9.

Making digital scholarly editions based on Domain Specific Languages

Simone Zenzaro, Federico Boschetti and Angelo Mario Del Grosso

Introduction

Over time textual scholars have refined the methods to represent the codicological, palaeographic, philological and other aspects relevant for the study of documents (that is, material objects) and texts (that is, immaterial entities). According to a general trend observable across the last four centuries not only in the STEM disciplines but in every domain of knowledge, the specific languages adopted by the scholars to represent the objects of their studies evolved, improving in both precision and concision (Bizzoni et al., 2020). It suffices to compare critical apparatuses sampled in a wide temporal span for a quick verification. Indeed, it is surprising that in the digital age the collective effort of the scholars to optimise the representation and the transmission of their domain-specific knowledge has been penalised and verbose solutions (for example, through XML encoding) or, on the contrary, nonverbal solutions (for example, through Graphic User Interfaces (GUIs)) have been adopted.

The classical scholarly practices represent a valuable synthesis of centuries of knowledge in specific domains, so it is paramount to preserve such standards.

Another relevant aspect is the ability to endow the scholars with a methodology that retains and expands all the expressiveness needed to deal with the text challenges. The digital counterpart has also produced and established standards.

The methodology based on Domain Specific Languages, shortened to DSLs (Zenzaro et al. 2022), requires the definition of a formal language derived from the well-established ecdotic practices that are already a set of editorial conventions and convey the analytical representation of the information in the text. For example, critical apparatuses are already a quasi-formal domain language and are therefore suitable for the definition of a DSL via a context-free grammar.

The next step is to feed a rich text-editing tool with the DSL in order to enable the corresponding language interpretation. The result is to provide scholars with a re-usable and modular computer-assisted environment that eases the creation and analysis of the scholarly edition. At the same time, computational functionalities empower the process with multimodal search, classification and prediction strategies of philological phenomena, consistent and systematic coherent checks of the editorial conventions and errors, analysis and recall of information deduced from the context or from external sources (for example, vocabularies and corpora) via machine learning algorithms, and so on.

Moreover, a fully collaborative environment allows scholars to contribute to an ongoing cooperative edition. In this context it is possible to widen the access to the text to scholars, students, practitioners and volunteers.

Finally, this approach ensures the compatibility with the standards accepting towards and producing from the DSL a compliant representation of the edited text that can interoperate with the digital humanities community and the galaxy of related tools.

The DSL-based methodology is well known and exploited mostly outside the scholarly editing domain. Being a formal language, a

DSL has its roots in the language theory and the first attempts saw the effort to use them to describe natural languages. That path has been proved to be infeasible due to the ambiguity of natural languages but this is not the case with the philological domain. The markdown language is an example of a commonly used DSL, but its scope is a general-purpose description of the structure of a document. Thus it is not meant to describe philological textual phenomena. Leiden,[1] instead, is a good example of the application of a DSL in the domain of traditional papyrology conventions.

Adopting a DSL for the scholarly editing process allows the philologist to remain close to the classical practices while enabling the possibility to improve the process with the digital capabilities. The only constraint enforced by this approach is the ambiguity elimination.

A challenging fourth revolution

After the passage from orality to written texts, from scrolls to codices, from manuscripts to printed books, the fourth revolution from Gutenberg to digital editions is under way (Roncaglia 2010).

Any changes of material support expose the documents produced in the previous epoch to the risk of oblivion, damage and loss. To avoid this risk, the evaluation of priorities and the cost-benefit assessment have been necessary. Thus the first collections of digital texts, such as the Thesaurus Linguae Graecae (TLG),[2] were based on canonical editions deprived of the critical apparatus, whereas the recent massive campaigns of digitisation gather the page images of a million books from the libraries all around the world, without axiological criteria.

Different outcomes are possible from the facsimile: the extraction of the plain text by Optical Character Recognition (OCR) applied

1 Cfr https://papyri.info/docs/leiden_plus.

2 http://stephanus.tlg.uci.edu.

to printed editions, or by Handwritten Text Recognition (HTR) applied to manuscripts for textual retrieval purposes, possibly re-mapped to the digital image, or the creation of digital scholarly editions (DSEs), which accurately annotate codicological, palaeographic and philological phenomena (Robinson 2013).

The Text Encoding Initiative (TEI)[3] provides guidelines (TEI Consortium 2022) internationally accepted as the *de facto* standard by the community of digital humanists (Schreibman, Siemens and Unsworth 2016), in order to grant the interchange of FAIR[4] data and a mild level of interoperability (Dumouchel et al. 2020). But the representation of the document (data and metadata related to the book, the layout, the script) and the encoding of the text (with or without abbreviations and normalisations) are strictly related to the use that a scientific community intends to make with a collection of DSEs (Pierazzo 2015). The answer to the question: what do you do with a million (digital) books? (Crane 2006) highly conditions the representation of knowledge, which must take into account not only textual facts (such as variant readings) but also scholarly interpretations (such as intertextual allusions or multiple levels of thematic, linguistic and stylistic analyses).

