Encoding the Critical Apparatus by Domain Specific Languages: The Case of the Hebrew Book of Qohelet

Luigi Bambaci
Department of Cultural Heritage,
University of Bologna
luigibambaci@yahoo.it

Federico Boschetti
CNR-ILC of Pisa &
VeDPH, Ca' Foscari Venezia
federico.boschetti@ilc.cnr.it

Abstract

English. Manually encoding critical apparatuses by markup languages such as TEI-XML is a non-trivial, error-prone task. It requires a technology expertise which is not within the competence of most traditional philologists and may therefore be perceived as an obstacle, instead of an aid, to their research activities. We illustrate how an approach based on Domain Specific Languages (DSLs) may simplify the creation of digital apparatuses, the data interchange, and the cooperation between the communities of traditional and digital philologists. Our case studies are represented by a sample digital edition of the biblical Hebrew book of Qohelet and by the digitalization of the collation of Hebrew manuscripts of the same book performed by Benjamin Kennicott at the end of the XVIII century. Both have been annotated through the web application based on DSLs named Euporia.

Italiano. La codifica manuale di apparati critici attraverso linguaggi quali TEI-XML è un compito complesso e soggetto ad errore. Essa richiede una preparazione tecnica che non rientra in genere tra le competenze dei filologi tradizionali e rischia pertanto di essere percepita come un ostacolo alle rispettive attività di ricerca, anziché come una risorsa. Nel presente articolo illustriamo in che modo un approccio basato su Domain Specific Languages (DSLs) sia in grado di semplificare la creazione di apparati digitali, lo scambio dei dati e la cooperazione tra le due comunità di filologi tradizionali e filologi digitali. I casi di studio sono rappresentati da un esempio di edizione critica digitale del libro biblico di Qohelet e dalla digitalizzazione della collazione di manoscritti ebraici medievali dello stesso libro eseguita da Benjamin Kennicott alla fine del XVIII secolo. Entrambi sono stati annotati attraverso Euporia, un'applicazione web basata su DSLs.

1 Introduction[1]

The work of the textual philologist consists of two main parts: 1. the gathering and the systematic analysis of all the available documents (the *witnesses*) of a literary work (*recensio*); 2. the removing of all the errors due to the textual transmission process (*emendatio*) (Timpanaro 2005). During the *recensio* phase, the scholar proceeds to a comparison of the witnesses with the purpose of detecting textual differences (the *variant readings* or simply *variants*). This procedure, named *collatio*, is one of the most important and delicate phase within the workflow of the text-critical praxis and represents a preliminary step to the preparation of a critical edition. The variants are presented in the critical apparatus, an instrument devised to show the reader the results of both the *recensio* and the *emendatio* by means of a conventional and formalized language, specific for the domain of textual philology (Domain Specific Language, cf. section 3). One of the tasks of the digital philologist consists in encoding variant readings and conjectural emendations. The encoding enables the creation of dynamic critical apparatuses: as with databases, the user can decide which data to extract and present, to combine the result of different queries, to transform the philological data into numerical format suitable for quantitative analysis and, finally, to prepare a digital version of the work. Unlike traditional, printed critical apparatuses, where the information is

[1]Even if both authors contributed equally to this work, L. Bambaci is responsible for sections 1-3 and 5-6 and F. Boschetti is responsible for section 4.

stored in a predefined, static way, a digital apparatus allows to retrieve, from the same encoded file, different types of information according to different research purposes and needs (Driscoll and Pierazzo 2016). The guidelines provided by the Text Encoding Initiative (TEI)[2] are among the best practices in the domain of the digital philology. TEI markup schemes pursue interoperability and reusability, making available for digital philologists a common interchange language covering a large set of text-critical phenomena.[3] TEI digital framework, moreover, is flexible enough to enable the user to add new tags and attributes, thus allowing to shape customized encoding vocabularies suitable for specific text-critical problems and for different literary traditions. The verbosity and complexity of XML language, however, combined with the necessity of being adherent to standards, is at risk of distracting the traditional philologist from his or her critical activity. Goal of this paper is to show how it is possible to encode variant readings by exploiting the nature of DSL which characterizes the language of the critical apparatus, without requiring from the philologist to deal with TEI technicalities and with problems of conformity to standards. Our case studies are represented by a sample digital collation of one book of the Hebrew Bible, the book of Qohelet also known as Ecclesiastes, conventionally dated to V-III BC, and by the digital version of the collation of Hebrew Medieval manuscripts of the same book, carried out by Benjamin Kennicott at the end of the XVIII century (Kennicott 1776). Both the collations are part of a forthcoming doctoral dissertation, which aims to prepare a digital critical edition of the literary work.

