Umanistica Digitale – ISSN: 2532-8816 – n. 20, 2025: Special Issue
S. Zenzaro, A.M. Del Grosso, F. Boschetti, G. Ranocchia– *CoPhiEditor. The DSLBased DSE Methodology within the ERC Advanced Grant 885222-GreekSchools*
DOI: http://doi.org/10.6092/issn.2532-8816/21231

CoPhiEditor: The DSL-Based DSE Methodology within the ERC Advanced Grant 885222-GreekSchools

Simone Zenzaro

Cnr-Istituto di Linguistica Computazionale "Antonio Zampolli" (CNR-ILC)
simone.zenzaro@cnr.it

Angelo Mario Del Grosso

Cnr-Istituto di Linguistica Computazionale "Antonio Zampolli" (CNR-ILC)
angelo.delgrosso@ilc.cnr.it

Federico Boschetti

Cnr-Istituto di Linguistica Computazionale "Antonio Zampolli" (CNR-ILC)
federico.boschetti@ilc.cnr.it

Graziano Ranocchia

Dipartimento di Filologia, Letteratura e Linguistica, Università di Pisa
graziano.ranocchia@unipi.it

Abstract

This paper explores the integration of traditional philological methods with computational approaches, aiming to establish a more effective and rigorous framework for textual scholarship. To this end, we propose a preliminary set of evaluation criteria—familiarity, transparency, completeness, compactness, consistency, and machine-actionability—to classify digital scholarly editing tools. Using this framework, we suggest adopting a methodology we call DSL-based DSE for the digital scholarly editions of the Herculaneum papyri in our case study, the ERC Advanced Grant 885222-GreekSchools.

Following our description of the methodology and the role of Domain-Specific Languages, we

introduce CoPhiEditor, a software platform that supports the creation of Digital Scholarly Editions by implementing this methodology. This formal approach enables scholars to retain familiar editorial workflows while benefiting from machine-actionable data representations and computational functionalities. CoPhiEditor offers several key features: a recursive data model for flexible text representation, automatic error-checking and suggestions system, TEI-conformant serialization for standard data output, and advanced collaboration capabilities that streamline teamwork on shared projects. Furthermore, its extensible and domain-agnostic architecture makes it applicable to a wide range of scholarly editing contexts. Ultimately, CoPhiEditor enables philologists to integrate digital tools without sacrificing their established editorial practices.

Keywords: Domain-Specific Languages, Computational Philology, Digital Philology, DSE tools

Questo articolo esplora l'integrazione di metodi filologici tradizionali con gli approcci computazionali, con l'obiettivo di stabilire un framework più efficace e rigoroso per lo studio scientifico dei testi. A tal fine, proponiamo un insieme preliminare di criteri di valutazione per classificare gli strumenti digitali di edizione scientifica: familiarità, trasparenza, completezza, compattezza, coerenza e machine-actionability. Utilizzando questo framework, suggeriamo l'adozione di una metodologia, che definiamo DSL-based DSE, per la creazione delle edizioni digitali dei papiri di Ercolano nel contesto del progetto ERC Advanced Grant 885222-GreekSchools, che costituisce il nostro caso di studio.

Dopo aver descritto la metodologia e il ruolo dei linguaggi specifici di dominio (Domain-Specific Languages), presentiamo CoPhiEditor, una piattaforma software che supporta la creazione di edizioni scientifiche digitali implementando tale metodologia. Questo approccio formale consente agli studiosi di mantenere i consueti flussi di lavoro editoriali, beneficiando al contempo di rappresentazioni dei dati machine-actionable e di funzionalità computazionali. CoPhiEditor offre diverse funzionalità: un modello dati ricorsivo per rappresentare la risorsa testuale in maniera flessibile, il controllo automatico degli errori e un sistema di suggerimenti, la serializzazione in formato standard XML-TEI e capacità collaborative avanzate che ottimizzano il lavoro su progetti condivisi. Inoltre, la sua architettura estensibile e indipendente dal dominio ne amplia l'applicabilità a contesti diversi dalla papirologia. L'obiettivo di CoPhiEditor è permettere ai filologi di integrare nel processo editoriale strumenti digitali senza rinunciare alle loro pratiche consolidate.

Parole chiave: linguaggi specifici di dominio, filologia computazionale, filologia digitale, DSE tool

Introduction

Traditional philology has established its own methods for the study of texts which, over time, have crystallized into rigorous processes and methodologies, albeit with differences due to the context of reference. On the other hand, the advent of digital and computational philology is changing the traditional working paradigm by proposing new analysis methods. An example is the encoding of texts using the TEI guidelines[1] [46]. In this context, we believe it would be interesting to investigate the relationship between the rigor of traditional methods and the

[1] https://tei-c.org/Vault/P5/current/doc/tei-p5-doc/en/html/ (P5 Version 4.9.0. Last updated on 24th January 2025, revision f73186978)

efficiency of the computational approaches. Textual scholars, of whom philologists are part, often perceive this paradigm shift as an imposition, and the available technological tools as inadequate or frustrating [1][2][3][9]. On one hand, representing philological knowledge in a machine-actionable model presents an intrinsic challenge, often resulting in tools whose design risks prioritizing technical convenience over methodological coherence; on the other hand, there remains a widespread reluctance to adopt novel approaches that could enhance scholars' workflows. This separation between the philologist's needs and the availability of tools that are not always aligned with the theoretical foundations of philological methods hinders a fruitful synergy between knowledge, methods, and instruments within traditional scholarly contexts.

Our aim is to contribute to the convergence of models, fostering a unified framework that effectively supports the work of philologists. By addressing the specific needs and challenges of textual analysis, we strive to develop an effective platform dedicated to the scholarly study of texts. We are limiting the scope of this challenge to the creation of Digital Scholarly Editions (DSEs) using digital and computational philology tools. This platform should integrate advanced digital tools and methodologies to enhance the analysis, annotation, and interpretation of textual data of literary and historical significance. Our research is developed as part of the ERC Advanced Grant 885222-GreekSchools, The Greek philosophical schools according to Europe's earliest history of philosophy.

Thus, the papyrological editions produced within the project provide a natural testing ground for our approach. For this reason, we will include a section dedicated to the GreekSchools project and its edition structure to provide all the necessary information for understanding the connection between the methodological framework, the software platform, and the edition workflow.

This article is structured into two main parts. The former describes the proposed theoretical approach to DSE using Domain Specific Languages (DSLs) as defined in [5]. The latter part illustrates an implementation of the above-mentioned approach. After providing an overview of the similar initiatives and tools , focusing on the available digital and computational software applications as well as some key infrastructural projects. Next, we explore the use of Domain-Specific Languages in digital philology. The following section highlights the GreekSchools project as a prime example of the benefits of digital approaches for textual scholarship and describes some challenges in Digital Scholarly Editing. We then introduce CoPhiEditor, a platform designed for creating Digital Scholarly Editions using the DSL-based DSE methodology, and finally conclude by highlighting the central conclusions.

An Overview of Digital Scholarly Edition Tools and Initiatives

Despite numerous initiatives in the Digital Humanities (DH) field, experts in philological disciplines still lack fully effective digital and computational tools [54]. Some examples of tools developed for the production, publication and analysis of digital editions are Textual

Communities[2] [47], LEAF-VRE[3] [48], TEI Publisher[4] [49], EVT[5] [36][37], CETEIcean [50], Voyant Tools[6] [51], EFES[7] [55], and MQDQ[8] [38], while initiatives such as Canonical Text Services[9] (CTS) [52] and CollateX[10] [40] have defined shared protocols within the scholarly community. Some of these tools focus on a specific challenge; for example, EVT and TEI Publisher are dedicated to the digital publication of TEI-conformant editions, while Voyant Tools focuses on the analysis of digital texts, CollateX supports the collation of texts, and MQDQ is a digital archive platform. Over the years, initiatives such as Bamboo[11] [39], Interedition[12], and DiXiT[13] have emerged one after another, aiming to develop effective methods and tools to support humanities studies. Nowadays, the major effort, aimed at organizing and managing tools, resources, services, as well as training activities, falls on infrastructural initiatives such as DARIAH[14], CLARIN[15], and Parthenos[16]. Among these, the SSHOC[17] (Social Sciences & Humanities Open Cloud) project is of particular interest as it falls within the objectives of the European Open Science Cloud. Indeed, infrastructure is necessary to support tools and essential for ensuring their long-term sustainability. With regards to the annotation of the text, another notable approach is Euporia [9], which belongs to the set of tools for Digital Scholarly Editions that support collaboration and text annotation, such as CATMA[18], INCEpTION[19],

[2] https://textualcommunities.org/

[3] https://www.leaf-vre.org/

[4] https://teipublisher.com/

[5] http://evt.labcd.unipi.it/

[6] https://voyant-tools.org/

[7] https://github.com/EpiDoc/EFES

[8] http://mqdq.it

[9] http://cite-architecture.github.io/ctsurn_spec

[10] https://collatex.net/

[11] One of the earliest attempts to develop an interoperable and multi-institutional infrastructure based on the availability of services for the arts and humanities: https://digital.humanities.ox.ac.uk/project/project-bamboo

[12] Which developed microservices for textual scholarship: http://www.interedition.eu/

[13] Investigating the intersection of humanities and computer science in digital scholarly editions: https://cordis.europa.eu/project/id/317436

[14] https://www.dariah.eu/

[15] https://www.clarin.eu/

[16] https://www.parthenos-project.eu/

[17] https://sshopencloud.eu/

[18] https://catma.de/

[19] https://inception-project.github.io

GATE workbench[20], BRAT[21], Hypothes.is[22].

