Umanistica Digitale – ISSN: 2532-8816 – n. 17, 2024
C. Martignano – *Critical Edition Ontology: a concept model for digital critical editions*
DOI: http://doi.org/10.6092/issn.2532-8816/19469

Critical Edition Ontology: a conceptual model for digital critical editions

Chiara Martignano

Università di Firenze
chiaramartignano@gmail.com

Abstract

Over the last thirty years, In the field of digital philology, several generic IT tools have been developed to produce digital scholarly editions (DSE). However, the majority of DSEs continue to be produced using custom-developed tools. This trend is unsustainable, but it is because generic tools are often less usable, difficult to customize, and unsuitable for the scientific objectives of scholars. The scientific community has been exploring how to create flexible and effective tools for digital philology. Two main responses have emerged in the literature: the need for shared editorial models that represent philologists' *desiderata*, and the adoption of software engineering practices to produce more robust, durable, and easily maintainable tools.

This article presents a modeling strategy that considers both perspectives and is based on the creation of a network of editorial models, each one theoretically framed within the philological approach in which it has emerged. The editorial models, formalized as ontologies, can be applied as conceptual models to determine the logical organization and functioning of a tool for digital philology, thanks to the principles of domain-driven design.

Keywords: conceptual modeling, software development, digital scholarly editions, ontologies, digital philology

Negli ultimi trent'anni, nel campo della filologia digitale, sono stati prodotti diversi strumenti informatici generici per la realizzazione di edizioni scientifiche digitali (ESD). Tuttavia, la maggior parte delle ESD continua a essere prodotta mediante strumenti sviluppati ad hoc per un particolare progetto. Questa tendenza è poco sostenibile, ma dipende dal fatto che gli strumenti generici sono spesso poco usabili, difficili da personalizzare e inadatti agli obiettivi scientifici degli studiosi. La comunità scientifica si è interrogata su come realizzare strumenti generici per la filologia digitale flessibili ed efficaci. Nella letteratura emergono due risposte: da un lato il bisogno di modelli editoriali condivisi che rappresentino i desiderata dei filologi, dall'altro il ricorso alle pratiche dell'ingegneria del software per produrre strumenti robusti, durevoli e facili da mantenere nel tempo.

In questo articolo si presenta una strategia di modellizzazione che mette insieme queste due prospettive e che prevede la realizzazione di una rete di modelli editoriali, ciascuno inquadrato teoricamente rispetto all'ambito della filologia in cui è emerso. I modelli editoriali, formalizzati come ontologie, fungono da modelli concettuali

che, grazie ai principi del domain-driven design, possono essere applicati per determinare l'organizzazione logica e il funzionamento di uno strumento informatico per la filologia digitale.

Parole chiave: modellazione concettuale, sviluppo software, edizioni critiche digitali, ontologie, filologia digitale

Introduction

In this article I present the research I carried out during my PhD, which focuses on how to develop flexible, durable and effective IT tools for digital philology, encouraging scholars to adopt them and produce digital scholarly editions more sustainably. The topic of the research stemmed from my previous working experience as a developer of visualisation tools for DSEs, within the Edition Visualization Technology project.

During the course of my PhD I have developed a modelling strategy, that aims at producing a network of editorial models formalised as ontologies. Each model should be theoretically framed within the philological approach in which it emerged, in order to adequately represent the scientific requirements of philologists. At the same time, each model should be used as a conceptual model or domain model, able to guide the development of IT tools, determining their logical organization and functioning. In this way, the tools will be an expression of the editorial model and considered usable and effective.

Beyond the modelling strategy itself, I have developed a model that represents critical editions, called Critical Edition Ontology, in order to test the feasibility of the strategy and the effectiveness of ontologies both as editorial models for philologists and conceptual models for software developers.

The article is structured as follows. First, I will introduce the topic of my research with an overview of the current generic IT tools for digital philology and the main problems related to them (section 1). Then, I will focus on the main research question, how to develop flexible and effective generic IT tools for digital philology, explaining why we need to produce a network of editorial models that can be used by developers as conceptual models (sections 2 and 3). Following I present the modelling strategy to produce the network of models (section 4), focusing on why ontologies seem like the right choice to formalise the models. After that I describe the Critical Edition Ontology (section 5), highlighting the main aspects of its development process and discussing some of the most interesting modelling problems from a philological point of view. Successively, I show how from an ontology it is possible to derive other kinds of models that can guide the development of a software (section 6). Lastly, I discuss the results my research has reached so far and the possible next steps.

1. An overview of the current IT tools for digital philology

The starting point of my research project was the analysis of the state of the art IT (or computer) tools for digital philology (ITTfDP).[1] By "IT tool for digital philology" I mean an

[1] The concrete result of the analysis is a list of generic IT tools for digital philology. By "generic" I mean tools that are meant to be used for the creation of multiple digital scholarly editions. For each tool have been catalogued the following data: the web site URL, the year of creation or first

application or computer program created in order to be applied in the field of digital philology or largely adopted by the community of digital philologists, which allows to carry out one or more activities for the purpose of making a scholarly digital edition, by helping or substituting the philologist.

By conducting this analysis, I have identified three main categories of ITTfDP: production tools, visualization tools and mixed tools. The first two categories derive from the popularity of the so-called "source-output" [13] model. As a matter of fact, most scholarly digital editions are based on source files (often in XML/TEI format) that contain the data, then the data needs to be visualised and made available to the final reader, hence generating an output. In the past few decades the TEI guidelines have become the *de facto* standard for creating digital scholarly editions (see [13]) and in the meantime a number of new IT tools have been developed in order to facilitate the creation and publication of TEI-based digital editions, like TEI Publisher,[2] Versioning Machine[3] etc. Additionally, other IT tools have been created to computationally carry out the activities that are typically part of the editorial process: for example CollateX[4] and its predecessor for automatic collation, the popular Transkribus[5] and the like for the automatic transcription of primary sources, and so on.

Therefore "Production tools" are tools that help philologists produce digital editions and their data. "Visualization tools" allow philologists to present the data in a human readable format. "Mixed tools" allow users to do both: preparing the data and visualizing the digital edition. Some examples are TextGrid,[6] eLaborate[7] and eScriptorium.[8]

Production and mixed tools can be further classified according to the "traditional" activities of the editorial work they carry out, like the ones mentioned above: transcription of primary

publication, a brief description, the creator, the input and output formats, technical information like the technologies it is based on and the type of usage, whether it is open source, whether it allows to work collaboratively, the license type, example of digital scholarly editions that have used the tool and a list of references to the publications about the tool. The full list is available in [14] and online (https://airtable.com/shrb1dVa8QsJ4P8Ga). It is possible to download the list as a dataset from Zenodo (https://zenodo.org/records/8398700).

[2] TEI Publisher (https://teipublisher.com/index.html) is a software that allows users to publish XML files on the web (mainly according to the TEI standard), generating a website, or to convert documents into other formats, such as PDF and ePUB.

[3] Versioning Machine (http://v-machine.org/) is a framework and web interface that allows users to view and compare multiple versions of the same text (for example the different witnesses of a work) encoded in XML/TEI.

[4] https://collatex.net/.

[5] https://transkribus.eu/Transkribus/.

[6] TextGrid (https://textgrid.de/) is an open source application based on exist-db for editing and publishing TEI-based resources. It is divided into two parts: TextGrid Lab, for collaborative or individual text editing, and TextGrid Repository, which acts as a database and platform for sharing and publishing.

[7] With eLaborate users can upload scans of documents, transcribe and annotate texts, and publish the results online as editions freely accessible to anyone on the web.