2 Background

As we have already discussed in Bambaci et al. 2018, 2019, many tools for encoding critical apparatuses are already available or currently developing.[4] The strategies adopted by these tools to avoid or facilitate the encoding process are mainly based on *ad hoc* graphical user interfaces or on annotation systems through abbreviated markers. Tools that allow to create digital TEI apparatuses directly from printed, traditional ones are lacking or not fully developed, such as in the case of the Classical Text Editor (Hagel 2007). The methodology we propose is implemented on Euporia, a web annotation system based on DSLs developed at the CoPhiLab of the CNR-ILC. Euporia has been formerly used for interpretative tasks, such as the identification of ritual frames in the ancient Greek tragedies (Mugelli et al. 2016), and also for educational purposes, namely in teaching Ancient Greek both in secondary school (Liceo Classico) and with BA students in Classics (first year students) at the University of Pisa.

3 Methodology

As stated in the introduction, the critical apparatus is the part of a critical edition or collation devoted to the collection of textual variants and conjectural emendations. There are no fixed guidelines for compiling a critical apparatus. The different methodologies are indeed the result of different traditions of study and research practices, which depend not only on the literary domain (a critical edition of a classical text will necessarily be different from a critical edition of a medieval text, Varvaro 1970), but also on internal developments and trends within each discipline: even different editions of the same text may vary in the choice of vocabulary or typographical standards, according to the different scholarly orientations or the editor's critical insights. Despite this extreme degree of variability, the critical apparatuses, however different the shapes they may assume, all strive for the same goal: to overcome the verbosity of the natural language and present the information in a way which is as concise as possible. The result of this process of departure from the natural language is an *artificial* (or *planned*) language (Blanke 2011, Libert 2018), specific to the domain of the textual philology (Domain Specific Language, DSL). Unlike a General Purpose Language (GPL), which is applicable across domains, a DSL is a language of limited expresiveness optimized for a particular domain of knowledge or domain of

[2] https://tei-c.org/

[3] Cf. in particular chapter 12 of the TEI Guidelines, devoted to the encoding of the critical apparatus: TEI Consortium, eds. "12 Critical Apparatus." TEI P5: Guidelines for Electronic Text Encoding and Interchange. [3.5.0.]. [16th July 2019].

TEI Consortium. https://tei-c.org/release/doc/tei-p5-doc/en/html/TC.html ([29/11/2019]).

[4] Cf. https://wiki.tei-c.org/index.php/Category:Editing_tools and https://wiki.tei-c.org/index.php/Editors for a list of the main editing tools.

application (Fowler 2010).[5]

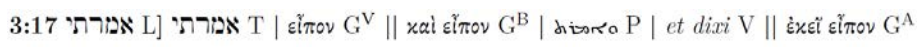


Figure 1: A sample apparatus entry from Qohelet 3:17

Let's take an example of apparatus entry from the third chapter of Qohelet, verse 17 (Fig. 1). Concepts such as "location", "lemma", "witness", "reading" and "variant reading" are encoded here by means of: 1. numbers (chapters and verses); 2. strings (the Latin *sigla* indicating the witnesses; the words of the readings); 3. characters (separators such as the square bracket "]", found after the lemma; a vertical line "|" which divides the readings and a double vertical line "||" which marks the end of a reading group). The function of the apparatus components is determined by their position within the sentence (the first reading group shows the readings supporting the lemma; the following groups contain the variants).[6] Similarly, in the apparatus entry of Qohelet 5:1 in Kennicott's collation (Fig. 2) we find integers for the