Among the reference initiatives in the field of digital papyrology, we recall the Papyri.info[23] project, which not only provides useful tools for the drafting of texts to support the editorial process with subsequent data persistence management, but above all highlights a design and development attitude aimed at including the reference community, responding promptly to its needs. Among the notable features of Papyri.info are the tracking of accesses and changes to the document, the use of a Domain-Specific Language (DSL) [4][5][6][7] for text editing (Leiden+), the SoSOL editing tool,[24], and an advanced search engine. Nevertheless, we believe that there is room for improvement regarding: (a) the possibility of processing textual data, (b) support for collaborative editing, (c) the use of multiple DSLs other than Leiden+ (e.g., for encoding apparatuses), (d) the integration of facsimile sources, and (e) software usability.

An attempt to classify Digital Scholarly Editing tools

As is well known in digital textual scholarship, creating a Digital Scholarly Edition (DSE) is complex [19][20], and multidisciplinary collaboration poses significant challenges [3][21][22].

When creating a DSE, scholars face multiple choices, ranging from editorial conventions and primary sources to editing and publication workflows. Selecting the most suitable technological environment can therefore be challenging without any direction to lean on.

In this sense, guidelines would be useful to facilitate the selection of the most suitable tool for creating a DSE. To this end, we propose a formal framework of dimensions—namely familiarity, transparency, completeness, compactness, consistency, and machine-actionability—for classifying tools in the context of digital scholarly editing.

These dimensions are, in part, derived from the concepts articulated in [9][10][11]; definitions for each are provided below. By familiarity, we refer to the possibility for the scholar to maintain their typical working paradigm while editing the digital text. This aspect is important, as changing the working environment is not always feasible or recommended. Transparency indicates the amount of technical knowledge required to adopt the proposed approach (a preliminary investigation on this topic is described in [12]). This dimension is taken into consideration when the domain experts (i.e., the philologists) are not accustomed to technology. Completeness refers to the amount of information that can be expressed (see also [13] for further details). This dimension is crucial because the domain experts must have the maximum freedom of expression to capture all textual phenomena. Compactness refers to the relationship between completeness and the extent of formalization in terms of Bytes and human readability (for specific references, see [14][15]). The role of this dimension is exemplified by the extremely compact critical apparatuses that convey a significant amount of information within a limited space. Consistency

[20] https://gate.ac.uk/

[21] https://brat.nlplab.org/

[22] https://web.hypothes.is/

[23] https://papyri.info/

[24] The SoSOL tool uses the Leiden+ textual conventions as an intermediate syntax to encode TEI EpiDoc which is an ODD-based customization of the TEI.

implies describing or treating the same phenomena uniformly across all instances; this implies that the representation of the same type of information is unambiguous. This is not always guaranteed by all the solutions considered; for example, it is possible to encode an abbreviation in XML-TEI in different ways within the same context (adopting the elements choice, ex, expan, etc.) without the guarantee of using the established editorial convention. Finally, the ability to extract or deduce information from the data is indicated by the machine-actionability dimension (an intrinsic characteristic of formal languages, described by a formal grammar and commonly accompanied by other source code processing components such as the lexer and the parser [4]).

By classifying the tools using these six dimensions, we identified four macro categories of approaches to DSE, which we believe represent sets of similar tools: word processors, structured text, GUI-centric tools, and DSLs. Naturally, this type of classification requires further refinement and exploration to account for the subtle differences and specific features of current tools.

All Microsoft Word-like tools—such as the Classical Text Editor[25]—would fall into the word processors category. The structured text category would includes tools highly focused on the use of structured text, such as the Oxygen XML editor[26]. GUI-centric tools encompass those that rely heavily on graphical user interface interaction, such as CEED [42] and Cadmus[27] [53]. Finally, the DSL category includes the tools that place Domain-Specific Languages at the core of their functionality.

The results of the classification attempt are presented in Figure 2, highlighting key advantages and limitations of different text editing approaches in terms of familiarity, transparency, computational applicability, compactness, consistency, and machine-actionability. For instance, the word processor category is both familiar and transparent to editors, as it aligns well with traditional writing and editing practices. However, despite its ease of use, this approach presents significant challenges when applying computational methods. Therefore, the primary limitation stems from its poor machine-actionability—word processors are designed for human readability rather than structured data processing, making automated analysis and transformation difficult.

[25] https://cte.oeaw.ac.at/

[26] https://www.oxygenxml.com/

[27] https://cadmus.fusi-soft.com/

	Word Processor	Structured Text	GUI-centric	DSL
Familiarity	✓			✓
Transparency	✓		✓	✓
Completeness		✓	✓	✓
Compactness	✓			✓
Consistency			✓	✓
Actionability		✓	✓	✓

Figure 1: Analysis of Approaches to Textual Editing

The structured text approach, in contrast, provides a framework that facilitates the application of computational methods. By enforcing a predefined structure, it allows for better integration with automated tools and algorithms. However, this approach does not inherently guarantee consistency across the entire text, as it requires users to adhere strictly to formatting rules, which may introduce variability or inconsistencies if not carefully managed.

The GUI-centric approach offers a different balance of strengths and weaknesses. It is designed to be highly machine-actionable and provides an intuitive interface that makes editing more accessible. However, this method is not as deeply embedded in common scholarly practices, meaning that users may require additional training or adaptation to fully utilize its potential. Furthermore, while the GUI-centric approach streamlines certain aspects of handling text such as admissible values, it lacks the compactness and direct control provided by manual text editing, which scholars find more flexible.

It might seem that relying on DSLs is the only way to maximize all six dimensions. Figure 1 might suggest that adopting DSLs is the most obvious choice in every context and resolves every limitation present in other approaches.

Obviously, this is not the case; the limitations of DSLs, in our use case, represent marginal obstacles while maintaining all the characteristics we consider important in an effective editing platform for papyrological texts.

It therefore seems appropriate to compare some weaknesses of DSLs with respect to other approaches. A DSL, being expressed in plain text, must forgo the semantics attributed to formatting such as bold text, adopted in many editorial practices by philologists who use word processors (for example, to indicate *litterae suppositae* in the diplomatic ranscriptions). A less elegant but equally effective solution is the use of minimal opening and closing markers, as in the Markdown language (e.g., `**a boldface text in Markdown**`). A DSL apparently provides less control over text input; however, a good editing component for formal languages (e.g., Monaco, Ace, CodeMirror) is not only able to signal syntax errors, but also provides suggestions for their resolution and autocompletion mechanisms. Finally, a DSL has more limited expressiveness compared to a complex markup schema like XML-TEI. However, this is only apparently a weakness. In fact, a DSL arises from the need to identify a specific context in which the language allows to easily, compactly, and consistently describe all the peculiarities of its domain, but it can be extended with additional modules just as with XML-TEI.

The DSL-based DSE methodology

We believe that leveraging the DSL-based DSE methodology [35] offers a robust framework for the creation of DSEs. To demonstrate this potential, we have developed a proof-of-concept software platform that exemplifies our vision for shaping future tools based on Domain-Specific Languages (DSLs) 52 . In particular, we applied this methodology to the GreekSchools project to support the creation of challenging papyrological editions of herculaneum papyri.

Our DSL-based DSE methodology for digital scholarly editing emphasizes that tools for DSE should offer a familiar (or at least easy-to-learn) environment that facilitates the editing process and enables seamless collaboration among scholars on a Digital Scholarly Editing web platform. We chose a DSL-based approach after analyzing various text editing solutions (see Fig. 1) in relation to the six proposed dimensions outlined in the previous section.

