011), *Dalla filologia tradizionale all'etnofilologia tradizionante*. In D. Fiormonte, ed., *Canoni liquidi. Variazione culturale e stabilità testuale dalla Bibbia a Internet*, ScriptaWeb, pp. 28-42.

Bodard G., Garcés J. (2009). *Open Source Critical Editions: A Rationale*, in M. Deegan, K. Sutherland, edd., *Text Editing, Print, and the Digital World*, Ashgate, pp. 83-98.

Bohnenkamp A. *et al.* (2011). *Perspektiven auf Goethes Faust. Zur historisch-kritischen Hybridedition des Faust*, in A. Bohnenkamp, ed., *Jahrbuch des Freien Deutschen Hochstifts*, Wallstein, pp. 23-67.

Brüning G., Henzel K., Pravida D. (2012). *On the dual nature of written texts and its implications for the encoding of genetic manuscripts*, a talk delivered at conference *DH 2012*.
URL=http://lecture2go.uni-hamburg.de/konferenzen/-/k/13957 [last retrieved 04.03.2013].

Bücheler F., Riese A. (1894-1926). *Anthologia latina sive poesis latinae supplementum*, B.G. Teubner.

Cozzo A. (2006). *La tribù degli antichisti: un'etnografia ad opera di un suo membro*, Carocci.

Driscoll M. J. (2006). *Levels of transcription*, in L. Burnard, K. O'Brien O'Keeffe, J. Unsworth, edd., *Electronic Textual Editing*, Modern Language Association of America.

Fiormonte D. (2012a). *Towards a Cultural Critique of Digital Humanities*. «Historical Social Research / Historische Sozial Forschung», Vol. 37 (3), n. 141 (=M. Thaller, ed., Proceedings of conference *Controversies around the Digital Humanities*), pp. 59-76. URL=http://www.cceh.uni-koeln.de/files/Fiormonte_final.pdf [last retrieved 02.03.2013].

Fiormonte D. (2012b). *Testo Tempo Verità*, «Humanist Studies & the Digital Age» 2 (1).

Haugen O. E. (2004). *Parallel Views: Multi-level Encoding of Medieval Nordic Primary Sources*, «Literary and Linguistic Computing» 19 (1), pp. 73-91.

Lespect J.-F. (2005). *Vespa, le Iudicium coci et pistoris iudice Vulcano (Anthologie Latine, 199) : introduction, texte latin, traduction et notes*, «Folia Electronica Classica» 9. URL=http://bcs.fltr.ucl.ac.be/fe/09/VespaIntro.html [last retrieved 10.03.2013].

Mordenti R. (2001). *Informatica e critica dei testi*, Bulzoni.
Mordenti R. (2011). *Paradosis. A proposito del testo informatico*, Accademia Nazionale dei Lincei.
Mordenti R. (2012). *Prospettive e problemi per l'edizione critica digitale dello Zibaldone Laurenziano (Plut. XXIX, 8) di Giovanni Boccaccio*, a talk delivered at the round table *L'informatica umanistica e i suoi problemi. L'edizione critica digitale dei testi. Roma, 20 giugno 2012*, Centro Linceo Interdisciplinare "B. Segre". URL=http://www.lincei.it/files/centro_linceo/Rela_Mordenti.pdf [last retrieved 05.03.2013].
Monella P. (2012). *In the Tower of Babel: modelling primary sources of multi-testimonial textual transmissions*, a talk delivered at the *London Digital Classicist Seminars 2012*, Institute of Classical Studies, London, on 20.07.2012. URL=http://www.digitalclassicist.org/wip/wip2012.html [last retrieved 17.03.2013].
Monella P. (forthcoming). *Why are there no digital scholarly editions of "classical" texts?* forthcoming in the proceedings of The fourth meeting on Digital Philology, Verona 13-15.09.2012. Pre-print: URL=http://www.unipa.it/paolo.monella/lincei/why.html [last retrieved 10.03.2013].
Omont H. A. (1903). *Anthologie de poètes latins dite de Saumaise: Reproduction réduite du manuscrit en onciale, latin 10318, de la Bibliothèque nationale*, Berthaud frères.
Orlandi T. (1999). *Ripartiamo dai diasistemi*, in *I nuovi orizzonti della filologia. Ecdotica, critica testuale, editoria scientifica e mezzi informatici elettronici, Conv. Int. 27-29 maggio 1998*, Accademia Nazionale dei Lincei, pp. 87-101.
Orlandi T. (2010). *Informatica testuale. Teoria e prassi*, Laterza.
Pasquali G. (1971). *Storia della tradizione e critica del testo*, Felice Le Monnier.
Perri A. (2009). *Al di là della tecnologia, la scrittura. Il caso Unicode*. «Annali dell'Università degli Studi Suor Orsola Benincasa» 2, pp. 725-748.
Pierazzo E. (2011). *A rationale of digital documentary editions*. «Literary and Linguistic Computing» 26 (4), pp. 463-477.
Pini F. (1958). *Iudicium coci et pistoris*, Gismondi.
Reynolds L., Wilson N. (1991). *Scribes and Scholars: A Guide to the Transmission of Greek and Latin Literature*, Clarendon Press.
Sampson G. (1990). *Writing Systems: A Linguistic Introduction*, Stanford University Press.
Schmidt D., Colomb R. (2009). *A data structure for representing multi-version texts online*. «International Journal of Human-Computer Studies» 67 (6), pp. 497-514.
Schmidt D., Fiormonte D. (2010). *Multi-Version Documents: A Digitisation Solution For Textual Cultural Heritage Artefacts*, «Intelligenza Artificiale» 4 (1), pp. 56-61.
Shackleton Bailey D. R. (1980). *Three pieces from the Latin anthology*, «Harvard Studies in Classical Philology» 84, pp. 177-217.
Shackleton Bailey D. R. (1982). *Anthologia Latina, I : Carmina in codicibus scripta, Fasc. 1 : Libri Salmasiani aliorumque carmina*, Teubner.
Thaller M. (2006). *Strings, Texts and Meaning*, in *Digital Humanities 2006. The First ADHO International Conference*. Université Paris-Sorbonne, 5-9.07.2006. Conference Abstracts, Centre de Recherche Cultures Anglophones et Technologies de l'Information, pp. 212-214.
Wittern C. (2006). *Writing Systems and Character Representation*, in L. Burnard, K. O'Brien O'Keeffe, J. Unsworth, edd., *Electronic Textual Editing*, Modern Language Association of America.