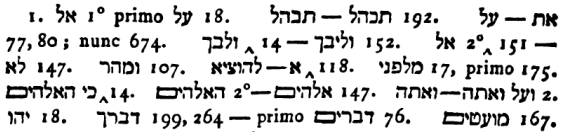


Figure 2: A sample apparatus entry of Qohelet 5:1 from Kennicott's collation

identification of the location ("verse 1"), of the manuscripts' *sigla* and of the word occurrence in the reference text (the numero sign following the roman number after the reading, e. g. "1°"); Hebrew words identifying lemmas and readings; special symbols for describing the variant typology (e. g. the symbol "^" standing for omission); annotations concerning the source description ("*primo*" for "first copyist's hand, "*nunc*" for "second copyist's hand") and finally special separators for discriminating reading groups ("—") and apparatus entries (".). A language of this sort, in which all the constituents are characterized in a concise, non-redundant and unambiguous way, is a formal language. From a computational point of view, a formal language is a language whose structure (its syntax) and meaning (its semantics) is clearly defined (Grishin 1989). A computer, therefore, is able to check that sentences are grammatically correct (well-formed) and to recognize their meaning and function (Reghizzi 2009).

4 Workflow

4.1 The Context Free Grammar

In order to allow the interpreter (or compiler) to parse the apparatus written in our DSL, we wrote the Context Free Grammar (CFG). A CFG is a formal grammar consisting of a set of rules describing a formal language (Parr 2010). An example of CFG suitable for analysing the lemma is shown in Fig. 3. Thanks to the parsing system provided by ANTLR software (Parr 2012), the CFG allows to tokenize and parse the whole apparatus entry, identifying the vocabulary symbols (token rules) and the syntactic structure (parser rules). The token rules (in the example, the rules in capital letters) allow to tokenize integers (`NUM`), alphabetic characters (`ALPHA_SEQ`), Hebrew words (`HEBW`) and separators (`R_BRACKET`). The parser rules allow to check the syntax: the lemma (`lem`), for example, is encoded as a sequence of Hebrew words (`w+`), witnesses sigla (`wit`) and separators (`LemSep`). The result of the parsing of the whole apparatus entry is an Abstract Syntax Tree (AST), as shown in Fig. 4 below. The AST attaches

[5]Examples of DSL can be considered the language of algebra for stating numerical relationships, the language of boolean logic for propositional calculus and, more in general, any kind of notation systems that allows, within a particular community of practice, the description of problems and solutions in a specific area of interest. In computer science, examples of DSLs are HTML for web pages, SQL for relational databases, LaTeX for text processing, XSLT for XML transformations and so forth. In software engineering, these languages can be called, more properly, Domain Specific Programming Language (DSPL), as opposed to General Purpose Programming Language (GPPL), such as Java, C, etc.

[6]Both vocabulary and structure of the critical apparatus of the sample edition have been shaped following the more recent critical edition of the book (Goldman 2004).

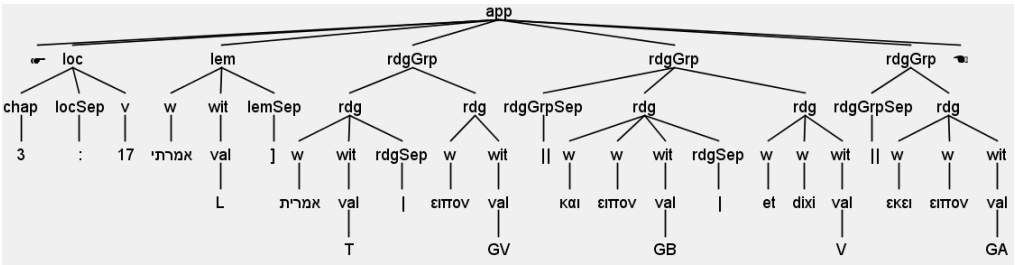
```
grammar QoheletEuporia;

app : loc lem;
lem : w+ wit lemSep;
loc : chap + locSep + v?;
chap : NUM;
v : NUM;
locSep : DOUBLE_POINT;
lemSep : R_BRACKET;
wit : ALPHA_SEQ;
w : HEBW ;

NUM : [0-9]+('.'[0-9]+)?;
ALPHA_SEQ : [a-zA-Z]+;
DOUBLE_POINT : ':';
R_BRACKET : ']';
HEBW : [\u0590-\u05ff]+;
```