Since the methodology relies on DSLs, we will now provide a definition. A DSL is a language defined by a formal grammar (usually a Context-Free Grammar) and designed for a specific domain of knowledge or activity. The constructs and lexicon of general-purpose languages (or metalanguages), such as Python, or descriptive languages, such as XML, often deviate significantly from natural language or formalisms familiar to specialists in the humanities [8]. Thus, the digital philologist finds themselves encoding texts in a way quite distant from the practices of traditional philologists: consider, for example, a critical apparatus represented in XML-TEI compared to a critical apparatus drafted by a papyrologist. One of the major advantages of a DSL is therefore the familiarity with the formalisms adopted within a domain of knowledge that boasts its own tradition of studies, which has allowed, over decades if not centuries, to optimize the representation of information relevant to the object of study. This implies the compactness of a DSL compared to a general-purpose language, because—as Shannon's information theory establishes—what occurs more frequently can be encoded with a smaller number of characters (through symbols or abbreviations perfectly understandable and familiar to the specialist), while what occurs less frequently requires a more verbose encoding.

Using a DSL in this context allows maintaining the main focus on the text without losing the advantages of a structured approach for the representation of textual information in which all data are made explicit (for example, through XML encoding). Furthermore, the representation derived from the automatic interpretation of the DSL opens up the possibility of using complex data processing techniques (e.g., statistical analyses and semantic classifiers, syntactic consistency checks, etc.).

To make a DSL accessible and familiar to scholars, it is essential to allow domain experts (e.g., philologists) to select intuitive editing conventions. In most cases, these conventions overlap with standard editorial and ecdotic practices, resulting in DSL syntax that closely resembles the usual *layout* of the actual text in the final edition.

For this purpose, we have decided to adopt the Domain-Driven Design (DDD) [16] principles for software design and development. We believe, in fact, that the principles and practices advocated by DDD are capable of rigorously and satisfactorily defining the specificities of the traditional philological method, enriching it, without forcing it, while enhancing it with the benefits of modern technologies, without compromising its integrity.

In particular, the process of defining and disambiguating concepts is realized through the creation of a shared lexicon –the ubiquitous language– that bridges the gap between domain experts with a philological background and computer scientists. For example, while the terms "supplement" or "fragment"may be clear to a domain expert, each should have a unique formal definition directly implementable in software. This methodology not only aids in modeling but

also fosters a deeper understanding of domain concepts.

Once a DSL is well-defined, the text can be composed in such a language, capturing all relevant textual phenomena. This representation retains the implicit encoding of such phenomena through editorial conventions. Subsequently, the text can be processed using a parsing function, which generates an Abstract Syntax Tree (AST)—a hierarchical structure that explicitly represents the text and its phenomena in a machine-readable format. The Abstract Syntax Tree (AST) serves as the foundation for associating computational functionalities with the text. In Figure 2, the editorial insertion "<*α*>" is edited as formatted plain text (in the middle), is recognized by the Context-Free Grammar (on the left) that defines the DSL, and is transformed into the AST node "editIns" (on the right). This exemplifies the DSL-based DSE approach.

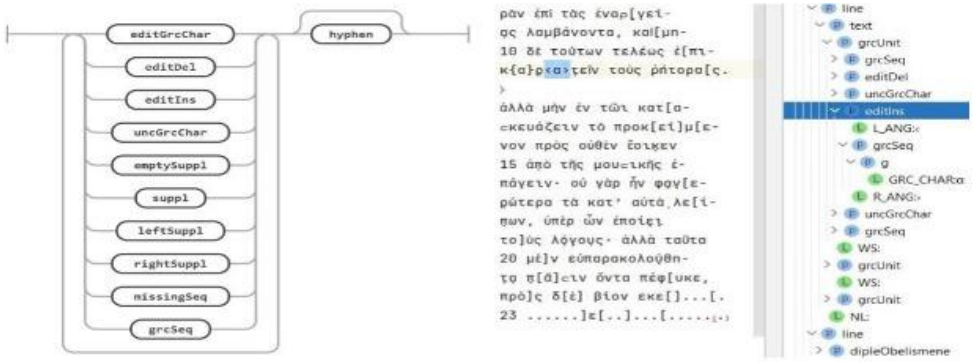

Figure 2: Example of DSL-based DSE method: Context-Free Grammar diagram (left), DSL-based editing (centre), Automatic recognition of textual phenomena from the DSL (right)

To demonstrate the practical implications of adopting this methodology for philologists, we now present how the editorial workflow operates using a real example: an excerpt from papyrus P.Herc. 1004, column 64. Suppose the philologist intends to edit the following Greek text: **ῥὰν ἐπὶ τὰς ἐναρ[γεί-**. The text would be typed much like in a standard word processor. Despite being a short character sequence, it encodes several editorial interventions, such as an editorial correction (i.e., ῥ), a supplement (i.e., [γεί), and a hyphenated line ending. We assume that a Context-Free Grammar has already been defined to describe such editorial interventions. A fragment (i.e., a production rule) of this grammar might be:

```
grcUnit: (editGrcChar | rightSuppl | grcSeq)+ hyphen?
```

This rule states that a Greek textual unit consists of one or more elements—corrected characters, supplements, or standard sequences of Greek letters—optionally followed by a hyphen.

```
<line><testo><grcUnit>
 <grcSeq><g>ρ</g><g>ὰ</g><g>ν</g>
 </grcSeq></grcUnit>
 <grcUnit><grcSeq><g>ἐ</g><g>π</g><g>ὶ</g>
 </grcSeq></grcUnit>
 <grcUnit><grcSeq><g>τ</g><g>ὰ</g><g>ς</g>
 </grcSeq></grcUnit>
 <grcUnit><grcSeq><g>ἐ</g><g>ν</g><g>α</g>
  </grcSeq><editGrcChar><g>ρ</g>̣</editGrcChar>
  <rightSuppl>[ <grcSeq><g>γ</g><g>ε</g><g>ί</g>
  </grcSeq></rightSuppl> <hyphen>-</hyphen>
 </grcUnit></testo></line>
```

Figure 3: DSL serialization into the Intermediate XML

To automatically recognize these editorial phenomena, a software component—the parser—generates an Abstract Syntax Tree (AST) that represents the text as a hierarchical structure according to the Context-Free Grammar. The AST is a data structure that can be serialized, for instance, into an intermediate XML fragment (see Figure 3). From this point, transforming the XML-serialized AST into TEI-compliant encoding involves applying an XSLT[28] transformation. This transformation produces a representation of the same editorial phenomena using TEI elements such as `<corr>`, `<supplied>`, and `<lb>`, along with the attribute `@break` (see Figure 4).

```
<lb n="8" /> ρὰν ἐπὶ τὰς ἐνα<corr resp="#gs-editor">ρ̣</corr >
<supplied reason="lost">γεί</supplied><lb break ="no" rend="-"/>
```

Figure 4: XML-TEI representation of the Greek edited text

As stated in [35][61], this approach offers several advantages. First, it lowers the technical barrier for domain experts, allowing them to input (almost) the same text they are already accustomed to. Second, it provides a clear separation between input and output formats. By leveraging the intermediate AST serialization in XML, all editorial phenomena can be retained in a form best suited to the philologist. At the same time, generating different output formats for different purposes becomes much easier than editing a TEI-XML document directly. For instance, modifying the encoding model or switching to a different version of TEI only requires adjusting the XSLT transformation for the intermediate XML, without risking changes to the original

[28] https://www.w3.org/TR/xslt-30/

encoded text (which is the DSL). This ensures the long-term preservation of the information while enhancing interoperability with other tools and systems used in the field of digital humanities. Moreover, other output formats such as DOCX or PDF can also be supported using the same intermediate XML. In addition to export formats, computational features can also be attached to the AST.

The main disadvantage of this method is that DSL grammars and their associated features must be developed and initialized before use. The same applies to the XSLT transformation functions, which must be prepared by experts to preserve the intended semantics of the edited text.

It is worth noting that the DSLs used for text editing are not limited to ad hoc formal languages. One could also adopt existing formats such as Leiden+ or even a dialect of XML (including XML-TEI). Furthermore, DSLs can be composed and embedded, allowing one DSL to be seamlessly integrated into another.

The GreekSchools Project

In this section, we provide a description of the ERC Advanced Grant 885222-GreekSchools [29], *The Greek Philosophical Schools According to Europe's Earliest History of Philosophy. Towards a new pioneering critical edition of Philodemus' Arrangement of the Philosophers*.