[8] eScriptorium (https://escriptorium.openiti.org/) is an open-source web application for the transcription of documents. The platform allows users to carry out the transcription both manually in a collaborative way and automatically applying machine learning algorithms.

sources, collation, etc. However, considering that these tools are computer tools, the TaDiRAH taxonomy[9] can be more effective in order to describe the tasks and activities a tool can carry out from a more technical point of view, for example the conversion between two file formats.

1.1 Generic tools and ad hoc (project-specific) tools

Among IT tools for digital philology another important distinction had already been pointed out a few years ago: the difference between tools that were created for a specific digital edition project ("haute-couture editing" [19]) and tools that aim to be generic and applicable to more digital editions ("prêt-à-porter editing", *ibid.*). In the past years many generic tools have been developed and made available to the community of scholars. During my analysis I have catalogued 57 tools and identified another 40 tools.[10] Although there are many generic tools available, it seems that most digital scholarly editions keep being produced using tools specifically created for them. In the *Catalogue of Digital Editions*[11] a data-field named "infrastructure" is available for 178 records on the 320 total scholarly digital editions catalogued.[12] In this data-field are listed the programming languages, the technologies and the tools used to develop a scholarly digital edition. Among these 178 records, a generic tool is mentioned only in 68 records (corresponding roughly to the 35%), considering not only the ones that are specifically for digital philology, but also very generic databases and frameworks for the development of web sites based on XML, like eXist-db[13] and xMod.[14] Considering the number of DSEs produced per year catalogued in the *Catalogue of Digital Editions* (leaving out the records for which the beginning date is not available), one can observe that there has not been an increase of the usage of generic tools for the development of DSEs. On the contrary, the trend has been quite stable, apart from a peak reached around 2014 and 2015 that may depend on the biases of how and when the catalogue has been filled out.

[9] The Taxonomy of Digital Research Activities in the Humanities (TaDiRAH) is the result of a community-driven initiative and was designed to organize and categorize DH content. The current version of the taxonomy (2.0) was formalised in SKOS and published open-source in 2020. https://vocabs.dariah.eu/tadirah/en/.

[10] A more generic collection of tools for scholars for studying texts is TAPoR (Text Analysis Portal for Research, https://tapor.ca/). Originally focused on text analysis tools, as the name suggests, the collection was later expanded to include tools that work on non-textual data and provide services used by digital humanists, such as publishing tools, GIS tools and communication tools. The collection is available as a database and online platform and currently includes 1687 records. Some of the tools that are specifically designed for digital philology that I have identified are listed in the TAPoR collection as well. Anyhow, the number of records collected in TAPoR suggests that nowaday scholars have available many generic tools that can assist them.

[11] https://dig-ed-cat.acdh.oeaw.ac.at/.

[12] The analysed data derived from the *Catalogue of Digital Editions* was last updated on the 7th of March 2023.

[13] https://exist-db.org/exist/apps/homepage/index.html.

[14] "xMod is an application framework which allows the transformation of a repository of TEI (or other, valid) XML into a finished static or dynamic website of conceivably any size" (https://www.tei-c.org/Vault/MembersMeetings/2008/xmod/index.html). This tool is no longer developed.

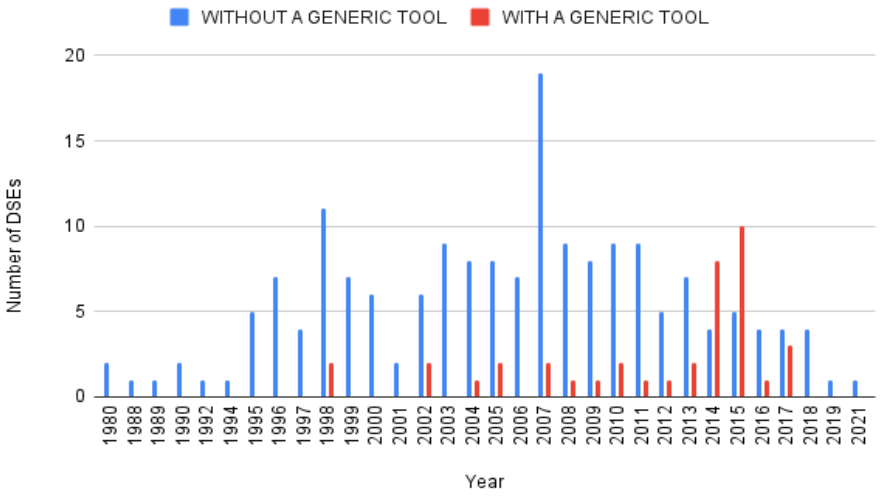

Figure 1 The number of digital scholarly editions implmented and published every year with or without a generic tool.

The first outcome of the analysis of the state of the art IT tools for digital philology is that, despite the high number of generic tools available, these tools are not largely adopted by the scientific community.

1.2 The issues of generic tools

In 2005 Peter Robinson published an article titled *Current issues in making digital editions of medieval texts—or, do electronic scholarly editions have a future?*, ascertaining the presence of a crisis in the field of DSEs due to the lack of usable tools:

> "If digital editions are so manifestly superior, then why indeed are we in the state of affairs described above? Why are so many scholars, and so many scholarly projects, still making print editions? I suggest that the answer to this lies almost wholly in the second aspect to this problem: the availability of usable tools." [20]

Given the number of tools identified, it is possible to affirm that the problem lies not in the lack of tools itself but most likely in the quality of the tools: most potential users do not consider the available tools usable. As pointed out in the aforementioned article *What future for digital scholarly editions? From Haute Couture to Prêt-à-Porter* by Elena Pierazzo [19]] generic tools offer different advantages. For example, they allow users to contain the costs of implementation and maintenance of a DSE and to develop digital scholarly editions faster. However there are many reasons for which a tool can be considered not usable (see [19], [13] and [15]). The main reasons can be emphatically summarised with three "I"s: instability, inaccessibility and inadequacy. The instability depends both on the rapid obsolescence of digital technologies, that require tools to be constantly maintained, and on the circumstances in

which the tools are developed. As a matter of fact, these kinds of tools are often developed by individual researchers and scholars who are not full-time developers. At the same time, development often depends on time-limited funds: when the fundings end, it is difficult to maintain the tools.

Users may find an IT tool for Digital Philology inaccessible because of a high-priced license or because the tool does not come with in-depth documentation or does not offer user support. These shortcomings are quite frequent considering that most of these tools are not developed by big software companies nor large developers' communities.

The most complex problem is inadequacy. Generally speaking, a tool is inadequate when it "cannot do everything required of it" [20], but the inadequacy can manifest itself under different aspects, for example:

- incompatible input and output data formats;
- the way the data are processed by the IT tool;
- the features offered by the IT tool;
- the way the data is visualised and presented to the final user.

The technical, graphical and logical requirements listed above are usually determined when a software (or, more generally, an IT tool) is designed according to the needs of the customers or stakeholders that have commissioned its development. In the case of IT tools for digital philology, who determines the requirements? On which grounds?

Ad hoc tools, i.e tools that are developed within a specific digital edition project, reflect the scientific needs and peculiarities of that project. On the contrary, in the case of generic tools the developers establish the requirements. If the developers are also scholars, it should be easier for them to identify the requirements and scientific needs of potential users. If the developers are not scholars, then they need to work collaboratively with "domain experts", i.e. the scholars, that can help them identify the requirements. However creating generic tools that easily adapt to different digital scholarly editions is a very difficult task.

So the second outcome of the analysis of the state of the art IT tools for digital philology is that: generic ITTfDP struggle to stand out, because of the three "I"s: instability; inaccessibility and inadequacy. To make up for the three I's of the IT tools for digital philology that are already available, new tools keep being developed. But this practice, alongside the trend to develop most DSEs with *ad hoc* software, is making digital scholarly editing not sustainable.