(a) CFG for the sample edition of Qohelet

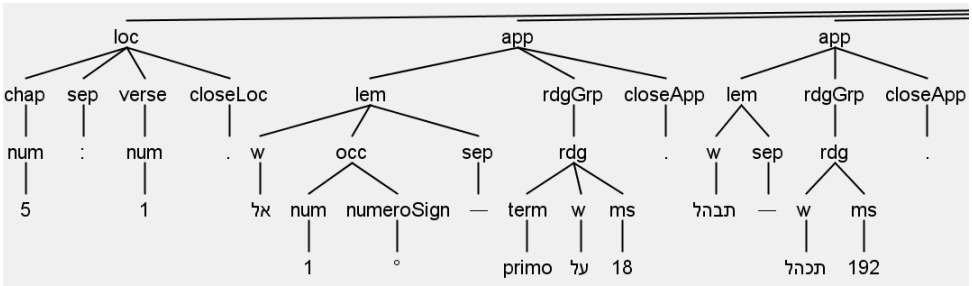
```
grammar kennicott;
all: listApp+;
listApp: loc app+ closeMainApp;
app: lem  rdgGrp+ closeApp | note? closeApp;
lem: (w+ occ? (missWord w+)? (sep)?) | (COMM NUM? (CONJ NUM)?);
rdgGrp: rdg+ (sep?);
rdg: ((w+)? (missWord w+)? (term+)? (w+)?) (w+|term+) ms+ note?;

w: HEBW rasura?;
missWord: MISSING+ (BRACKET_OP NUM BRACKET_CL MISSING+)?;
loc: chap sep verse closeLoc;
endVs: NUM END;
term: MAN_DESC;
ms: NUM ALPHA_SEQ? commaSep?| ALPHA_SEQ+ commaSep? ;
closeApp: END;
closeLoc: END;
```

(b) CFG for Kennicott's collation

Figure 3: Examples of Context Free Grammars



(a) AST from the sample edition



(b) AST from Kennicott's collation

Figure 4: Examples of syntactic trees (AST)

to the textual items (the nodes) a label which remind the function assumed in the context and shows the syntactic hierarchical relationships existing between them (the branches).

4.2 The Visitor

In order to convert any DSL in XML, we wrote a software component, named "AstToXmlVisitor", which generates an XML file structured on the AST. The Visitor passes through the tree nodes and slavishly translates the parser rules into XML markers (Fig. 5). The result is a structured, well-formed XML file, whose elements take the name from the parser rules and the hierarchical structure from the AST. The Visitor has been implemented in Java language, through the set of tools available in ANTLR4 software.

```
<app>
  <loc>
    <chap>3</chap>
    <locSep>׃</locSep>
    <v>17</v>
  </loc>
  <lem>
    <w>אמרתי</w>
    <wit>L</wit>
    <lemSep>]</lemSep>
  </lem>
  <rdgGrp>
    <rdg>
      <w>אמרתי</w>
      <wit>T</wit>
      <rdgSep>|</rdgSep>
    </rdg>
    <rdg>
      <w>εἶπον</w>
      <wit>GV</wit>
      <rdgSep>|</rdgSep>
    </rdg>
  <rdgGrp>
...
```

(a) XML output of Qoh. 3:17 from the sample edition

```
<listApp>
   <loc>
      <chap>
         <num>5</num>
      </chap>
      <sep>׃</sep>
      <verse>
         <num>1</num>
      </verse>
      <closeLoc>.</closeLoc>
   </loc>
   <app>
      <lem>
         <w>אל</w>
         <occ>
            <num>1</num>
            <numeroSign>˚</numeroSign>
            <sep>—</sep>
         </occ>
      </lem>
      <rdgGrp>
         <rdg>
            <term>primo</term>
            <w>על</w>
            <ms>18</ms>
         </rdg>
      </rdgGrp>
      <closeApp>.</closeApp>
   </app>...
</listApp>
```