The digital representation of an artefact is optimal only when the operations that can be applied to it are clearly defined (Shillingsburg 2015). For example, the operations that can be applied to the images, such as scaling, rotating, tuning brightness and contrast, and many others, are available in most applications or web API which deal with images, and the file formats that represent the images are optimised for these operations. Surprisingly, the TEI provides guidelines for digital representation of text without defining the operations to deal with it, which are much more complex for scholarly editions than for ordinary documents managed by a word processor. Scholars need to compare multiple texts, align them at different levels of granularity

3 https://tei-c.org/.

4 Findability, Accessibility, Interoperability and Re-use https://www.go-fair.org/fair-principles/.

(character by character, word by word, block by block), associate to each textual unit multiple linguistic analyses and order textual units according to multiple criteria.

Humanists across the centuries before the digital age have optimised the representation of textual phenomena by conveying the maximal amount of information relevant in the domain of textual studies in the minimal amount of characters: a critical apparatus is by far more concise and readable for a domain expert than the equivalent apparatus encoded in XML/TEI.

The Leiden+ system demonstrates that the scientific community starts acquiring awareness about the necessity to join conciseness, familiarity, and human readability with machine actionability and interoperability. The introduction of DSL in the realm of Digital Humanities, and in particular of textual studies, is oriented in this direction.

Methodology

The goal of a methodology based on DSLs, as mentioned in the previous section, is to provide the scholarly editors with a familiar and rich environment empowering the editing process while, at same time, retaining the long-standing and well-established textual scholarship good practices (Boschetti and Del Grosso 2020).

Approaching the text by applying this methodology is a process made of three steps:

1. define one or more DSLs with the active participation of the domain experts (textual scholars/editors);
2. feed a rich text editing tool exploiting the underlying DSL;
3. integrate the tool in a collaborative (many participants for a unitary task) and cooperative (many participants for many subtasks) environment.

The resulting editing environment will be endowed with a set of properties that we consider to be not only beneficial but also empowering to the text editing process.

The first consequence of this approach is the ability to retain the expressiveness that the classical textual scholarship practices have already refined over time in their abiding domain-specific knowledge and, in doing so, to implement generic tools and specific languages (Voelter 2014). This will ensure that all the text challenges faced during the construction of the edition can be overcome, since they have already been addressed and encoded in such practices. An example taken from the papyrological domain presents the need to define a formalism to address the presence of different superimposed layers of text. The common practice in this domain is to mark the text of a superimposed layer with a superscript number (for example, $\nu\tau\alpha^{+1}$). This means that the DSL must encode this phenomenon in order to give the editors the possibility to write it as closely as possible to their usual way as well as recognise it automatically and implicitly (from the editor point of view). This also means that a software environment that implements such a methodology should and must be realised as compositional modular components (Boschetti and Del Grosso 2015; Del Grosso, Giovannetti and Marchi 2017). In other words, the single parts of the model must strive to be self-contained, replaceable and re-usable in order to maximise the modularity of the whole system.

In addition, attaching a well-defined set of operations to the text smooths the editing process and favours the analysis of the text by both the editors and the future readers. Examples of such operations are multimodal search, classification and prediction strategies, consistent and systematic coherence checks of the editorial conventions and errors, analysis and recall of information deduced from the context or from external sources (for example, vocabularies and corpora).

The collaborative and cooperative nature of such an environment creates the opportunity to widen the access to the text for scholars,

students, practitioners and volunteers by lowering the barrier to entry, and by allowing users to work remotely and in a networked way. As a consequence of editing the DSL-encoded text, the edition can be seen as an ongoing process that refines the text dynamically as a collective effort (Bordalejo and Robinson 2015).

Treating the text as a software code written in the formal language defined by a DSL implies that it is possible to derive a machine-produced interpretation of the text as an Abstract Syntax Tree (AST) that represents the structure and the relations of and between the textual phenomena (Parr 2014). The AST representation is suitable, for example, to generate a version of the text that is compatible and interoperates with the already available standards for DSEs (for example, TEI/XML). This way the DSL-based methodology complements and enhances the state of the art tools in the Digital Humanities (Boschetti et al. 2023).

*

Although adopting a DSL-based DSE approach brings several advantages both to the editing process and to the actual final edition, there are two major constraints to the application of this methodology. One is technical, the other is about interpersonal relations.