So the initial research question is, how to develop more adequate, accessible and stable digital tools for digital philology, in order to encourage the implementation of DSEs through reusable software.

2. The lack of a common model

In various publications different scholars have reported that the reason behind the production of instable, inaccessible and inadequate software is the lack of a shared model that indicates how ITTfDP should be developed.

> "Despite increasing theoretical awareness, the tools of representation and analysis produced so far have not satisfactorily addressed the problem of the specificity and complexity of the cultural and literary studies domain. In fact, the intellectual investment in the definition of new models and languages for the formal representation and processing of complex cultural objects has been rather low. Most commonly we have inherited and applied models and languages developed in computer science for different domains and necessities." [3]

> "Scholarly editing has been a leader in the use of computing in the humanities and yet many of its tools are borrowed or adapted from applications that were designed for other purposes. This is good to the extent that adaptations built on the shoulders of previous efforts have saved time and provided desired results, but it is bad when off-the-shelf solutions result in compromises, not exactly what the scholarly editor wanted, even though they might help a project meet a funding deadline. The call for tools designed and developed for specific purposes in scholarly editing has been already made by many. [...] However, while varieties of interesting projects sporting new tools and designs are necessary to explore the options and possibilities that can be developed, we lack a comprehensive and thoughtful set of principles to define the characteristics that our tools, content, displays, and environments do or should incorporate." [23]

> "Although the digital turn of textual scholarship is nowadays a reality and many advancements have been made in encoding and visualizing textual resources, flexible and shared models in the construction of tools for scholarly editing are still missing. This lack typically leads to the development of ad hoc – i.e. not reusable software." [6]

> "In order to be able to offer a stable infrastructure for digital editions it is therefore necessary to reflect on the features that the scholarly community will consider essential to a particular type of text or scholarly problem and to agree on some essential models which take into account the new affordances offered by the digital." [19]

In the past few years some scholars have tried to propose models to guide the development of IT tools for digital philology.

A sort of generic "model", more precisely a set of principles, was elaborated within the Humanities Research Infrastructure and Tools (HRIT) project, of the Textual Studies and Digital Humanities center of the Loyola University (Chicago), coordinated by Peter Shillingsburg. The HRIT project has developed an infrastructure of IT tools for the creation of digital textual archives and scholarly editions. The website of the project offers an actual handbook for developers that explains how to use and further develop the tools that are part of the infrastructure. At the same time, the handbook enunciates the principles that have guided the creation of the infrastructure itself. There are four principles, but only the third one specifically concerns software development, exhorting to adopt a modular approach when designing and organising software components.

Other models were developed for a particular type of production tools, for example the well-known "Gothenburg model", elaborated by the developers of the automatic collation softwares CollateX and Juxta. Whereas in the field of automatic transcription it has been recently developed the model SegmOnto [10], a controlled vocabulary that defines the contents of a handwritten or printed book, in order to homogenize the data required in input by layout analysis tools.

Even though some attempts have been made, a common model for the development of IT tools for digital philology is still missing or at least none of the models produced so far has managed to become a standard within the scientific community.

The solution to the problem of the inadequacy and more generally the little usability of generic tools for digital philology could be the creation of such a model. But it is vital to understand how this model exactly should be in order to be of any help.

Concerning this point, in the literature emerge two different perspectives that complement each other: the philologist's perspective and the developer's perspective.

The first one can be summarized as stated in [19]:

> "What most tools have in common is the fact that their development seems to have been generated from data models (how a file is encoded), from specific projects, and from specific requests made by early users, more than from **an effort to assess the editorial models needed by the scholarly community**. Some of these tools are very specialised and only cater for a very specialised data model (like the Versioning Machine), while others are more adaptive (like the TEI Publisher), but what is missing is **the proposal of a scholarly model which is theoretically and editorially based**. The risk is that the availability of these (and of course any other) tools, because of the paucity of such tools overall, will transform them into de facto editorial models even when they lack solid scholarly backing".

In other words, scholars need a model that represents their scientific needs, starting from the editorial models and methodologies that have been so far defined within the different fields of philology.

The second one can be summarized as stated in [6]:

> "The development of applications in the field of Digital Humanities (DH) does not adequately take into account domain modelling, software design principles and software engineering methodologies. In fact, many systems developed in the context of DH-related projects have not been conceived to be modular, extensible, and scalable: they only tend to solve specific problems such as data-driven and project-oriented tools. In addition, most projects focus on the requirements of humanists (as end users), but leave out the needs of software developers".

That is, to be able to guide the development of an IT tool, the model must apply the best practices of software design.

Both perspectives need to be taken into account in order to solve the problem: the model must represent the scholars' requirements effectively and at the same time it must be developed according to shared software design principles and methodologies.

3. A conceptual model

Once the overall goal of the research was set, the first research question was how an editorial model can be able to represent the scientific requirements of the philologist and at the same

time guide the development of an IT tool, while satisfying the technical requirements of software design. What kind of model can satisfy both needs?

To answer that question, it felt appropriate to consider and evaluate the different models that are used in the field of software engineering. Also in this scientific field, there are many kinds of models that vary depending on the aspect of the system they represent, on the abstraction level of the representation and on the main objective they are intended for. Generally speaking, there are three main types of models that constitute three different levels of abstraction in the representation of a system [17]:

- "Business or domain models – models of the actual people, places, things, and laws of a domain. The "instances" of these models are "real things", not representations of those things in an information system [...]
- Logical system models – models of the way the components of a system interact with each other, with people and with organizations to assist an organization or community in achieving its goals
- Implementation models – the way in which a particular system or subsystem is implemented such that it carries out its functions. Implementation models are typically tied to a particular implementation technology or platform."

The first type of model seemed to be the most suitable type of model for our goal, for two reasons. First, it allows one to define the domain, i.e. the application area of the IT tool, on an abstract level starting with the definition of the objects that scholars manipulate and produce, for example witnesses, transcriptions, texts, etc., and the processes they apply. Secondly, an abstract model is more durable and flexible, as it can be used as the starting point to implement an IT tool with different technologies and platforms. As a matter of fact, from a domain model it is possible to derive the other two kinds of models that guide the development of a software, applying the principles of a software philosophy called Domain Driven Design (DDD) [9]. This software design approach is founded on the idea that by analysing thoroughly the domain of application of a software it is easier to develop it and maintain it. When applying the DDD approach, developers and domain experts analyse the domain together and create a model, "a system of abstractions that describes selected aspects of a domain and can be used to solve problems related to that domain" [8]. In other words, it is a model that represents the domain and at the same time determines the logic and functioning of the software: what data is managed in the software, what are the different logical components, how the components interact with each other, etc. This means that the collaboration between developers and domain experts concretely determines how the code is written, not just the expected final result.

This collaboration leads first to the definition of a common vocabulary and the creation of a language, which reflects the domain, its problems and its functioning. This language is the bridge between the design phase and the development phase: when the terms established during the design are also used in writing the actual code, the code is understandable to all members of the group and is easier to evaluate the consistency between the code and the domain. This language is called "ubiquitous", as it is present in all phases of work and used by all members of the development team.

In summary, it can be said that the DDD approach requires that, by analyzing the domain, domain experts and developers create a model together and define a common language to be

used in all phases of software development, from its design to writing the code. Therefore, the software code becomes an expression of the model.

The kind of model that could solve the problem described in the previous section is a "domain model". It could also be called more generally "conceptual model",[15] as both types of models are abstract and meant to identify the concepts that distinguish a given domain. From this moment on we will use the term "conceptual model".