(b) XML output of Qoh. 5:1 from Kennicott

Figure 5: Visitor's XML outputs

4.3 From XML to TEI-XML

A final conversion from XML code to a TEI compliant critical apparatus has been carried out through an XSLT stylesheet (Fig. 6). During the transformation phase from XML to TEI-XML, the philologist can choose which elements represent in the encoding and which to rule out (punctuation, separators and so forth). The encoding of both the apparatuses rely on the TEI model of critical apparatus: the apparatus of the digital edition of Qohelet has been encoded with the parallel segmentation method, while the encoding of Kennicott's collation follows the location-referenced method.

5 Results

So far, the first three out of twelve chapters of Qohelet have been collated. Kennicott's collation has been totally digitalized and automatically encoded. The critical apparatus of both the collations hosted in Euporia have been successfully converted into a compliant TEI file. Using XSLT stylesheets, it is possible to re-convert the TEI file back to our DSL, without loss of information. TEI encoding schemes and our DSL are therefore isomorphic. The implementation of AstToXmlVisitor in JavaScript language represents an important point of the work-flow. It allows to automatically create a well-formed XML file from any AST and is therefore applicable to different DSLs. It is written once and for all by the computer scientist and needs no further customization according to the input file. Such a division of tasks meets the needs and habits of digital philologists, who are generally more accustomed to manipulating XML code, rather than working with general-purpose programming languages.

```xml
<div type="chap" n="3">
  <ab n="17">
    <app>
      <lem wit="#L">אמרתי</lem>
      <rdgGrp>
          <rdg wit="#T">אמרית</rdg>
          <rdg wit="#GV">εἶπον</rdg>
      </rdgGrp>
      <rdgGrp>
          <rdg wit="#GB">καὶ εἶπον</rdg>
          <rdg wit="#P">ܐܡܪ̈ܬ</rdg>
          <rdg wit="#V">et dixi</rdg>
      </rdgGrp>
       <rdgGrp>
          <rdg wit="#GA">ἐκεῖ εἶπον</rdg>
      </rdgGrp>
    </app>
  </ab>
</div>
```

(a) TEI apparatus of Qoh. 3:17

```xml
<listApp>
   <app loc="5 1">
      <lem>
         <w>אל</w>
         <num>1</num>
         <pc>˚</pc>
      </lem>
      <rdgGrp>
         <rdg wit="#K18">
            <term>primo</term>
            <w>על</w>
         </rdg>
      </rdgGrp>
   </app>
   <app loc="5 1">
      <lem>
         <w>תבהל</w>
      </lem>
      <rdgGrp>
         <rdg wit="#K192">
            <w>תכהל</w>
         </rdg>
      </rdgGrp>
   </app> ...
</listApp>
```