The first requirement regards the necessity to have a full disambiguation of the textual scholarship practices. It happens that such practices use the same visual clue to represent different phenomena in the same context. For example, the sublinear dot below a letter (the visual clue), for the Herculaneum papyrology, has the meaning of an *uncertain* or *illegible letter*, depending on the context. When this case occurs, it poses a problem to the automatic recognition of the phenomenon by a machine that, instead, requires a unique representation for each phenomenon to be able to correctly parse the information. This constraint is linked to the nature of a DSL: as a formal language, each text phenomenon must be described by a formal grammar, and in particular by a context-free grammar. Nevertheless, in our experience,

failing to map the textual scholarship practices to a DSL is rare and, even in such unfortunate cases, it is often possible to divert slightly from the specific language adopted by the scholars to find a close alternative that grants an unambiguous formal grammar.

The second requirement takes into account the necessity to establish a tight, respectful and frequent communication between the domain experts (usually the scholars) and the more technical figure (a computer scientist or, preferably, a digital humanist). This kind of communication is paramount to understand the domain peculiarities and to translate them into an effective DSL. The aim of this requirement is to bridge the gap between the descriptions of the phenomena in the text and the computational tools that will manage them in the digital environment. This process needs to be completed in an iterative fashion until a satisfactory definition of the domain is reached, and must be repeated for each single domain (although each DSL definition can be re-used or extended as needed). The definition of a correct DSL is aided by the application of the Domain-Driven Design (DDD) principles and by the specification of suitable Abstractions.

A toolkit for the DSL-based methodology

Abstract data types (ADTs) are the theoretical foundation of the DSL-based methodology. Proposed by Barbara Liskov and Stephen N. Zilles, an ADT is a useful mathematical model that can be defined as a 'class of objects whose logical behavior is defined by a set of values and a set of operations', that are independent from the actual implementation (Liskov and Zilles 1974).

In the context of DSEs, ADTs allow the DSL-based methodology to remain focused on both data definition and the related operations.

With data we refer to all the information needed to describe text phenomena and, in such regard, we want to underline that data is highly dependent on the domain of application. For example, what an apparatus entry is and which information needs to be represented is

highly dependent on the domain. A real-world instance of this example can be found in the context of papyrological editions, where this is particularly true since, often, there are two kinds of apparatuses: one for the diplomatic transcription and one for the literary text. Each text is then enriched by its own (diplomatic or philological) apparatus that follows different editorial rules for their entries. For this reason, we have chosen to define the different philological data as different DSLs.

But data isn't enough. Operations on data play an important role in crafting an edition and browsing its content. Therefore we propose a set of core operations inherently connected with textual scholarly data: (1) *edit* the textual data cooperatively, (2) *store* the edited text via standard formats such as XML/TEI, (3) *search* for textual phenomena considering different scholarly perspectives (philological, linguistic, historical and so on), (4) *define relations* between textual units such as between tokens and named entities, (5) *check* and *validate* the text against supplied editorial conventions, (6) *CRUD* (Create, Read, Update, Delete) operations defined for the different textual objects, (7) *align* different versions or witnesses of the same text, (8) *serialise* the encoded text in different file formats, (9) ensure *identity* and equality operations for text collation, (10) *cite* and *reference* textual passages at different granularities such as sentence or word, (11) produce a scholarly *mise en page* via PDF file format, and(12) *comment* and *annotate* custom selections of the text.

This is an effort to make the methodology framework modular, namely a set of composable or interchangeable and re-usable components that concur together to cover the needs of the digital scholarly edition. This set is not exhaustive but consists of a solid and usually DSL-agnostic foundation to start using the data. Of course, when the domain or context of application requires more specific operations, this set should be extended. And, if some of the operations are superfluous the set may be shrunk. Since the methodology nurtures the textual scholarship practices, it is paramount to adopt a framework that, on one hand, promotes understanding the target domain, involving the philological experts in the whole development process, and, on the other hand, that ensures data and

behaviour abstractions to be defined by means of a shared language. In our case this framework is the DDD that strives to formally model domain concepts within nonambiguous semantic contexts. For instance, the representation of a lacuna in a papyrus (domain problem) is modelled as a domain concept ('lacuna') defined in nonambiguous semantic context ('lacuna in any papyrus' vs 'lacuna in any manuscript') on which suitable operations can be formally defined (for example, supply the lacuna).

Within this perspective, the DSL approach allows us to express the domain model (data and operations, hence the ADT), by adopting formal languages familiar to textual scholars. Borrowing the idiomatic term from the DDD framework, the shared language is called 'ubiquitous language'.

ADT, DSL and DDD are all the foundations we need to put in practice the methodology that finds its concrete realisation within a collaborative and cooperative editing platform.

In the following sections we will describe each of these aspects and then we will present a few examples of how to use, in practice, the DSL-based methodology for textual scholarship.