4. A common modelling strategy

In [19] the author states that we need "editorial models", plural, framed into the disciplinary context they were found. This means that we need as many models as the number of editorial models produced so far in the different subfields of philology and as the number of tasks an IT tool for digital philology should carry out. Therefore, instead of producing one big model that condenses all the different editorial models and technical tasks, the aim of the research should be to establish a common modelling strategy. A modelling strategy that sets the common ground on which the scholarly community can collaboratively create a network of conceptual models able to guide the development of IT tools for digital philology.

After choosing conceptual models (or domain models) as the type of models that can effectively represent scientific needs and determine technical requirements, another aspect needs to be considered, to establish the modelling strategy: the modelling language one should use.

According to the principles of the MDA [17], the modelling language has to be standard, platform-independent and open, so that the model can be durable and easily applied for the implementation of tools that are based on different technologies.

4.1 The choice of developing the conceptual models as ontologies

A viable solution is to develop the conceptual models as ontologies,[16] for two main reasons. First of all, ontologies can be considered as "philologist-friendly"(see [4]). An ontology is based on a few key concepts, such as "class", "attribute" and the "child-parent relationship", which are easily understandable also by people who do not have advanced ontology skills. Secondly, in an ontology it is possible to provide for each class and property labels and definitions expressed in natural language, so as to firmly anchor the model to its theoretical basis. The labels can correspond to the technical terms used in that specific philological field in different

[15] In support of this terminological choice consider these official definitions given in the ISO-standard Systems and Software Engineering Vocabulary [12]: "a conceptual model is a model of the concepts relevant to some endeavour" and "the domain model describes the commonalities and variabilities among requirements for software systems in the domain".

[16] "In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their mean-ing and constraints on their logically consistent application." (Ciotti e Tomasi 2016, p. 4)

languages, while the definitions can also explicitly include bibliographical references. In this way the domain experts, i.e. the philologists, can understand the model with ease.

As an example, we report an extract of the ontology Scholarly Editing,[17] where the philological concept of "Reading" is defined and also presented with the translations of the term in multiple languages.

```
scholarly-editing:Reading

    rdfs:isDefinedBy <http://e-
editiones.ch/ontology/scholarly-editing#>;

    a rdfs:Class;

    rdfs:label "reading"@en, "Lesung"@de,
"lezione"@it, "lectio"@la, "lecture"@fr,
"lettura"@it;

    rdfs:comment """Copy of one or more words or
phrases of an original or witness."""@en;

    rdfs:subClassOf scholarly-
editing:EditorialStructure, text:CopiedText.
```

Listing 1 A snippet of the Scholalrly Editing Ontology showing the definition of the class "Reading".

Secondly, ontologies encourage the connection between different models and the reuse of existing ontologies. So the application of ontologies would help create a network of conceptual models. Some key concepts would be shared by different models, for example the concept of edition, witness, work, etc. Other concepts could be peculiar to one editorial model. Furthermore, thanks to the "Open World Assumption",[18] the adoption of ontologies would allow to progressively define different editorial models with increasing in-depth information.

Ontologies can be expressed through different standard languages (RDF, Turtle, etc.). For this research I have decided to use OWL 2 [16]. OWL (Web Ontology Language) is a language for defining web ontologies and standards maintained by the W3C. The first version of OWL (OWL 1) was published in 2004, while in 2012 the second version (OWL 2) was published, which consists of an extension and revision of the first, with the aim of facilitating the development and the sharing of ontologies across the web. The main difference between the two versions is that in OWL 2 an ontology and each element within it must be identified using IRIs (Internationalized Resource Identifiers). In this way, both the ontology in its entirety and a single element are referable from the outside, facilitating the sharing and reuse of different ontologies across the web. For this reason OWL 2 was identified as the most suitable modeling language for the purposes of this research, also considering that it is a very expressive language and has been widely used by the scientific community.

[17] https://github.com/nie-ine/Ontologies/blob/master/Nie-ontologies/Generic-ontologies/scholarly-editing.ttl.

[18] "The "Open World Assumption" is a term from knowledge base systems. It characterizes knowledge base systems that assume the information stored is incomplete relative to the universe of discourse they intend to describe. This incompleteness may be due to the inability of the maintainer to provide sufficient information or due to more fundamental problems of cognition in the system's domain" [1].

4.2 The limitations of ontologies

A further aspect needs to be considered about the expression of conceptual models as ontologies: how easy and effective it is to integrate a conceptual model expressed in the form of an ontology into a software development process? The theoretical premises for conducting experiments in this sense are there. As a matter of fact, in the field of software engineering there is a development approach called Ontology-Driven Software Development (ODSD, see [18]), which aims at integrating ontologies in the different phases of software production. In literature there are IT experts who have experimented with this so-called ODSD. However this approach is not a major trend in software engineering. Using an ontology to express the conceptual model may be limiting and impractical for many software developers. In the field of software engineering the best known and most used modeling language is UML. On the other hand, UML[19] would be less accessible to philologists. For these reasons a good compromise could be to use ontologies, expressed in OWL 2, as the starting point for the modelling activity. Then, once the ontologies are set, the conceptual models can be translated into UML diagrams. These diagrams would be oriented to the software developer and the final product of the modelling activity.

To sum up, the modelling strategy consists of the creation of a network of conceptual models that represent the various editorial models defined so far in the field of philology, and that guide the development of IT tools for digital philology able to undertake the different tasks of the editorial process. The modelling strategy is founded on the close collaboration between philologists and software developers, to analyse and model the domain, taking into account both the scientific and the technical requirements. Finally, the conceptual models should be developed as ontologies at first and later used to produce UML diagrams that guide the development of an IT tool on a logical and implementation level.

5. The Critical Edition Ontology (CEO)

After defining the technical premises for the modelling strategy, the next step of my research was to start testing its effectiveness, by creating a conceptual model. I have created a first test conceptual model, called the Critical Edition Ontology. As the name suggests, this ontology describes critical editions and the main goal of the CEO is to guide the development of a visualisation tool, by defining the components that constitute a critical edition and the logical relationships between them. I have set this goal for the test conceptual model, drawing from my professional expertise as a web developer of visualisation tools for digital scholarly editions.

By "critical edition", which is the core concept of the model, I mean the edition of a work, mostly of a literary work, produced by a scholar (or a group of scholars) by taking into account

[19] UML (Unified Modeling Language) is a standard visual modeling language used to design, describe, and analyze computer systems.

In concrete terms, UML models are diagrams that can be used in a similar way to the floor plan of a building: they indicate how the application should be created, before starting to write the code.

UML was created in the 1990s by three computer developers, Grady Booch, Ivar Jacobson and Jim Rumbaugh. In 1997, version 1.1 of UML became a standard adopted by the Object Management Group, an organization that still develops and maintains it. The most recent version of UML, as well as the one taken as reference in this research work, is 2.5.1.1 published in 2017.

the textual transmission of the work. Usually, the main objective of a critical edition is to assess the original form (or multiple forms) of a literary work when possible, i.e. critical editions primarily focus on a reconstructive intent.[20] I have picked this particular domain for the test model because of my academic background in romance philology and because critical editions tend to be less popular than non-reconstructive editions (for example, diplomatic or interpretative editions) on the web, despite the fact that they keep being produced in the print format (see [20] and [22]). Therefore, the ultimate objective of the CEO is guiding the production of generic visualisation tools that can meet the scientific needs of that part of the scientific community that prefers to publish its own editions in print or with *ad hoc* IT tools.

The test conceptual model has been developed as an OWL 2 ontology that comprehends three main parts:

- a generic part that frames and describes the main concepts and that can be shared with other editorial models in order to create a network of conceptual models, such as the concepts of work, witness, textual transmission, etc.
- a part that describes the relationship between the critical text and the textual transmission of the work;
- a part that analyses the components of a critical edition. This last part is the one that mostly aims at guiding the development of the visualisation tool.