This section aims to describe the structure of the papyrological editions and to provide a glimpse into the philological challenges inherent in implementing a tool that supports the entire editorial process of making a DSE. Consequently, we intend to use the GreekSchools challenges as a testing ground for both the DSL-based DSE methodology and the CoPhiEditor tool.

The GreekSchools project object of study is a precious treatise: the Index of Philosophers by Philodemus of Gadara (75-50 BC) – also known as the Sýntaxis – whose original papyri are in poor condition and whose available editions are outdated. Consequently, GreekSchools aims to produce a new Open Access digital scholarly edition of the entire treatise, which includes six papyri, equipped with paleographic and literary transcription, corresponding critical apparatuses, commentaries, and translations.

The philologists working in the GreekSchool project are reconstructing the Greek text conveyed by the Herculaneum papyri, most of which are carbonized and highly damaged.

The diplomatic transcription records the visible text from a portion of a papyrus comparing dubious readings with other witnesses such as the Oxonienses and Neapolitans drawings and the copper engravings, with all details about surviving vestiges specified in the form of a paleographic apparatus. Once the diplomatic transcription is complete, the literary transcription is prepared to display the reconstructed text, its subdivision into words, and the accents. Editorial choices made at this stage are documented in the philological apparatus which records novel conjectures and the readings of previous editors. Finally, a translation of the ancient Greek text is typically provided in either Italian or German. Figure 5 presents an excerpt from the printed edition of column 64 of the papyrus PHerc. 1004, illustrating the format of a papyrological edition within the context of the GreekSchools project.

To produce the edition, scholars typically collaborate through in-person workshops, bringing

[29] https://greekschools.eu/

together experts from various fields, including papyrology, ancient Greek studies, and history. These workshops focus on intensive editorial work targeting specific sections of the text intended for publication.

Figure 5: An example of edition for the PHerc 1004 Col. 64 papyrus

CoPhiEditor

In this section, we introduce the features of CoPhiEditor [23][24], a web-based collaborative authoring platform for Digital Scholarly Editions (DSE) that implements the DSL-based methodology described in previous sections. We will provide an overview of its capabilities and functionalities.

The two primary goals of CoPhiEditor are to support papyrologists in producing a new critical edition of the *Arrangement of the Philosophers* by Philodemus of Gadara and to provide scholars with a text-centered authoring platform that implements the DSL-based DSE methodology [35].

Although CoPhiEditor is being developed as part of the ERC Advanced Grant 885222-GreekSchols project, its design is meant to be domain agnostic. This is achieved both at the software architecture level and also as a consequence of choosing the Domain Specific Languages to address domain specific requirements. Our objective is to ensure that the software platform is easy to learn, maintaining a low learning curve by allowing scholars to adhere to their well-established editorial conventions. At the same time, the platform enables the automatic generation of a TEI/EpiDoc-compliant [43] digital edition from the edited texts, seamlessly bridging traditional practices with modern technological standards. In this article, we will not delve into the implementation choices and technical details of the software. The platform is available on the official website[30], but accessing and editing the texts requires credentials. The

[30] https://cophieditor.greekschools.eu/

source code for the platform is available on GitHub[31], although most of the repositories are currently private due to the GreekSchools project policies. However, they will be made public by the end of the project. For the same reason, the demo site for the platform is still on hold.

Data representation

First and foremost, we will discuss how data is represented within CoPhiEditor. The underlying model for the text is based on the recursive data representation described in Omega [24][33]. We call such a model a unit within CoPhiEditor.

A unit represents any portion of text, ranging from entire works and single chapters to apparatuses, comments, annotations, or even smaller segments. This uniform representation provides maximum flexibility in determining the granularity of the annotated text and allows for infinite hierarchical depth in annotations.

The distinction between different types of text is defined by the DSL associated with each unit. In addition to the DSL and the actual text, a unit includes a set of metadata that describes attributes such as its title, creation date, references to other units, and pointers to the specific text portions they relate to. From a technical perspective, this data is represented as a JSON object on the web application and stored in a database. The type of database (relational, object-oriented, XML, etc.) is abstracted from the user. In our case we opted for eXist-db54.[32]

The annotation model inside CoPhiEditor is also compatible with the Web Annotation Data Model[33] (WADM) and makes it possible to derive the WADM representation of the comments. Although this representation is not meant to be directly read by philologist, WADM is a well known W3C standard that "describes a structured model and format to enable annotations to be shared and reused across different hardware and software platforms" and in our viewpoint, ensuring the compatibility with this standard is an important step toward interoperability and social [25] and assertive scholarly editing [26]. Moreover, since the texts of the edition are constantly changing until they are considered *ne varietur* (i.e. it must not be changed) by the editor, the comments may refer to some content that has been modified or that is not present anymore. In this case, the comment itself may be (or may be not!) considered obsolete, incorrect, or no longer relevant. WADM defines the concept of state that "provides the information needed to retrieve the correct representation" of the annotation. This way CophiEditor can manage comments to the text that have been changed to notify the editor of a specific edition.

The Use of Domain-Specific Languages

In CoPhiEditor, a DSL defines a specific type of text intended for editing as part of creating the critical edition.

In the GreekSchools project, for example, there are six different DSLs that represent the diplomatic and literary transcriptions, paleographic and philological apparatuses, the modern translation, and the sources for a given column or fragment of papyrus.

At first glance, the need for up to six different DSLs to edit the text may seem like an unnecessary

[31] https://github.com/orgs/CoPhi/teams/greekschools/repositories

[32] https://exist-db.org/

[33] https://www.w3.org/TR/annotation-model/, https://www.w3.org/TR/annotation-protocol/

complication. However, the key idea is that each DSL is designed to address a specific philological "textual representation," in line with the principle of "generic tools and specific languages" [60] and the Domain-Driven Design bounded context pattern [16]. Moreover, each DSL is designed by researchers in close collaboration with expert papyrologists (following the DDD approach), as the language must represent, unambiguously but in a way very close to traditional practices (see Figure 6 as an example), the information necessary to produce the diplomatic and literary editions of the papyrus text and their respective apparatuses. This form of co-design requires continuous meetings and iterations throughout the platform's development process. The result is a formal language that, in the majority of cases, exhibits a close correspondence with the philologist's community-based editing style, while preserving its machine-actionability.

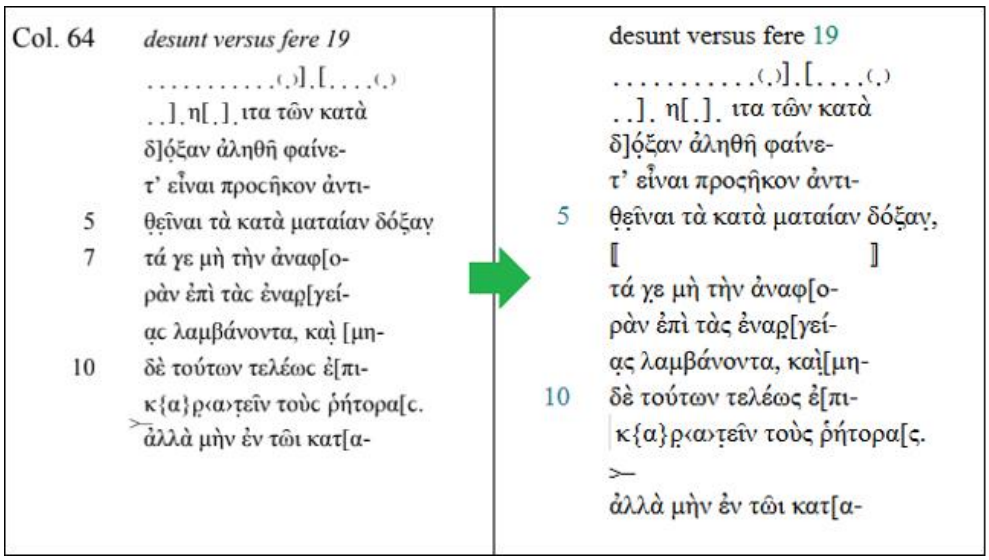

Figure 6: Comparison of Literary Transcription in Printed Edition (left) and DSL (right)

Once the text of a unit is parsed into its AST it is possible to attach several computational functionalities. For example consistency checks and error detection are useful to identify non-compliance with editorial conventions. Figure 7 illustrates the tree representation of a fragment of the apparatus, where a closed parenthesis is flagged as an error due to its non-conformance with the editorial rules defined in the DSL grammar. This approach enables the system to notify the editors of such errors – errors that, from the perspective of a given DSL, constitute syntax errors –, allowing them to address the issues and ensure a final result that better aligns with their intentions.