Domain Specific Languages

A DSL is a formal language that is specialised for a particular domain of applications (Parr 2007). A context-free grammar is a formalism that has been defined by the linguist Noam Chomsky, initially for the characterisation of the structure of sentences and words in natural languages. Later on, context-free grammars have been widely adopted for the definition of programming languages in computer science and formal languages in general. DSLs together with General Purpose Languages (GPLs) belong to the larger family of computer languages, and context-free grammars play a primary role in the definition of the syntactic structure of a language and its machine actionability. We will refer to the text written in a DSL

language as encoded text. A grammar, from this point of view, is a set of productions (or rules)

$$A \rightarrow \alpha$$

where A is a nonterminal symbol denoting some grammatical structure and α is a string representing the result of the application of such production. So, for example, the productions

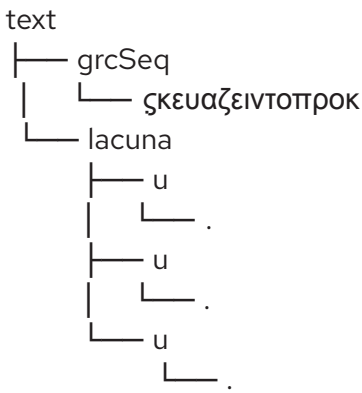
```
[parser]
lacuna → L_BRA (u | opt)+ R_BRA          // textual lacuna
opt    → L_SML_PAR u R_SML_PAR           // optional uncertain or
                                            missing character
u      → DOT | GS_DOT                    // uncertain character
                                            or missing character
grcSeq→ GRC_CHAR+                        // sequence of Greek
                                            characters
Text   → (grcSeq | lacuna)+;             // text definition

[lexer]
L_BRA  →'['                              // open in lit. ed. lacuna
                                            integrated by editors
R_BRA →']'                               // close in lit. ed. lacuna
                                            integrated by editors
L_SML_PAR → '('                          // open optional char
R_SML_PAR→ ')'                           // close optional char
DOT →('.'|'.'|'.')                       // unreadable or uncer-
                                            tain char
GS_DOT → '\ue5ce'                        // dot rendered by the
                                            specific font
GRC_CHAR → [\u0370-\u03ff\u1f00-\u1fff\u2019] // Greek char-
                                                acters
```

define a language that can recognise the lacunae and interpret them through a computer program, for example, the following DSL encoded text

represents the actual text a scholar must write to get the digital world functional enhancements in a DSL-based text editor. The text, in this case, adheres exactly to the way the domain experts use to write, but, at the same time, it is processable by a machine.

From this example, the derived AST is as follows, that shows the syntactic structure of the excerpt as understood by the machine.



As it is possible to verify from the example above, a grammar is composed of two sets of rules: one for the lexer and one for the parser. The lexer is in charge of recognising the terminal characters while the parser holds the rules for the syntactic structure. This separation is a type of modularity that improves the re-use of already defined grammars. For instance, if the concept of lacuna is captured by a set of characters inside a pair of square brackets (just like the example) in the papyrology domain, it is possible to adapt the lexer or the parser accordingly to another domain. If the editorial conventions for this other domain state that the lacunae must be surrounded by curly brackets, it suffices to change the L_BRA and R_BRA productions to the opened and closed curly brackets characters. On the contrary, if the other domain uses the same symbols (the square brackets) but with another syntactic structure, it is the parser rules that need to be modified while re-using the lexer part.

In the field of DSEs, DSLs can be successfully used to describe most – and usually all – the textual phenomena. Applying DSLs to the textual tradition creates a win-win condition that is beneficial both to the editors (philologists, papyrologist, epigraphist, etc.) and the digital exploitation of the text. Once the DSL(s) is defined, there is no need to force the scholars to change their usual approach to the text since the process to edit text will remain (mostly) the same or, at most, slightly deviate from their well-known and established practices (Mugelli et al. 2016). This differs from the currently proposed alternatives that instead require a preliminary training for the scholar that needs to learn and understand some technical jargon that appears to be far from the text itself (see the TEI/XML approach for example). When applied, the DSL approach enables all the enhancements that the digital world can already and will bring with zero or minimal cognitive effort for the domain expert (Bucchiarone et al. 2021).

Although the DSL-based approach differs in practice from TEI/XML based approach – the de facto standard for DSEs – and the GUI approach (the other most-known approach), it is not meant as a replacement for it, conversely it complements and embraces the others.

As an example, the AST can be translated to TEI/XML by transforming the XML representation of the AST:

AST to XML	**XML to TEI/XML**
<text> <grcSeq>ϛκευαζειντοπροκ</grcSeq> <lacuna> <u>.</u> <u>.</u> <u>.</u> </lacuna> </text>	<ab> <seg type="grc-seq">ϛκευαζειντοπροκ</seg> <gap reason="illegible"quanity="3"unit="character" /> </ab>

The history of DSLs has been twofold. On one hand, their wide adoption in computer science has established their usefulness and solidity. On the other hand, DSL adoption to the natural languages did not find a complete success due to their intrinsic ambiguity.