5.1 Relationship with existing ontologies

To develop the ontology, I have used several existing ontologies. The CIDOC-CRM[21] and its compatible model LRMOO[22] have been used to frame and define the main concepts, such as Witness and Work, for two main reasons: first, the definition of wide and complex concepts such as that of work is very difficult. Reinventing the wheel trying to give a new definition seemed risky, especially considering that the international scientific community had already agreed on the definitions given within the Library Reference Model (formerly F*unctional Requirements for Bibliographic Records*).[23] The second reason is that, by framing the main concepts of my model within wider concepts defined by the CIDOC-CRM and one of its compatible models it is possible to propose the critical edition ontology as another compatible model of

[20] "La conclusione naturale di uno studio di critica testuale è la realizzazione di un'edi-zione critica. Si tratta di un'edizione 'scientifica' dell'opera, tale cioè che possa essere utilizzata dal lettore come testo 'ufficiale' e affidabile, e nella quale si affrontino i pro-blemi posti da quello specifico testo in ordine al suo stato di conservazione e, ove ne-cessario, alla sua ricostruzione. Essa può consistere nella riproduzione dell'originale, se è conservato; o in un'ipotesi di ricostruzione dell'originale, se esso non è conservato; o ancora nella pubblicazione comparativa di testi diversi, ognuno dei quali gode della qualifica di originale o è comunque importante ad illustrare il processo di sviluppo te-stuale" [2].

[21] The Conceptual Reference Model developed by the CIDOC is one of the most known and used ontologies in the field of cultural heritage (https://cidoc-crm.org/).

[22] LRMoo (https://cidoc-crm.org/frbroo/) is an ontology that represents the Library Reference Model (LRM) developed by the IFLA (International Federation of Library Associations and Institutions), a conceptual model that aims to make explicit the general principles that govern the logical structure of bibliographic information. LRMOO is one of the compatible models of the CIDOC-CRM.

[23] See previous note.

the CIDOC-CRM. Although the current version of the ontology is not stable enough, in the long term the compatibility with CIDOC-CRM and LRMoo would help preserve the model effectively and propose it as a standard.

Other two ontologies that are more strictly "philological" have played a fundamental role in the development of the CEO:

- the Critical Apparatus Ontology (CAO),[24] developed by Francesca Giovannetti and based on the TEI guidelines, it represents the apparatus of textual variants and the philological arguments that accompany a digital critical edition (see [11]);
- and Scholarly Editing,[25] an ontology developed by Hans Cools and Roberta Padlina, within the NIE-INE framework, whose main objective was to use semantic web technologies to formally express critical editions, so as to allow interoperability between different edition projects and, in the long term, create "a semantic space for an interdisciplinary formal knowledge domain for the Humanities" [5].

Both ontologies were used as a starting point to find the best modeling solutions in order to model the domain of critical editions. Some of the classes and properties contained in both ontologies that fit the purposes and the scientific requirements of the CEO have been included, while others have been discarded.

The DoCO ontology[26] has been used as a basis for comparison on how the components of a critical edition should be identified and described.

Lastly, other ontologies have been used for compatibility reasons, as they are used in the CAO ontology, in particular the HiCO,[27] the PROV-O[28] and the Web Annotation Ontology.[29]

5.2 How the ontology is theoretically founded

In order to model the domain of the CEO and develop a "ubiquitous language" (see above [3]), it was fundamental to analyse: how philologists work in order to create critical editions; which objects they manipulate; which technical terms they use. For this reason, two main kinds of sources have been used. On one hand, handbooks of critical editing and stemmatology, with a focus on the handbooks published in Italy in the field of romance philology, that illustrate with real examples how editors work in order to establish the critical text and prepare a critical edition (for example).[30] On the other hand, glossaries of philology, in particular the Lexicon of

[24] https://w3id.org/cao.

[25] https://web.archive.org/web/20210614221418/https:/e-editiones.ch/ontology/scholarly-editing.

[26] DoCO (Document Components Ontology, http://www.sparontologies.net/ontologies/doco) is an ontology that represents the components of a document, both structural (for example, chapter, paragraph, etc.) and rhetorical (for example, introduction, acknowledgements, appendix, etc.). DoCO is one of the SPAR ontologies (http://www.sparontologies.net/).

[27] https://marilenadaquino.github.io/hico/.

[28] https://www.w3.org/TR/prov-o/.

[29] https://www.w3.org/ns/oa#.

[30] Some of the manuals used are for example: *Filologia romanza. Vol. 1: Critica del testo* by Lino Leonardi, published in 2022; *A che serve un'edizione critica?: leggere i testi della letteratura romanza medievale*

Scholarly Editing [7] and the Parvum Lexicon Stemmatologicum [21], which have been crucial to decide how to name the classes and properties of the ontology.

On a few occasions the TEI guidelines have been considered as well, in particular the Critical Apparatus module, as they represent a fundamental reference for scholarly editing and offer a set of technical terms and definitions used in the field of philology.

5.3 Problematic concepts and modelling solutions

As mentioned above, the CEO consists of three different parts. A more general part, that introduces the main concepts of the domain, that can be shared with other future editorial models. A part that focuses on the relationship between the edited text and the witnesses, allowing to describe in a detailed manner how the editor has prepared the critical text. And last but not least, a part that identifies the components of a critical edition, in order to guide the development of a visualisation tool.

A complete description of the classes and properties included in the CEO is beyond the scope of this article. However, in this section I present for each part of the ontology a problem I had to solve and the design solution I adopted, as to briefly showcase the way I have worked. The ontology is available in its entirety online.[31] Furthermore in an appendix to my doctoral thesis [14], it is possible to consult the bibliographic references used to establish the definition and name of each class.[32]

5.3.1 The relationship between critical and scholarly editions

The core concept on which the whole ontology is based is that of "critical edition". For this reason, the concept of "critical edition" had to be accurately framed and defined.

Among the existing ontologies, Scholarly Editing offered a class named "Critical Edition" defined as follows: "Edition to which a critical editor added evidence and interpretation of a text creation and historical transmission, using a critical apparatus, allowing readers to assess the editor's choice." The concept expressed by this class is part of the wider concept of "Edition", defined as "Derived text expression as scholarly edited text" and labeled in english as "scholarly edition".

```
<rdfs:Class rdf:about="http://e-
editiones.ch/ontology/scholarly-editing#Edition">

    <rdfs:label xml:lang="fr">édition
scientifique</rdfs:label>

   <rdfs:subClassOf rdf:resource="http://e-
```

by Pietro Beltrami, published in 2010; *L'edizione critica dei testi volgari* by Franca Brambilla-Ageno, published in 1984.

[31] The ontology is available on GitHub (https://github.com/ChiaraMartignano/critical-edition-ontology/blob/main/docs/Critical_Edition_Ontology_rdf). In order to consult it in a more user-friendly way, it is also possible to visualise it with the tool LODE: purl.org/critical-edition-ontology/lode.

[32] As the CEO aims at describing the domain model of a software for visualizing digital critical editions, it is mostly composed of classes that represent the main concepts of this particular domain and by properties that connect these concepts. The current version of the CEO provides a few attributes as well. However, in the next steps of the research, new elements could be added to the ontology, in order to describe the domain in a more detailed and effective way.

```
editiones.ch/ontology/text-
editing#DerivedTextExpression"/>

     <rdfs:label xml:lang="de">wissenschaftliche
 Edition</rdfs:label>

     <rdfs:label xml:lang="en">scholarly
 edition</rdfs:label>

     <rdfs:comment xml:lang="en">Derived text
 expression as scholarly edited text.</rdfs:comment>

     <rdfs:isDefinedBy rdf:resource="http://e-
 editiones.ch/ontology/scholarly-editing#"/>

     <rdfs:subClassOf rdf:resource="http://e-
 editiones.ch/ontology/scholarly-editing#EditedText"/>

     <rdfs:label xml:lang="it">edizione
 accademica</rdfs:label>

   </rdfs:Class>
```

Listing 2 A snippet of the Scholalrly Editing Ontology showing the definition of the class "Edition".