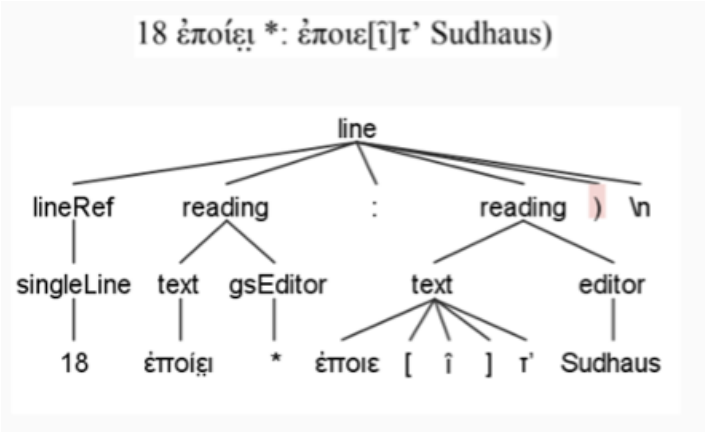

Figure 7: Example of automatic error checking using a DSL

Another application of our approach concerns the implementation of an advanced search engine capable of efficiently navigating the encoded textual data. This functionality would allow users to perform complex queries, retrieve specific information, and explore the content thoroughly (the platform uses a specific implementation of Lucene[34] search engine customized for hierarchical structures —specifically, the textual units in our case— in addition to full-text sequences).

In cases where it is possible to automatically propose conjectures, the platform will submit a list of candidates to the editor's judgment, implementing computational language models [56]. For example in the GreekSchools project we trained a prediction system for the ancient Greek language based on a Maximum Likelihood Estimation (MLE) n-grams model [57] and BERT-like models [58][59].

CoPhiEditor also provides the capability to export the entire edition of a papyrus or a portion of it to TEI/EpiDoc [43] to support digital edition publication, as well as to the DOCX or other formats required by publishers that can be used as a basis for the printed version of the same edition.

Software Architecture

From a technical point of view, the platform's architecture (see Figure 8) is built upon microservices [45], with each one dedicated to a specific isolated feature. These features include authentication, authorization, data storage, IIIF viewer, DSL management, collaboration among users, and the web application itself, which serves as the entry point for users to interact with texts and witnesses.

[34] https://lucene.apache.org/

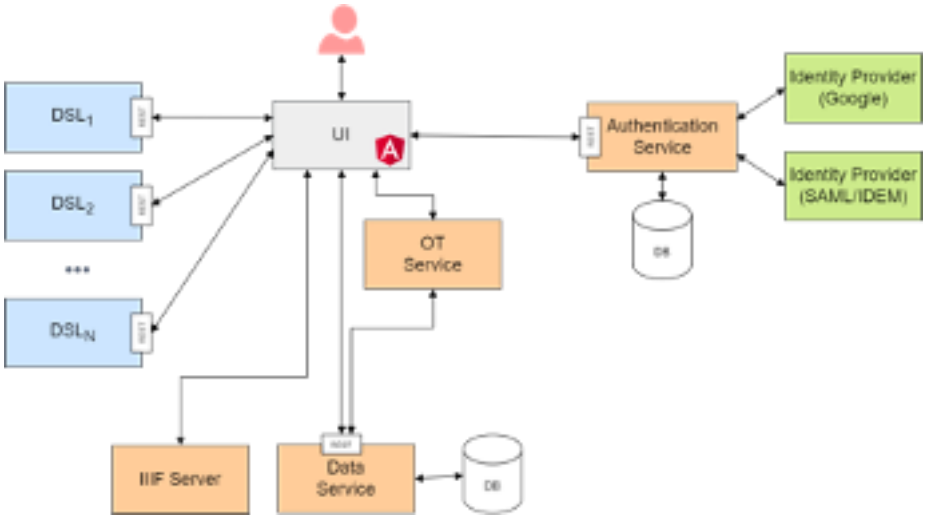

Figure 8: CoPhiEditor Architecture

The architecture includes the Angular web application, which acts as the interface between the users and other services; a set of DSL Services, each complying with a shared RESTful API [32] definition inspired by the Language Server Protocol,[35] offering error checking (as in the example in Figure 7), suggestion, and export capabilities; an Authentication Service providing a unified API for logging users into CoPhiEditor with different providers (OAuth2, SAML, username and password);a Data Service offering a RESTful API for data persistence and communication with the eXist-db [34] database; and the OT Service which enables two-way communication between the service and the client,[36] notifying the web application of events related to textual data and changes via a WebSocket and Operational Transformation.

Support for Different Types of Editions

CoPhiEditor was initially developed to address the editorial challenges of digital papyrology. However, as the web platform is adaptable to domains beyond its original scope, its design is intentionally oriented toward a domain-agnostic and cross-domain environment.

Supporting multiple domains is achieved by providing a corresponding set of DSL services tailored to the specific requirements of each type and level of edition (e.g., an interpretative edition of a medieval codex or a critical edition of texts in ancient languages). We plan to introduce a registry service that will act as a catalogue of the DSL services available to the web application. The registry service will collect, filter, and provide access to DSLs, while the web application will select the appropriate core subset of DSLs for a specific type of edition.

In CoPhiEditor, all information contributing to the representation of a Digital Scholarly Edition's unique characteristics is encapsulated within the concept of a project. Specifically, a project comprises metadata and data. The metadata includes details such as the project title, its

[35] https://microsoft.github.io/language-server-protocol/

[36] Usually the server can only send information to the client on demand, this approach makes it possible to send data to the clients on events that are not generated on the client.

owner, a list of team members, defined roles, linked resources, and references to relevant DSLs. The data consists of the project's individual units.

Additional materials, such as the Liddell-Scott-Jones Greek-English Lexicon or other sources of papyrological texts used in the GreekSchools project, are collected as project-specific data and made referenceable.

Access Control

Access control refers to the ability to grant users of the software platform access to resources (typically text and operations performed on it). For example, in the GreekSchools project, there are three types of users: editors, collaborators, and viewers. Each role has different permissions regarding, for example, the ability to view, modify, or annotate text.

While philological questions fall within the domain of textual scholarship, access control addresses the practical requirements for sharing or restricting access to collections of texts.

CoPhiEditor uses a form of Attribute-Based Access Control (ABAC) [44] policy to define what users can do with resources in terms of activities and roles. An activity corresponds to a permitted operation on a resource, possibly restricted by parameters (e.g., modifying a unit representing an apparatus), while roles are sets of allowed activities.

```
CreateUnits activity can POST/units
GetUnits activity can GET/units
GetNoCommentsUnit activity can GET/units where dsl.language is not comment

Contributor role can GetUnits
Admin role can GetUnits, CreateUnits
Viewer role can GetNoCommentsUnit

user1 is Admin
user2 is Viewer
user3 is Contributor
```

Figure 9: Access Control DSL example

The access control strategy is flexible enough to allow the definition of activities and roles that extend beyond the scope of the GreekSchools project. Furthermore, because the platform is based on DSLs, we also provide an Access Control DSL to easily define the desired user roles (see Figure 9).

Editing Workflow

From the point of view of supporting the editing process, CoPhiEditor implements a workflow based on work sessions managed by the editor in which external collaborators can comment on the texts and propose conjectures. These proposals will be evaluated and, possibly, integrated into the text, thus realizing a continuous and collaborative revision process (see Figure 10).

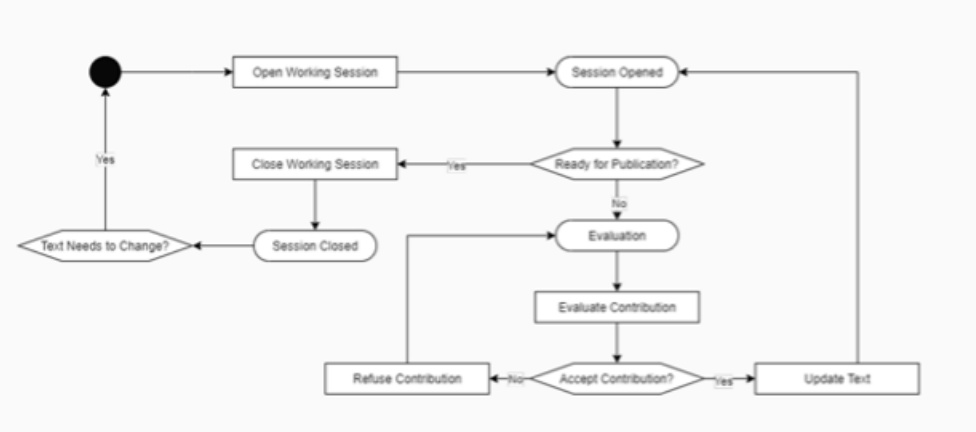

Figure 10: Continuous and Collaborative Revision

There are three main roles for accessing the text: editors can always modify it, commenters can add annotations in the form of comments, and viewers can only read it. If an editor does not grant access to contribute to the text when inviting others, access is denied according to the access control strategy described in the following subsection.