Fortunately, this latter is not the case for the DSEs. The textual practices for a scholarly edition are already DSL and the vast majority of such languages are already formal enough to be described by context-free grammars. This consideration makes the DSL-based methodology sound and applicable. And even in the occasional presence of ambiguous editorial conventions, it is often possible to modify the language slightly to disambiguate it.

Indeed, the only real constraint to the application of a DSL-based methodology to DSEs is the successful disambiguation of the domain language towards a shared and ubiquitous language.

DDD

In order to design and implement a DSL-based DSE, we follow the principles and patterns of the DDD: a software design approach introduced by Eric Evans in 2003 which fosters collaboration within a multidisciplinary context (Evans 2003; Evans 2014).

DDD focuses on the description of the problem space (the domain) and on the corresponding definition of formal models by using the proper traditional language adopted by the domain experts. This common language is called *ubiquitous language* (Millett and Nick 2015).

Among the different artefacts that DDD suggests, the ubiquitous language eases the development of the common and rigorous DSL used to build the DSE core features, which is mainly (already) defined by the domain experts. These DSLs become the formal sources and the vocabulary used also to define the domain models and the software implementation.

Thanks to this method, digital textual scholars, unawares, define their own data and operations abstracting from the details regarding both the factual data structures and the computational algorithms actually implemented in the system. Therefore, we use a DSL to capture the concepts of the domain of interest. The aim is to obtain re-usable Domain Specific Abstract Data Types, which will provide the basic composable bricks of the computational framework for the digital scholarly editing environment.

DDD provides a sound and well-established design process to delve into domain specific modelling that offers, contextually, a comprehensive perspective in regard to the domain of interest.

By adopting the DDD approach, we start the modelling activities with the definition of the problem space in the domain, then we break it down into smaller components (called sub-domains) and progressively refine the ongoing formal models and DSLs.

In particular, DDD is a specific domain modelling process able to manage different views on high-level and low-level technical and conceptual perspectives. This way, together with the experts, we are able to identify the main capabilities of the field being modelled and strive to design coherent domain-specific solutions: the bounded contexts.

Sticking to this process, we believe that the different digital components needed to profitably meet the requirements of the textual scholarship domain can be powerfully designed.

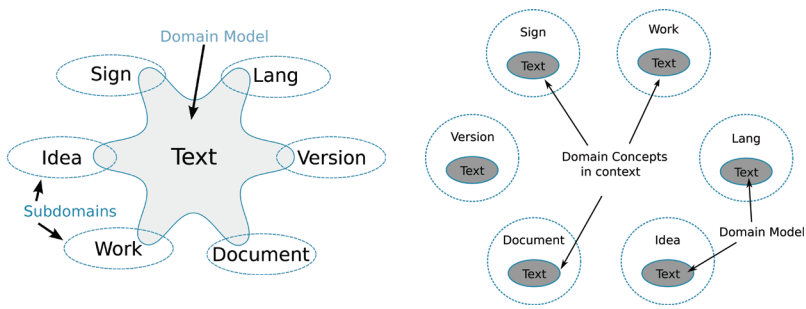
Specifically, the definition of the DSE bounded contexts provide well-designed abstractions of the domain of textual scholarship, which guarantee at the same time a high degree of decoupling among the different components (the ability to be prepared for changes via self-contained modules), as well as the definition of nonambiguous concepts among different models that can co-exist in the system.

For example, within a DSE, the concept concerning the 'uncertain' character may have different meanings, based on the different types of the edition, namely (1) diplomatic edition and (2) philological edition. The first meaning refers to a character difficult to read or even missing; the second meaning refers to a lacuna. Each meaning lives within its own bounded contexts described by the ubiquitous languages. As a result, the concept defined within the DSL is not ambiguous and can be linked to specific digital operations and computational services.

(1) u → DOT // uncertain character or missing character
(2) m → DOT // lacuna

Each bounded context consists of a core model which defines one, and only one, meaning of a shared concept. Furthermore, each bounded context defines domain specific components borrowing domain operations and domain services. It is then natural to use the microservices architecture to deploy the DSL-based DSE environment. In such a way, components are also independent of each other, ensuring the modularity and the re-usability features we require in the DSL-based DSE method.

Finally, adopting the DDD approach means that the edited text can be modelled under different and independent but interrelated perspectives (see Figures 9.1 and 9.1a).



Figures 9.1 and 9.1a Bounded Context for different text models in the Domain Driven Design. Source: Authors.

Core operations

The definition of the grammar of a DSL is a crucial step for the methodology, but that is not enough to provide a fully functioning environment that manages the text of the edition. In this section we will not address the fine-grained operations on the text (for example, adding or removing characters) but we will give an overview of a wider range of operations on the text. Following the microservice architectural pattern, a language service inspired by the wel-known Language Server Protocol[5] is in charge of the interpretation of the parsed text written in a specified DSL.