Therefore, in the Scholarly Editing ontology critical editions are one kind of many different editions, each represented by a class, as shown in the figure below.

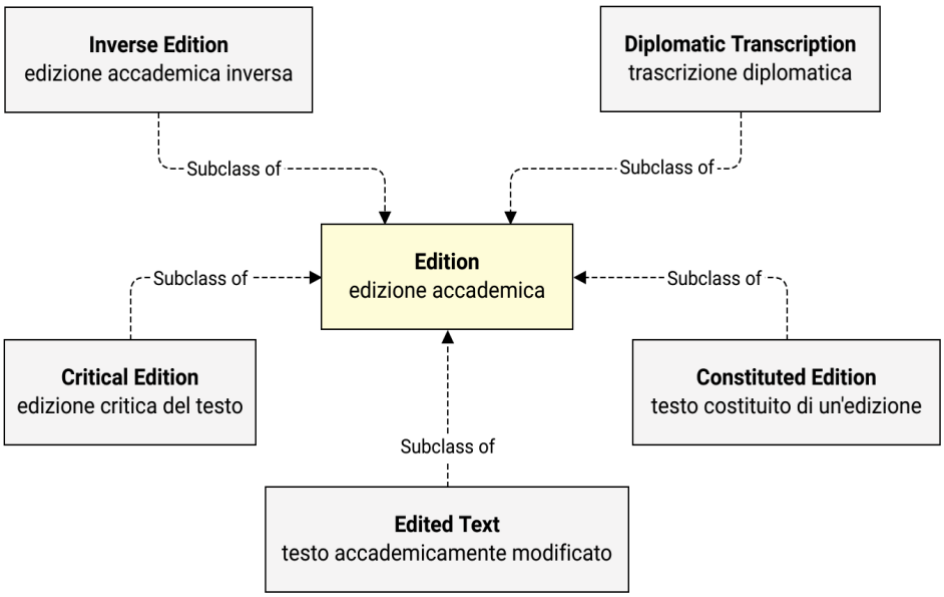

Figure 2 Diagram of the class "Edition" of the Scholarly Editing ontology and its subclasses.

The definition of the class "Edition" refers to the concept of "text expression", which is one of the main concepts of the LRMOO, `F2 Expression`. A critical edition, and more in

general an edition, can be defined on a more abstract semantic level, as one of the expressions of a work.

For these reasons, in the CEO the concept of critical edition is represented by a native class, Critical Edition, which is a subclass of the native class Scholarly Edition, following the example of the Scholarly Editing ontology. In this way, the class Scholarly Edition can be the starting point for the modelling of the other editorial models. Also, the main concepts presented in the domain, like those expressed by the classes Witness, Work, Edited Text, and Accompanying Material, can be linked to the more general Scholarly Edition class, setting a common ground for the future models. Lastly, Scholarly Edition is a subclass of the LRMOO class `F2 Expression`.

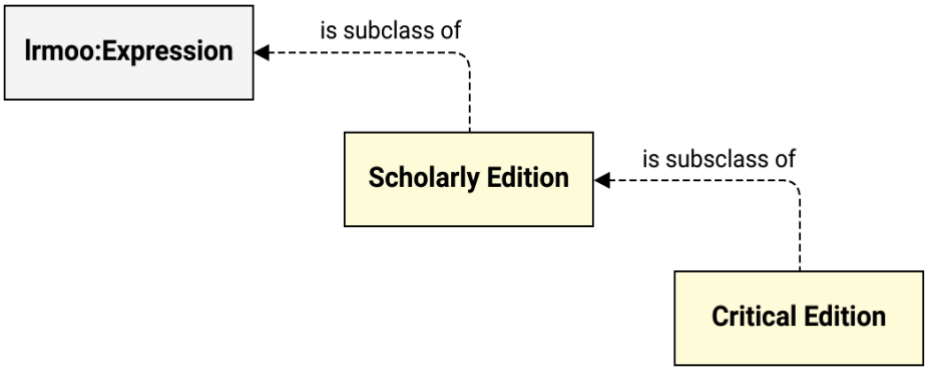

Figure 3 Diagram of the class Critical Edition and its superclasses in the CEO.

When analysing the language of domain experts, the distinction between "critical edition" and "scholarly edition" sometimes may not be so clear.

Consider this definition of "critical edition":

> "La conclusione naturale di uno studio di critica testuale è la realizzazione di un'edizione critica. Si tratta di un'edizione 'scientifica' dell'opera, tale cioè che possa essere utilizzata dal lettore come testo 'ufficiale' e affidabile, e nella quale si affrontino i problemi posti da quello specifico testo in ordine al suo stato di conservazione e, ove necessario, alla sua ricostruzione. Essa può consistere nella riproduzione dell'originale, se è conservato; o in un'ipotesi di ricostruzione dell'originale, se esso non è conservato; o ancora nella pubblicazione comparativa di testi diversi, ognuno dei quali gode della qualifica di originale o è comunque importante ad illustrare il processo di sviluppo testuale" [2].

In the Italian context a critical edition varies depending on the characteristics of the textual transmission of the edited work: it can be a reproduction of the original, or its reconstruction or the publication of two or more texts in a comparative view. The concept of critical edition so defined may comprehend other editorial models, for example the diplomatic edition, the interpretative edition and the comparative edition. But the prerequisites of this definition of critical edition are that the philologist (1) considers the whole textual transmission of the work as a process and (2) aims at representing the original form (or forms) of the work in the best way possible.

However, editorial models other than the critical edition are often chosen for other purposes than the reconstruction of the original form(s) of the work. For example, diplomatic/interpretative transcriptions of manuscripts that are particularly relevant on their own.

Moreover, in an international context, the term "critical edition" is mostly used to identify editions that are based on multiple sources and have a clear reconstructive intent:

> "A critical edition is an edition in which the text has been constituted on the basis of more than one source according to the genealogical principle. [...] What sets the critical edition apart is that it is based on a strict recension of the manuscript sources and the attempt to edit the archetypal text (possibly with some changes where the archetype is clearly faulty), and it is thus closely associated with the Lachmannian tradition of textual editing. Editors who do not agree with this reconstructive tradition, usually refer to their editions in other terms than critical" ("edition, critical" in [21]).

While the term "scholarly edition" is used to identify different types of editions:

> "Scholarly editing […] is used to describe any sort of edition prepared by a person claiming to be a scholar. […] The term should be reserved for editorial efforts designed to make available for scholarly use works not ordinarily available or available only in corrupt or inadequate forms" [24].

In conclusion, even if the model is theoretically grounded into the philological practice and theory developed in Italy over the course of the past century, it seemed more appropriate to distinguish the two concepts clearly and to take into account the different uses of the two terms on an international level. In this way, the model should be more flexible and usable in various contexts.