Graphical User Interface Design

From the point of view of the graphical user interface (GUI), the dialogue with domain experts has allowed us to identify a familiar working method consisting of juxtaposing sources with the text being created and frequently comparing them with other texts. For this reason, as shown in the screenshot in Figure 11, the GUI allows any number of sources to be placed alongside the texts being edited, leaving the user free to decide their spatial arrangement. Facsimile images management is designed in accordance with the IIIF[37] protocol, also providing tools to support their visualisation (e.g., modifying brightness and contrast properties). In this way, the GUI will adapt to the editor's habits. Furthermore, navigation of the text structure allows focusing on any level of granularity (column, text, apparatus, line, apparatus entry, etc.), making the recursive data model explorable via the GUI as well. The GUI is developed as an Angular[38] web application that interacts with the underlying microservices architecture by means of Application Programming Interfaces.

[37] https://iiif.io/

[38] https://angular.dev/

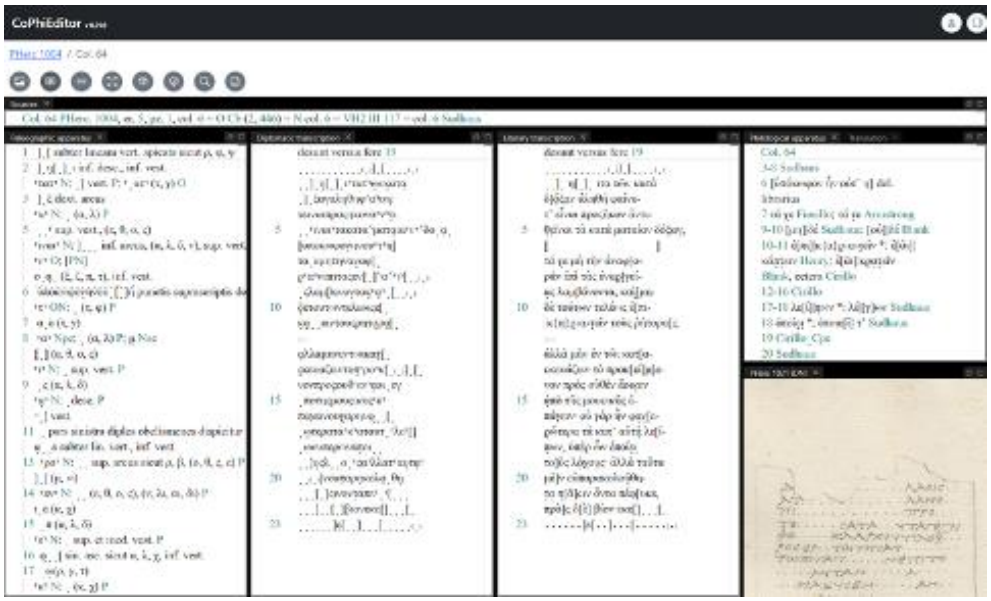

Figure 11: GUI, Integrated Resources, and Editing Structure in CoPhiEditor

Collaboration and Cooperation

Being a web application, CoPhiEditor does not require any particular physical location to work on editing, commenting and annotating the text, other than an electronic device able to connect to the Internet and a browser to interact with the World Wide Web ecosystem, making remote collaboration possible.

CophiEditor is designed to both support collaboration (i.e. many participants for a single task) and cooperation (i.e. many participants for many sub-tasks).

Within the platform, scholars have access to the DSE working drafts and can edit or annotate each one by adding a comment to the text.

The collaboration side of text editing implements an environment based on Operational Transformation [17][18], a technology used to provide parallel and collaborative editing, as, for example, in Google Docs. In such an environment, the platform users can modify the same text concurrently without worrying to destroy the edits of other users since they will be integrated seamlessly into the text.

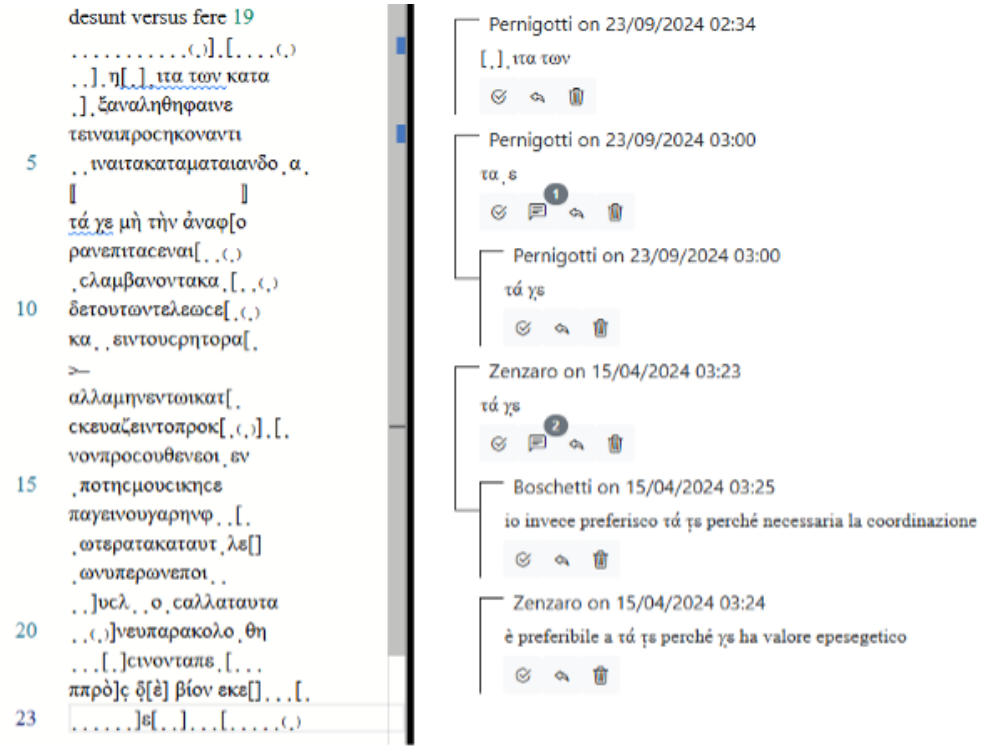

Figure 12: Example of a discussion thread

The editing platform implements cooperation through a collaborative review process that involves threads of discussions based on annotations of the text.

The selection of a target text for an annotation supports single and multiple selections, in case the annotation extends over separate lines of text. The annotations are visible as comments to the other scholars who can reply to them and start a discussion thread (see Figure 12), or add other comments to the editing text. Each comment update is propagated to all users connected to the platform who can henceforth participate in the discussion in real time leading to the constitutio textus.

Effective collaboration among scholars also requires tracking and attributing responsibility for interventions, synchronization, and constant verification of consistency between the modified parts.

In the context of GreekSchools, discussion threads within CoPhiEditor are used to raise questions about difficult passages in the text and to propose conjectures and readings that may eventually be incorporated into the literary transcription or added as entries in the philological apparatus.

Conclusions

In this paper, we have explored the potential for bridging the gap between traditional philological

methods and computational approaches. We suggest that such integration could foster a more effective and rigorous framework for studying texts.

As part of this effort, we have presented criteria for analyzing text editing approaches, identifying six evaluation dimensions: familiarity, transparency, completeness, compactness, consistency, and machine-actionability. The resulting evaluation framework supports the choice of using a DSL-based approach for the context of digital papyrological editions, in our case study the ERC-885222 GreekSchools project.

We briefly presented the DSL-based DSE methodology and the CoPhiEditor, a software platform designed to support the creation of Digital Scholarly Editions. CoPhiEditor utilizes Domain-Specific Languages to represent textual data, allowing scholars to maintain familiar workflows while benefiting from structured, machine-actionable data representation.