The language service implements a RESTful API (Application Programming Interface) that provides access to the language information and functionalities and models the part of the operations on the text (Fielding 2000). In particular the API defines the following end points:

- /info: the set of information that defines the language managed by the server such as the language identity, its name, the capabilities implemented for the language;
- /errors: the set of syntactic or semantic errors inferred by a given text (for example, that list of discrepancies between the text and the editorial conventions);
- /suggestions: the set of suggestions for completing a given text in a context (e.g. the position of the cursor);
- /highlighter: a data structure that defines the set of rules for highlighting significant portions of the DSL text (for example, the witnesses' names or the verse number);
- /xml: the XML representation of the plain text interpreted by the DSL definition, possibly with a given schema (for example, TEI/XML).

These operations refer to the functionalities strictly tied to the DSL syntax and semantics. Different DSLs can provide other

5 https://microsoft.github.io/language-server-protocol/.

functionalities, for example, the 'to PDF' function that produces a PDF file from a text written in a specified DSL. This kind of operation also realises the critical separation between data representation and data presentation that is often overlooked by scholars since the two are usually mixed together or simply implicitly defined.

Another important operation for a DSE is the ability to search data. The DSL-based methodology includes search capabilities in a modular fashion just like every other aspect of the methodology. There is no one for all search capability, instead different types of search should be considered. We distinct search based on the source on which the search is performed: towards the edited (or currently editing) text and towards external sources.

Searching the edited text is useful to analyse the text, while searching external sources (vocabularies, witnesses, parallel loci, etc.) is useful, for example, to gather information or to compare occurrences.

Text annotation is probably one of the most useful operations when creating a scholarly edition. An annotation may take multiple forms, namely a comment to the text, a note to oneself, a conjecture and so on. Following the principle of modularity of its component, the DSL-based methodology defines this kind of annotation uniformly with respect to the definition of any other kind of text: an annotation is nothing less than a full-blown text defined by its DSL. This choice enables a uniform management of each text while maintaining their specificity. Of course, if there are no special phenomena to deal with, a DSL for an annotation can be defined by a simple plain text.

This overview of operations shows that the notion of operation in the DSL-based methodology is versatile and that this approach lays the foundations for potentially any kind of text processing. Moreover the variety of specialised operations is addressed emphasising the modular and re-usable aspects of them. So, for example, the module that manages annotations can be easily re-used for the creation of very different editions 'as it is' or with limited modifications or replacement of the DSLs behind the definition of the text types.

Co-editing

The environment in which a scholar can edit the text benefits from the capability of changing the text concurrently inside a rich text editor. We differentiate between collaborative and cooperative editing. With cooperative editing we refer to the collective effort from different scholars that concur to the realisation of the edition. This translates to the need of a multi-user platform and the consequent definition of roles and permissions for the operations on the text.

With collaborative editing we refer to the concurrent access to the text and the reconciliation of conflicting operations on the text. One possible implementation of this kind of interaction is the use of the so-called operational transformation (OT), that is the same technique used by Google in its GDoc web application.[6]

Moreover, it is usually important for an editor to track the changes back to their contributors in order to assess the responsibility for each part of the text.

DSL-based methodology in practice

In this section we briefly present two significant examples where the methodology has been applied to the scholar's satisfaction. The first example concerns the domain of digital papyrology and the second one concerns the domain of digital epigraphy.

Digital papyrology

The ERC AdG 885222-GreekSchools[7] aims at the creation of a new critical edition of the Philodemus of Gadara's Arrangement

6 https://svn.apache.org/repos/asf/incubator/wave/whitepapers/operational-transform/operational-transform.html.

7 https://greekschools.eu/.

of the Philosophers by recovering as much text as possible from highly damaged papyri. The classical philological approach involves comparing different facsimilar witnesses for the designated text and producing an edition composed by a diplomatic edition of the papyrus with its palaeographic apparatus, a literary transcription with its philological apparatus and the translation. Therefore the editor needs to manage five different types of interrelated texts. Applying the DSL-based methodology to ease and empower the classical philological process means to mimic the analogic approach in a digital space (the editing platform) without disorienting the scholars by keeping them in a familiar environment. At the same time, the digital environment endows the scholars with the automatic and semi-automatic tools which integrate in one place their usually scattered sources, providing consistency and error checks.

The definition of one DSL for each type of text (and the corresponding editorial conventions) faces the challenge to correctly represent the philological phenomena in the digital space. Applying the serialisation operation to such DSLs, it is also possible to create the, otherwise hardly readable, TEI/XML version of the edition without any effort from the scholars by delegating the transformation to the editing platform. Consider the excerpt 'ῥ[α]νεπιταςεν[[.]]ᾳ[ι] [..(.)' that describes compactly and in a readable form multiple information; its corresponding XML appears as a highly polluted text that hinders the understandability even for domain experts.