5.3.2 Reading: an umbrella-term

If the definition of a "ubiquitous language" starts with the analysis of how domain experts, in this case philologists, talk, the term "reading" is one of the most fascinating and complex to analyse. When asking a philologist what a reading is, the answer would most likely be something like "what one reads within a witness" (see definitions in [2] and [21]). However, the term reading is used to identify various concepts that on a semantic level slightly differ, for example variant reading, the reading presented in a critical apparatus, the reading of the copyist, conjectural reading, etc. The main ambiguity behind the concept of reading is that between reading as the object of the interpretation and reading as the result of the interpretation. One could say that a reading is a matter of fact, a concrete piece of data that can be analysed with objectiveness. At the same time, one could assert that a reading is in its nature the result of an interpretation, either by the copyist in the process of copying the text or by the editor in the process of setting the critical text.

In the Scholarly Editing ontology, as shown above, reading is defined as "Copy of one or more words or phrases of an original or witness." While in the CAO, there is a clear distinction between the sequence of characters or words within a witness that is the object of the interpretation and the result of the interpretation that the editor carries out on that sequence (or more sequences, in case of works transmitted in multiple witnesses). The first concept is

represented with the class `F23 Expression Fragment`[33] of the FRBROO ontology,[34] while the second one with the class `cao:Reading`,[35] defined as follows: "A reading is a scholarly claim about or interpretation of a particular textual fragment contained in a witness. Multiple readings may be given for the same textual fragment."

The CEO proposes again this distinction between object of the interpretation and result of the interpretation. But to better reflect how philologists talk and at the same time model the components of a critical edition, two native classes have been created:

- Reading, which identifies the object of the interpretation and is defined as follows: "a sequence of characters, words, or phrases that can be read in a preserved or lost witness";
- Reading In Apparatus, which is equivalent to the concept expressed by the class Reading of the CAO and is defined as the "reading presented in a critical apparatus entry. A reading in the critical apparatus may correspond to two or more readings of different witnesses. The reading in the critical apparatus can be presented in a diplomatic (according to a particular witness) or interpretative form."

5.3.3 How to describe the components of a critical edition

The last part of the model identifies and describes the components of a critical edition. Given that the goal of the CEO is to guide the development of a generic visualisation tool, the components of the critical edition should correspond to the components of the visualisation tool as well. So, the main problem in the modelling of this last part was how to describe components in an effective way both for philologists and software developers.

Two different aspects had to be considered. On one hand, most of the digital critical editions developed so far propose again the components that are typically found in printed critical editions. In other words, there is a continuity between printed editorial products and digital editions, not only on a graphical level, but also on a logical level. The TEI guidelines, for example, include elements that clearly refer to graphical components of printed editions, like <back> and <front>. On the other hand, defining the components depending on their graphical aspect on a printed page (like in the DoCO ontology) could limit the usability of the model and compromise its longevity, not allowing developers to implement the components in innovative ways.

The solution was met halfway: the model includes concepts that were born in the context of printed editions, but each component is described focusing not on its graphical aspect, but on its functionality. For example, the concept of "apparatus" is represented by an homonymous native class, defined as: "Component of a scholarly edition that accompanies the edited text and consists of a set of annotations that the editor has appended to edited text for a par-ticular scientific purpose."

[33] "This class comprises parts of Expressions and these parts are not Self-contained Expressions themselves." (https://www.iflastandards.info/fr/frbr/frbroo#F23).

[34] https://www.iflastandards.info/fr/frbr/frbroo.html.

[35] https://fgiovannetti.github.io/cao/#Reading.

6. The conversion from OWL into UML

The second step in the modeling strategy is to use the conceptual model that represents a particular editorial model to design the structure of an IT tool and its features, according to the principles of domain-driven design [9]. The idea is to derive "logical system models" and "implementation models" [17] (see section 3) from the conceptual model (or "domain model"). This derivation should be executed after having tested and fine-tuned the conceptual model and after having established the technologies that need to be used to develop the tool.

The translation from OWL to UML is a delicate transition, which must be carried out by computer developers together with the philologists (the domain experts), to ensure that the "semantics" of the model, i.e. what the model "means" or "expresses", is not modified. Ideally, the goal of this phase of the modeling strategy should be the development of morphisms, i.e. precise indications of how the ontologies should be translated into UML diagrams. In this way the implementation models could be derived directly from a conceptual model in UML, rather than from the ontology in OWL, which in any case reflects the editorial model expressed by the ontology.

On a more concrete level, how should the conversion from OWL to UML be carried out? The first step is the creation of UML class diagrams. As an ontology is based mainly on a set of classes and relationships, the classes can be easily converted into UML classes, while the relationships in UML can become different kinds of associations.

As an example of this operation, consider the part of the Critical Edition Ontology that describes the concepts of Apparatus and Apparatus Entry and their subclasses, respectively Critical Apparatus and Commentary and Critical Apparatus Entry and Note.

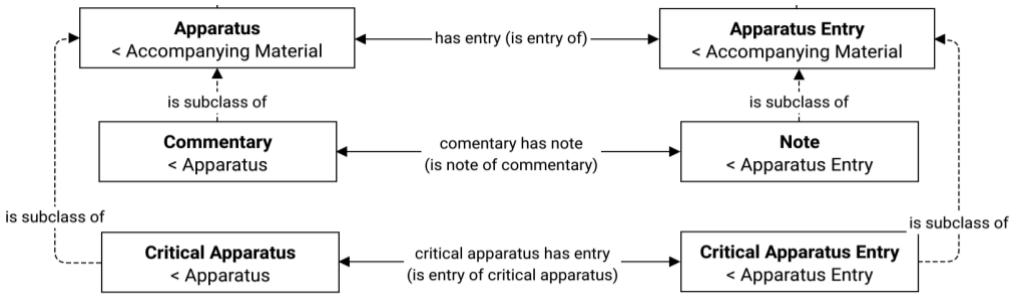

Figure 4 Diagram of the Apparatus and Apparatus Entry classes and their subclasses in the CEO.

The six OWL classes can be converted into the same amount of UML classes. To describe the relationship between the more general classes (Apparatus and Apparatus Entry) and their specifications, in UML we can use the type of association called "generalization". Visually, generalizations are represented by an arrow with a white tip that links the subclasses to their superclasses. In the CEO the property that links an apparatus (and its subclasses) to an apparatus entry (and its subclasses) is called "has entry (is entry of)". On a semantic level, this property expresses that the apparatus is composed by a set of apparatus entries. In UML the same meaning can be expressed using a kind of association called "aggregation". Aggregations

in UML diagrams figure as arrows with a white diamond shape at one of its extremities. The scheme presented above can be expressed as a UML class diagram as follows.

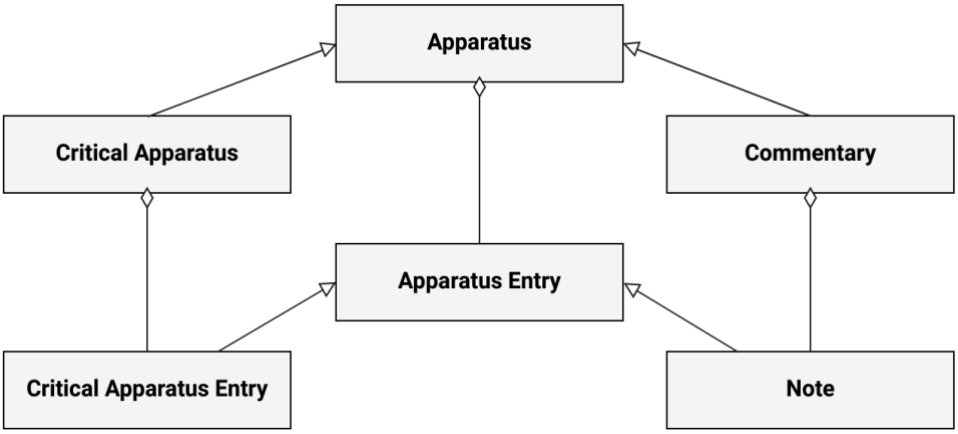

Figure 5 UML class diagram of the Apparatus and Apparatus Entry classes and their subclasses.