We have outlined the core features of CoPhiEditor, including its data model, use of DSLs, and functionalities for error checking, and automatic suggestions. The platform also facilitates advanced collaboration, enabling seamless teamwork on shared projects. Additionally, CoPhiEditor's extensible and domain-agnostic design makes it adaptable to a wide range of scholarly editing endeavors, extending its utility beyond digital papyrology.

By combining user-friendly design with computational advancements, the platform aims to empower philologists to incorporate digital tools without abandoning established editorial practices. Its collaborative features, in particular, offer opportunities to enhance knowledge sharing and cooperation within the field.

Looking ahead, we aim to enhance CoPhiEditor and consolidate the already developed features.

We would also like to make CoPhiEditor fully domain-agnostic by including a DSL registry and extracting all the GreekSchools-specific features as plugins for the platform, which can be defined as configurations. Moreover, a set of preconfigured templates addressing the already established philological disciplines (e.g., papyrology, Romance philology) would serve as the starting point for further scholarly edition projects. In addition, we would like to explore and experiment with a wide range of textual traditions, spanning diverse historical periods and languages. By doing so, we aim to identify and extract common features that can be applied across a variety of scholarly editions, allowing for a deeper understanding of the textual specifics and methodologies that transcend particular fields. This effort will help refine the platform's ability to accommodate distinct scholarly needs while maintaining the flexibility to address the unique requirements of different academic traditions.

We hope that the platform will prove to be a valuable resource for scholars and play a role in advancing research within the digital humanities.

Acknowledgment

This research has received funding from the European Research Council (ERC) under the European Union's Horizon 2020, Excellent Science (Grant agreement No. 885222).

References

[1] Bradley, John. 2015. "How about Tools for the Whole Range of Scholarly Activities?" In Global Digital Humanities. ADHO. Sydney, Australia: Paul Arthur, University of Western Sidney. https://dh-abstracts.library.virginia.edu/works/2269.

[2] Nantke, Julia, and Frederik Schlupkothen, eds. 2020. Annotations in Scholarly Editions and Research. Functions, Differentiation, Systematization. Berlin, Boston: De Gruyter. https://doi.org/doi:10.1515/9783110689112.

[3] Robinson, Peter. 2017. "Some Principles for Making Collaborative Scholarly Editions in Digital Form." Digital Humanities Quarterly 11 (2). http://www.digitalhumanities.org/dhq/vol/11/2/000293/000293.html.

[4] Parr, Terence. 2014. Language Implementation Patterns. Create Your Own Domain-Specific and General Programming Languages. Pragmatic Bookshelf.

[5] Parr, Terence. 2013. The Definitive ANTLR 4 Reference. 2nd ed. Pragmatic Bookshelf.

[6] Fowler, Martin. 2010. Domain-Specific Languages. Addison-Wesley Signature Series (Fowler). Pearson Education.

[7] Boersma, Meinte. 2021. Domain-Specific Languages Made Easy. Manning Publications Co.

[8] Mugelli, Gloria, Federico Boschetti, Riccardo Del Gratta, Angelo Mario Del Grosso, Fahad Khan, and Andrea Taddei. 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.

[9] Boschetti, Federico, and Angelo Mario Del Grosso. 2020. "L'annotazione di testi storico-letterari al tempo dei social media." Italica Wratislaviensia (Print) 11:65–99. https://doi.org/10.15804/IW.2020.11.1.03.

[10] Zowghi, Didar, and Vincenzo Gervasi. 2003. "On the Interplay between Consistency, Completeness, and Correctness in Requirements Evolution." Information and Software Technology, Eighth International Workshop on Requirements Engineering: Foundation for Software Quality, 45 (14): 993–1009. https://doi.org/10.1016/S0950-5849(03)00100-9.

[11] Buzzetti, Dino. 2002. "Digital Representation and the Text Model." New Literary History 33 (1): 61–88.

[12] Melighetti, Francesco. 2019. "Domain-Specific Languages per Facilitare Annotatori Inesperti Di XML-TEI Nella Codifica Di Documenti Storici." Tesi di laurea triennale, Pisa: Università di Pisa, corso di laurea in Informatica Umanistica.

[13] Mugelli, Gloria, Giulia Re, and Andrea Taddei. 2020. "Annotazione Digitale Di Testi Antichi. Lingue Antiche e Digital Humanities, Tra Ricerca e Didattica." Umanistica Digitale 5 (9): 35–60. https://doi.org/10.6092/issn.2532-8816/9962.

[14] Bambaci, Luigi, and Federico Boschetti. 2020. "Encoding the Critical Apparatus by Domain Specific Languages: The Case of the Hebrew Book of Qohelet." In La Svolta Inevitabile: Sfide e Prospettive per l'Informatica Umanistica., edited by Marco Passarotti, Cristina Marras, and Eleonora Litta. Quaderni Di Umanistica Digitale.

Milano: Università Cattolica del Sacro Cuore.

[15] Zunke, Saurabh, and Veronica D'Souza. 2014. "JSON vs XML: A Comparative Performance Analysis of Data Exchange Formats." IJCSN International Journal of Computer Science and Network 3 (4): 257–61.

[16] Evans, Eric. 2003. Domain-Driven Design: Tackling Complexity In the Heart of Software. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

[17] Sun, Chengzheng, and Clarence Ellis. 1998. "Operational Transformation in Real-Time Group Editors: Issues, Algorithms, and Achievements." In Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, 59–68. CSCW '98. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/289444.289469.

[18] Mehdi, Ahmed-Nacer, Pascal Urso, Valter Balegas, and Nuno Perguiça. 2014. "Merging OT and CRDT Algorithms." In Proceedings of the First Workshop on Principles and Practice of Eventual Consistency, 1–4. PaPEC '14. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2596631.2596636.

[19] Fischer, Franz. 2013. "All Texts Are Equal, But… Textual Plurality and the Critical Text in Digital Scholarly Editions." Variants 10:77–92.

[20] Bozzi, Andrea. 2019. Percorsi Di Linguistica e Di Filologia Computazionali. Edited by Maria Sofia Bozzi and Giacomo Ferrari. Edizioni ETS.

[21] Bordalejo, Barbara, and Peter Robinson. 2015. "A New System for Collaborative Online Creation of Scholarly Editions in Digital Form." In 1st Dixit Convension on Technology, Software, Standards for the Digital Scholarly Edition Workshop. The Hague.

[22] Robinson, Peter. 2016. "The Digital Revolution in Scholarly Editing." Stockholm University Press, November. https://doi.org/10.16993/baj.h.

[23] Zenzaro, Simone, Angelo Mario Del Grosso, Federico Boschetti, and Graziano Ranocchia. 2022. "Verso La Definizione di Criteri per Valutare Soluzioni Di Scholarly Editing Digitale: Il Caso d'uso GreekSchools." In Culture Digitali. Intersezioni, Filosofia, Arti e Media. Università del Salento - AIUCD. https://doi.org/10.6092/unibo/amsacta/6848.

[24] Del Grosso, Angelo Mario, Emiliano Giovannetti, and Simone Marchi. 2017. "The Importance of Being... Object-Oriented: Old Means for New Perspectives in Digital Textual Scholarship." In Advances in Digital Scholarly Editing, 1:269–74.

[25] Siemens, Ray, Meagan Timney, Cara Leitch, Corina Koolen, Alex Garnett, with the ETCL INKE, and PKP Research Groups. 2012. "Toward Modeling the Social Edition: An Approach to Understanding the Electronic Scholarly Edition in the Context of New and Emerging Social Media." Literary and Linguistic Computing 27 (4): 445–61.

[26] Vogeler, Georg. 2019. "The 'Assertive Edition.'" International Journal of Digital Humanities 1 (2): 309–22. https://doi.org/10.1007/s42803-019-00025-5.

[27] Ellis, Clarence A., and Simon John Gibbs. 1989. "Concurrency Control in

Groupware Systems." SIGMOD Rec. 18 (2): 399–407. https://doi.org/10.1145/66926.66963.

[28] Zenzaro, Simone, Angelo Mario Del Grosso, Federico Boschetti, and Graziano Ranocchia. 2022. "Verso La Definizione di Criteri per Valutare Soluzioni Di Scholarly Editing Digitale: Il Caso d'uso GreekSchools." In Culture Digitali. Intersezioni, Filosofia, Arti e Media. Università del Salento - AIUCD. https://doi.org/10.6092/unibo/amsacta/6848.