The DSL representation of the data' along with the language service' also ensures that there are no violations of typographical or editorial conventions, which are otherwise often introduced by mistake due to the vast production of text and its consequent problematic revision.

Another consequence of using the collaborative and cooperative platform is to allow the scholar to work on the text remotely and asynchronously, giving the opportunity to continue the work that otherwise would have been limited to occasional workshops.

Digital epigraphy

The ItAnt[8] project, *Languages and Cultures of Ancient Italy. Historical Linguistics and Digital Models*, aims at creating a digital archive of fragmentary texts from Ancient Italy linked to a multilingual computational lexicon containing morphosyntactic and semantic analyses.

As a proof of concept, a sample of text encoded in TEI/EpiDoc is also encoded through a DSL with the same expressivity. An improvement in readability, compactness and manageability is asserted by the epigraphists of the project.

As shown in Figure 9.2, the same information appears to be inflated in the TEI version, while it is succinctly described by the DSL. This consequence of using a DSL has been greatly appreciated by the scholars involved in the proof of concept that has also pointed out how the compactness of the text is beneficial to its manageability at a glance.

Moreover, the automatic conversion from the DSL-encoded text to the XML format relieves the scholar from the distractions due to unfamiliar practices.

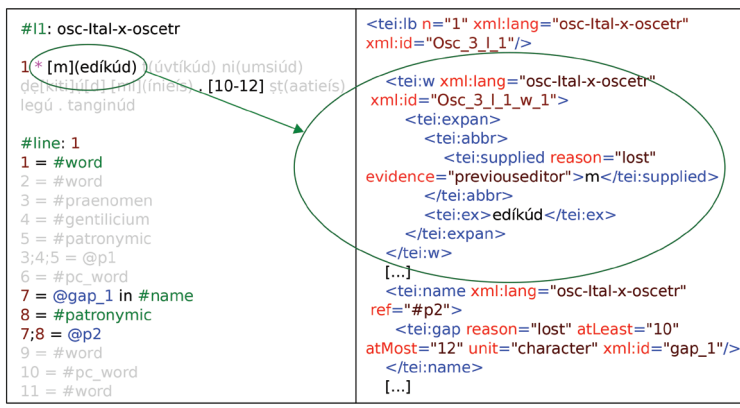


Figure 9.2 An excerpt of the ItAnt-DSL encoded text compared to the corresponding TEI/EpiDoc document. Source: Authors.

8 https://www.prin-italia-antica.unifi.it/index.html?newlang=eng [last accessed 01/09/2022].

Conclusions

In this contribution we presented the DSL-based DSE methodology to encode scholarly text. The methodology tries to address some of the challenges that the 'fourth revolution', namely the digital turn, has posed in the context of digital scholarly editing. In particular, scholars have felt that the current digital best practices have introduced a substantial discontinuity against their traditional and well-established editorial process. Among textual scholars, a rather strong reticence arose to the adoption of the digital environment and, consequently, it also narrowed the related benefits.

Nevertheless, there are other interesting directions in which the digital practices for scholarly text can be pursued. The methodology that we proposed is based on four key points: DSL; DDD; ADT; collaborative and cooperative editing. The DSL formally describe traditional scholarly best practices. DDD provides a well-known approach to derive the ubiquitous language that models the scholarly editing domain while preserving the traditional terminology and to create an effective software architecture that supports the whole editing process. ADT are the theoretical foundation for the description of both data types and domain operations. By having a collaborative and cooperative editing process, the scholars participate together in an ongoing review process that evolves and refines the text concurrently.

We already applied this methodology to several editions, gathering the warm and welcoming feedback from the scholars. The actual results demonstrate the effectiveness and efficiency of the proposed DSL-based DSE approach. Two examples of these editions have been briefly described to witness the soundness of our methodology.

References

Bizzoni, Y., Degaetano-Ortlieb, S., Fankhauser, P., and Teich, E. 2020. 'Linguistic Variation and Change in 250 Years of English Scientific Writing: A Data-Driven Approach.' *Frontiers in Artificial Intelligence* 3 (73). https://doi.org/10.3389/frai.2020.00073.

Bordalejo, B. and Robinson, P. 2015. 'A New System for Collaborative Online Creation of Scholarly Editions in Digital Form.' In *1st Dixit Convention on Technology, Software, Standards for the Digital Scholarly Edition Workshop*. The Hague.

Boschetti, F., Bambaci, L., Del Grosso, A. M., Mugelli, G., Khan, A. F., Bellandi, A. and Taddei, A. 2023. 'Collaborative and Multidisciplinary Annotations of Ancient Texts: The Euporia System.' In *The Ancient World Goes Digital: Case Studies on Archaeology, Texts, Online Publishing, Digital Archiving, and Preservation,* edited by Juloux, V.B., Di Ludovico, A. and Matskevich, S. Brill.