After this simple translation, developers and philologists can produce further diagrams that represent how the tool should be developed on a more concrete and detailed level. For this phase various aspects need to be considered. First, the technologies used to develop the tool, i.e. the developer platforms, the programming languages, etc. Other aspects depend on the kind of tool one wants to develop and the functionalities the tool should execute. For example, in order to develop a visualisation tool, the developers need to consider where and how the data that needs to be visualised is stored, in which format is the data preserved, etc.

Conclusions and next steps

This article presented a modelling strategy for the development of IT tools for digital philology. The strategy was developed considering two main answers to the lack of usable and adequate tools and to the current unsustainable practices of scholarly editing found in scientific literature. On one hand, the scientific community needs to establish "editorial models" [19], on the other hand the tools have to be developed taking into account the good practices and methodologies of software design and software engineering [6]. For these reasons, the modelling strategy aims at producing a network of editorial models formalised as ontologies, that can express the philologists scientific requirements and as conceptual models actually guide the development of an IT tool. The research I conducted over the course of my PhD is only preliminary compared to this goal. Concretely, the results reached so far are: the definition of the modelling strategy and the creation of a first conceptual model that represents critical editions, the Critical Edition Ontology.

The next steps of the present research should ascertain the effectiveness of the modelling strategy and how well an ontology can be useful for both philologists and software developers. More specifically, the next steps are:

- a review of the CEO, by removing superfluous elements, adding new concepts and specifying more technical aspects like the cardinality of the relationships;
- testing the effectiveness of the CEO against some real critical digital editions;
- establishing the morphisms necessary for the conversion from OWL to UML on the CEO;
- the development of a visualisation tool based on the CEO.

A part from the individual steps that are still necessary to complete the research, the key to the success of the modeling strategy is the collaboration between philologists and software developers. Hopefully, this article has proved that this collaboration is not only possible, but it could help solve the lack of flexible, usable and durable tools for digital philology.

Once the modelling strategy is solid, among the future developments of this research, there is first and foremost the creation of other conceptual models, to represent further types of edition, and, consequently, the implementation of tools for digital philology capable of managing multiple types of edition. Finally, a further development consists in applying the CEO, especially the part dedicated to the relationship between the critical text and the witnesses, in the field of digital semantic editions, which exploit semantic web technologies to offer their own contents in the form of linked open data, encouraging potential reuse of the data.

Acknowledgements

I would like to thank my tutors, professors Maria Sofia Lannutti and Roberto Rosselli Del Turco, and the other members of the commission of my doctoral thesis discussion, professor Lino Leonardi and Emiliano Degl'Innocenti, for the precious feedback. A special thanks to Simone Zenzaro, for helping me out with the most IT-related bits of my thesis.

References

[1] Bekiari, Chryssoula, George Bruseker, Erin Canning, Martin Doerr, Philippe Michon, Christian-Emil Ore, Stephen Stead, and Athanasios Velios. 2022. 'Definition of the CIDOC Conceptual Reference Model v. 7.1.2'.

[2] Chiesa, Paolo. 2002. *Elementi di critica testuale*. Pàtron.

[3] Ciotti, Fabio. 2014. 'Digital Literary and Cultural Studies: State of the Art and Perspectives'. *Between* 4 (8). https://doi.org/10.13125/2039-6597/1392.

[4] Ciotti, Fabio, and Francesca Tomasi. 2016. 'Formal Ontologies, Linked Data, and TEI Semantics'. *Journal of the Text Encoding Initiative*, no. Issue 9 (September). https://doi.org/10.4000/jtei.1480.

[5] Cools, Hans, and Roberta Padlina. 2021. 'Formal Semantics for Scholarly Editions'. In , edited by Elena Spadini, Francesca Tomasi, and Georg Vogeler, 15:97–124. Norderstedt: BoD. http://www.uni-koeln.de/.

[6] 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*, 269–74. Leiden: Sidestone Press.

[7] Dillen, Wout. 2020. 'Lexicon of Scholarly Editing'. Lexicon of Scholarly Editing. 2020. https://lexiconse.uantwerpen.be/.

[8] Evans, Eric. 2015. 'DDD Reference'. https://www.domainlanguage.com/ddd/reference/.

[9] Evans, Eric. 2004. *Domain-Driven Design : Tackling Complexity in the Heart of Software*. Addison-Wesley.

[10] Gabay, Simone, Jean-Baptiste Camps, Ariane Pinche, and Nicola Carboni. 2021. 'SegmOnto, A Controlled Vocabulary to Describe the Layout of Pages'. Paris/Genève. https://github.com/SegmOnto.

[11] Giovannetti, Francesca. 2019. 'The Critical Apparatus Ontology (CAO)'. OWL 2. https://fgiovannetti.github.io/cao/.

[12] 'ISO/IEC/IEEE International Standard - Systems and Software Engineering–Vocabulary'. 2017. *ISO/IEC/IEEE 24765:2017(E)*, 1–541. https://doi.org/10.1109/IEEESTD.2017.8016712.

[13] Mancinelli, Tiziana, and Elena Pierazzo. 2020. *Che cos'è un'edizione scientifica digitale*. Carocci.

[14] Martignano, Chiara. 2023. 'Un modello concettuale per le edizioni critiche digitali'. Tesi di Dottorato, Siena: Università degli Studi di Siena. http://dx.doi.org/10.25434/chiara-martignano_phd2023.

[15] Martignano, Chiara. 2021. 'Un modello concettuale per favorire lo sviluppo e il riutilizzo di app per edizioni digitali'. *Umanistica Digitale*, no. 10 (September): 71–88. https://doi.org/10.6092/issn.2532-8816/12620.

[16] Motik, Boris, Peter Patel-Schneider, and Bijan Parsia. n.d. 'OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)'. Accessed 21 May 2023. https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.

[17] 'Object Management Group Model Driven Architecture (MDA) Guide Rev. 2.0'. 2014. https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.

[18] Pan, Jeff Z., Steffen Staab, Uwe Aßmann, Jürgen Ebert, and Yuting Zhao, eds. 2013. *Ontology-Driven Software Development*. Berlin Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-642-31226-7.

[19] Pierazzo, Elena. 2019. 'What Future for Digital Scholarly Editions? From Haute Couture to Prêt-à-Porter'. *International Journal of Digital Humanities*, May. https://doi.org/10.1007/s42803-019-00019-3.

[20] Robinson, Peter. 2005. 'Current Issues in Making Digital Editions of Medieval Texts—or, Do Electronic Scholarly Editions Have a Future?' *Digital Medievalist* 1 (0). https://doi.org/10.16995/dm.8.

[21] Roelli, Philipp, and Caroline Macé. 2015. 'Parvum Lexicon Stemmatologicum'. 13 November 2015. https://wiki.helsinki.fi/display/stemmatology/Parvum+lexicon+stemmatologicum.

[22] Sahle, Patrick. 2016. 'What Is a Scholarly Digital Edition?' In *Digital Scholarly Editing*, edited by Matthew James Driscoll and Elena Pierazzo, 1st ed., 4:19–40. Theories and Practices. Open Book Publishers. https://www.jstor.org/stable/j.ctt1fzhh6v.6.

[23] Shillingsburg, Peter. 2013. 'Development Principles for Virtual Archives and Editions'. *Center for Textual Studies and Digital Humanities Publications*, January. https://ecommons.luc.edu/ctsdh_pubs/4.

[24] Shillingsburg, Peter. 1996. *Scholarly Editing in the Computer Age: Theory and Practice*. University of Michigan Press.