[29] Zenzaro, Simone, Angelo Mario Del Grosso, Federico Boschetti, and Graziano Ranocchia. 2023. "Ease the Collaboration Making Scholarly Editions: The GreekSchools Case Study." In La Memoria Digitale: XII Convegno Annuale AIUCD, 230–32. Siena: Alma Mater Studiorum-Università di Bologna. https://doi.org/10.6092/unibo/amsacta/7721

[30] Del Grosso, Angelo Mario, Federico Boschetti, Simone Zenzaro, and Graziano Ranocchia. 2023. "GreekSchools: Making Traditional Papyrology Machine Actionable Through Domain-Driven Design." In 2023 7th IEEE Congress on Information Science and Technology (CiSt) (IEEE CiSt'23), edited by Mohammed El Mohajir, Mohammed Al Achhab, Badr Eddine El Mohajir, Bernadetta Kwintiana Ane, and Ismail Jellouli, 6. Agadir - Essaouira, Morocco: IEEE Computer Society.

[31] Parr, Terence. 2014. Language Implementation Patterns. Create Your Own Domain-Specific and General Programming Languages. Pragmatic Bookshelf.

[32] Fielding, Roy Thomas. 2000. "Architectural Styles and the Design of Network-Based Software Architectures." Publication, University of California, Irvine. https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

[33] Del Grosso, Angelo Mario, Davide Albanesi, Emiliano Giovannetti, and Simone Marchi. 2016. "Defining the Core Entities of an Environment for Textual Processing in Literary Computing." In DH2016 Abstracts, 771–75. Kraków.

[34] Retter, Adam, and Erik Siegel. 2014. eXist: A NoSQL Document Database and Application Platform. United States of America: O'Reilly Media.

[35] Del Grosso, Angelo Mario, Simone Zenzaro, Federico Boschetti, and Graziano Ranocchia. 2025. "Bridging Traditional and Digital Papyrology with Domain-Specific Languages: The GreekSchools Case Study." In The Digital Critical Edition of Greek Papyri: Issues, Projects, and Perspectives, edited by Nicola Reggiani, 125–52. Berlin, Boston: De Gruyter. https://doi.org/doi:10.1515/9783111070162-009.

[36] Rosselli Del Turco, Roberto, Giancarlo Buomprisco, Chiara Di Pietro, Julia Kenny, Raffaele Masotti, and Jacopo Pugliese. 2014. "Edition Visualization Technology: A Simple Tool to Visualize TEI-Based Digital Editions." Journal of the Text Encoding Initiative 8. https://doi.org/10.4000/jtei.1077.

[37] Cacioli, Giulia, Chiara Di Pietro, Sara Maenza, Roberto Rosselli Del Turco, and Simone Zenzaro. 2022. "There and Back Again: What to Expect in the next EVT Version." In AIUCD 2022 - Culture Digitali. Intersezioni: Filosofia, Arti, Media. Proceedings Della 11a Conferenza Nazionale, Lecce, 2022, edited by Fabio Ciracì, Giulia Miglietta, and Carola Gatto, 212–17. Lecce: Quaderni di Umanistica Digitale.

[38] Venuti, Martina, Angelo Mario Del Grosso, Federico Boschetti, Luigi Tessarolo, Alessia Prontera, Dylan Bovet, Gianmario Cattaneo, and Valeria Melis. 2023. "La

'Galassia MQDQ': un concetto di filologia tradizionale, digitale, sostenibile." Relations 4 (1): 71–120. https://doi.org/10.30687/mag/2724-3923/2023/07/003.

[39] Dombrowski, Quinn, and Seth Denbo. 2013. "TEI and Project Bamboo." Journal of the Text Encoding Initiative, no. 5. https://doi.org/10.4000/jtei.787.

[40] Dekker, Ronald H., and Gregor Middell. 2011. "Computer-Supported Collation with CollateX: Managing Textual Variance in an Environment with Varying Requirements." Supporting Digital Humanities, 17–18.

[41] Hagel, Stefan. 2007. "The Classical Text Editor. An Attempt to Providing for Both Printed and Digital Editions." In Digital Philology and Medieval Texts, 77–84. http://opac.regesta-imperii.de/id/1327309.

[42] Zenzaro, Simone, Daniele Marotta, and Amos Bertolacci. 2018. "CEED: A Cooperative Web-Based Editor for Critical Editions." AIUCD 2018, 93.

[43] Bodard, Gabriel. 2010. "EpiDoc: Epigraphic Documents in XML for Publication and Interchange." Latin On Stone: Epigraphic Research and Electronic Archives, 101–18.

[44] Hu, Vincent C., D. Richard Kuhn, David F. Ferraiolo, and Jeffrey Voas. 2015. "Attribute-Based Access Control." Computer 48 (2): 85–88. https://doi.org/10.1109/MC.2015.33.

[45] Alshuqayran, Nuha, Nour Ali, and Roger Evans. 2016. "A Systematic Mapping Study in Microservice Architecture." In 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), 44–51. https://doi.org/10.1109/SOCA.2016.15.

[46] Lou, Burnard. 2014. What is the Text Encoding Initiative? Encyclopédie numérique 3. Marseille: OpenEdition Press. http://books.openedition.org/oep/426.

[47] Robinson, Peter M. W. 2022. "An Approach to Complex Texts in Multiple Documents." Digital Scholarship in the Humanities 37 (4): 1179–96. https://doi.org/10.1093/llc/fqab108.

[48] Chagué, Alix, Floriane Chiffoleau, Matthias Gille Levenson, Hugo Scheithauer, and Ariane Pinche. 2024. "Chaînes d'acquisition, de Traitement et de Publication Du Texte." Consortium Ariane - Axe 1. https://hal.science/hal-04734959.

[49] Morlock, Emmanuelle. 2017. "TeiPublisher for EpiDoc." In Visible Words: Digital Epigraphy in a Global Perspective. Providence, United States: John Bodel, Brown University and Michèle Brunet, Université Lumière Lyon 2, HiSoMA. https://shs.hal.science/halshs-01773203.

[50] Cayless, Hugh, and Raffaele Viglianti. 2018. "CETEIcean: TEI in the Browser." In Proceedings of Balisage: The Markup Conference 2018. Vol. 21. Balisage Series on Markup Technologies. https://doi.org/10.4242/BalisageVol21.Cayless01.

[51] Rockwell, Geoffrey, e Stéfan Sinclair. 2016. Hermeneutica: Computer-Assisted Interpretation in the Humanities. Ebook Central. MIT Press.

[52] Tiepmar, Jochen, Christoph Teichmann, Gerhard Heyer, Monica Berti, and Gregory Crane. 2014. "A New Implementation for Canonical Text Services." In Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences,

and Humanities (LaTeCH), Gothenburg, Sweden, edited by Kalliopi Zervanou and Cristina Vertan, 1–8. New York, NY, USA: ACL.

[53] Fusi, Daniele. 2017. Tecnologie informatiche per l'umanista digitale. Roma: Edizioni Nuova Cultura. https://doi.org/10.4458/9323.

[54] Buzzetti, Dino. 2023. "Towards an Operational Approach to Computational Text Analysis." In On Making in the Digital Humanities: The Scholarship of Digital Humanities Development in Honour of John Bradley. UCl Press. https://library.oapen.org/handle/20.500.12657/62237.

[55] Bodard, Gabriel, and Polina Yordanova. 2020. "Publication, Testing and Visualization with EFES: A Tool for All Stages of the EpiDoc XML Editing Process." Studia Universitatis Babeş-Bolyai Digitalia 65 (1): 17–35. https://doi.org/10.24193/subbdigitalia.2020.1.02.

[56] Assael, Yannis, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, and Nando de Freitas. 2022. "Restoring and Attributing Ancient Texts Using Deep Neural Networks." Nature 603 (7900): 280–83. https://doi.org/10.1038/s41586-022-04448-z.

[57] Jurafsky, Daniel, and James H. Martin. 2024. Speech and Language Processing (3rd Edition). Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

[58] Riemenschneider, Frederick, e Anette Frank. 2023. «Exploring Large Language Models for Classical Philology». In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), a cura di Anna Rogers, Jordan Boyd-Graber, e Naoaki Okazaki, 15181–99. Toronto, Canada: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.846.

[59] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, e Kristina Toutanova. 2019. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». arXiv. http://arxiv.org/abs/1810.04805.

[60] Voelter, Markus. Generic Tools, Specific Languages. Delft University of Technology, 2014.

[61] O'Sullivan, James, Michael Pidd, Bridgette Wessels, Órla Murphy, Michael Kurzmeier, Sophie Whittle, Andrew Prescott, et al. Digital Editing and Publishing in the Twenty-First Century. Scottish Universities Press. Scottish Universities Press, 2025. https://books.sup.ac.uk/sup/catalog/download/4/8/49.