Boschetti, F. and Del Grosso, A. M. 2020. 'L'annotazione di testi storico-letterari al tempo dei social media.' *Italica Wratislaviensia* 11 (1): 65–99. http://dx.doi.org/10.15804/IW.2020.11.1.03.

Boschetti, F. and Del Grosso, A. M. 2015. 'TeiCoPhiLib: A Library of Components for the Domain of Collaborative Philology.' *Journal of the Text Encoding Initiative* 8. https://doi.org/10.4000/jtei.1285.

Bucchiarone, A., Cicchetti, A., Ciccozzi, F. and Pierantonio, A. 2021. *Domain-Specific Languages in Practice with JetBrains MPS*. Springer.

Crane, G. 2006. 'What Do You Do with a Million Books?' *D-Lib Magazine* 12 (3). https://doi.org/10.1045/march2006-crane.

Del Grosso, A. M., Giovannetti, E. and Marchi, S. 2017. 'The Importance of Being … Object-Oriented: Old Means for New Perspectives in Digital Textual Scholarship.' In *Advances in digital scholarly editing: Papers presented at the DiXiT conferences in The Hague, Cologne, and Antwerp*, edited by Boot, P., Cappellotto, A., Dillen, W., Fischer, F., Kelly, A., Mertgens, A., Sichani, A. M., Spadini, E. and van Hulle, D. Sidestone Press.

Dumouchel, S., Blotière, E., Breitfuss, G., Chen, Y., Di Donato, F., Eskevich, M., Forbes, F. et al. 2020. 'GOTRIPLE: A User-Centric Process to Develop a Discovery Platform.' *Information* 11 (12): 563. https://doi.org/10.3390/info11120563.

Evans, E. 2003. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Longman Publishing Co., Inc.

Evans, E. 2014. *Domain-Driven Design Reference: Definitions and Pattern Summaries*. Dog Ear Publishing.

Fielding, R. T. 2000. 'Architectural Styles and the Design of Network-Based Software Architectures.' PhD diss., University of California, Irvine. https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

Liskov, B. and Zilles, S. 1974. 'Programming with Abstract Data Types.' In *Proceedings of the ACM SIGPLAN Symposium on Very High Level Languages*. Association for Computing Machinery. https://doi.org/10.1145/800233.807045.

Millett, S. and Tune, N. 2015. *Patterns, Principles and Practices of Domain-Driven Design*. John Wiley & Sons.

Mugelli, G., Boschetti, F., Del Gratta, R., Del Grosso, A. M., Khan, F. and Taddei, A. 2016. 'A User-Centered Design to Annotate Ritual Facts in Ancient Greek Tragedies.' *Bulletin of the Institute of Classical Studies* 59 (2): 103–20. https://doi.org/10.1111/j.2041-5370.2016.12041.x.

Parr, T. 2007. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Bookshelf.

Parr, T. 2014. *Language Implementation Patterns Create Your Own Domain-Specific and General Programming Languages*. Pragmatic Bookshelf.

Pierazzo, E. 2015. *Digital Scholarly Editing: Theories, Models and Methods*. Ashgate.

Robinson, P. 2013. 'Towards a Theory of Digital Editions.' *Variants: The Journal of the European Society for Textual Scholarship* 10: 105–31.

Roncaglia, G. 2010. *La quarta rivoluzione: sei lezioni sul futuro del libro*. Laterza.

Schreibman, S., Siemens, R. and Unsworth, J., eds. 2016. *A New Companion to Digital Humanities*. John Wiley & Sons.

Shillingsburg, P. 2015. 'Development Principles for Virtual Archives and Editions.' *Variants: The Journal of the European Society for Textual Scholarship* 11: 9–28. https://doi.org/10.1163/9789401212113_002.

TEI Consortium, eds. n.d. *TEI P5: Guidelines for Electronic Text Encoding and Interchange. Version 4.4.0*. Last modified 19th April 2022. TEI Consortium. http://www.tei-c.org/Guidelines/P5/.

Voelter, M. 2014. *Generic Tools, Specific Languages*. Delft University of Technology.

Zenzaro, S., rio Del Grosso, A. M., Boschetti, F. and Ranocchia, G. 2022. 'Verso la definizione di criteri per valutare soluzioni di scholarly editing digitale: il caso d'uso GreekSchools.' In *AIUCD 2022 – Culture digitali. Intersezioni: filosofia, arti, media. Preceedings della 11a conferenza nazionale, Lecce,* edited by Ciracì, F., Miglietta, G. and Gatto, C. Associazione per l'Informatica Umanistica e la Cultura Digitale. https://doi.org/10.6092/unibo/amsacta/6848.