(b) TEI apparatus of Qoh. 5:1

Figure 6: TEI compliant XML encoding

6 Discussion

There are several advantages in using a DSL for ecdotic purposes. First of all, the compactness of the DSL respect to TEI encoding. The annotation through a DSL is significantly less verbose than TEI annotation, as it can be seen by comparing the number of characters of the traditional apparatus shown in Fig. 1 and the TEI counterpart of Fig. 4 (on the right). Compactness is an important feature: the verbosity of XML language may compromise human readability and make the encoding difficult to handle, especially for traditional scholars not accustomed with long in-line encoded files. Manually encoding is a time-consuming task. Markup vocabularies require a long apprentice time to be mastered; once the encoding is complete, moreover, the encoder must check whether it is internally coherent, perfectly TEI conformant and in line with best practices. The cognitive stress derived from such a mixture of disciplinary content and cross-disciplinary formalism may stray away the world of humanistic academic research from the potentialities of computational technologies, thus contributing to increase the gap between the respective communities of scholars. A DSL-based approach, on the contrary, is entirely domain-centered: the scholar is not compelled to acquire skills which fall outside his or her cultural background, nor to make his or her research practices adhere to external, technology-conditioned standards. It is up to the digital philologist, who best knows how to organize the data according to standards, to create a perfectly conformant TEI encoding from the results of the XML general exporter. Finally, the DSL may represent a good way to exercise tighter control not only on transcriptional errors, but also on semantic errors. Thanks to the tokenization, indeed, the parser is able to assign a semantic value to each apparatus component directly from the data type to which it belongs: so, for example, tokens such as “omit” (abridgement for Latin “omittit”, non-capital character), “K1” (capital alphabetic character + integer), “McNeile(1904)” (string of alphabetic characters, integers and punctuation), will always be parsed differently and automatically assigned to different XML tags or attributes (In TEI encoding, respectively, `@ana`, `@wit`, `@resp`). In a manual encoding, on the other hand, the encoder must decide, each time, which tags or attributes are more suitable for expressing his or her interpretations of textual phenomena: this may often lead to an incoherent or erroneous choice of markers and increase the possibility of semantic errors, which are very difficult to be detected. In a DSL-based approach, on the contrary, the choice of markers is not entrusted to human decision, but it is determined by the form of the apparatus components and automatically performed by the Visitor.

References

Luigi Bambaci, Federico Boschetti, and Riccardo Del Gratta. 2018. Qohelet Euporia — A Domain Specific Language to Annotate Multilingual Variant Readings. In *Proceedings of the 5th International Congress on Information Science and Technology, Marrakech, Morocco, October 21-27, 2018*. Piscataway, NJ, pages 266–69.

Luigi Bambaci, Federico Boschetti, and Riccardo Del Gratta. 2019. Qohelet Euporia — A Domain Specific Language for the Encoding of the Critical Apparatus. *International Journal of Information Science and Technology* 3(5):26–37.

Detlev Blanke. 2011. Planned Languages — a Survey of some of the main Problems. In *Interlinguistics: Aspects of the Science of Planned Languages*, Walter de Gruyter, pages 63–87.

Matthew James Driscoll and Elena Pierazzo. 2016. *Digital Scholarly Editing: Theories and Practices*. Open Book Publishers, Cambridge.

Martin Fowler. 2010. *Domain-Specific Languages*. Pearson Education.

Yohanan A. P. Goldman. 2004. Qohelet. In *Biblia Hebraica Quinta: Megilloth: Ruth, Canticles, Qoheleth, Lamentations, Esther*, Deutsche Bibelgesellschaft, Stuttgart.

V. N. Grishin. 1989. Formalized language. In *Encyclopaedia of Mathematics*, Kluwer Academic Publishers, volume 4, pages 61–62.

Stefan Hagel. 2007. The Classical Text Editor. An attempt to provide for both printed and digital editions. In A. Ciula and F. Stella, editors, *Digital philology and medieval texts*, Pacini, University of Michigan, pages 77–84.

B. Kennicott. 1776. *Vetus Testamentum Hebraicum cum variis lectionibus*, volume 2. Clarendon, Oxford.

Alan Reed Libert. 2018. *Artificial Languages*. Oxford University Press.

Gloria Mugelli, Federico Boschetti, Riccardo Del Gratta, Angelo Mario Del Grosso, and Fahad Khan. 2016. A user-centred design to annotate ritual facts in ancient Greek tragedies. *BICS* 59(2):103–120.

Terence Parr. 2010. *Language Implementation Patterns: Create Your Own Domain-specific and General Programming Languages*. Pragmatic Bookshelf.

Terence Parr. 2012. *The Definitive ANTLR 4 Reference*. Pragmatic Bookshelf.

Stefano Crespi Reghizzi. 2009. *Formal Languages and Compilation*. Springer.

Sebastiano Timpanaro. 2005. *The Genesis of Lachmann's Method*. University of Chicago Press, Chicago / London.

Alberto Varvaro. 1970. *Critica dei Testi Classica e Romanza — Problemi Comuni ed Esperienze Diverse*. L'Arte Tipografica, Napoli.