Modelling Knowledge for the PAVES-e Project: a Formal Ontology of Cesare Pavese's Work

Giuseppe Arena[1], Salvatore Cristofaro[2,*], Giovanni Gafà[3,*] and Daria Spampinato[2]

[1]*Independent Researcher*
[2]*CNR - ISTC, Catania, Italy*
[3]*Università di Catania, Italy - Université Jean Moulin Lyon 3 de Lyon*

Abstract

This paper describes the OntoPavese ontology developed as part of the PAVES-e project. OntoPavese is a large-sized ontology that takes care of various aspects of Cesare Pavese's works; it implements different representation levels of Pavese's production items, along with a number of other related features dealing with such entities as temporal events, persons and places. An ad hoc created WEB tool–*O*ntoPavese*PathExplorer*–is also described that allows to visually explore individual relationships defined within the ontology.

Keywords

Ontology, Cesare Pavese, Digital Scholarly Edition, Semantic Edition, FRBR, LRM, CIDOC-CRM, RiC, Visual Relationship Exploration, Knowledge extraction

1. Introduction

This paper presents OntoPavese,[1] a formal ontology designed to represent Cesare Pavese's work both in breadth and depth, structuring the available knowledge into a rigorous semantic hierarchy. OntoPavese is part of the PAVES-e project, which aims to represent Pavese's work in accordance with FAIR principles and to provide an integrated, web-based study environment–*Hyperedition*–as illustrated in [1, 2].[2] Several of Pavese's works, encoded in XML/TEI,[3] already constitute a digital scientific semantic archive edition–*PaveseInTesto*–accessible via a dedicated interface in TEI Publisher.[4] OntoPavese is a rich ontology that includes the entire bibliography of Pavese, spanning various literary genres, and will be further enhanced with relevant secondary literature.

The organization of Pavese's bibliographic, archival, and literary heritage has been handled according to the best practices and procedures of the DH community, particularly those related to digital scholarly editions [3],[5] the most recent digital semantic editions [4],[6] as well as bibliographic ontology models [6, 7, 8].[7]

In collaboration with domain experts, a list of competency questions (CQ) was identified to be answered, which guided the organization of the data. These CQ include: "What editions did work Y have?"; "What works did Pavese write of type W?"; "Which poems are part of collection Z?"; "What letters did Pavese write in period V?".

Proceedings of the Joint Ontology Workshops (JOWO) - Episode XI: The Sicilian Summer under the Etna, co-located with the 15th International Conference on Formal Ontology in Information Systems (FOIS 2025), September 8-9, 2025, Catania, Italy
*Corresponding author.
arenagiuseppe137@gmail.com (G. Arena); salvatore.cristofaro@cnr.it (S. Cristofaro); giovanni.gafa@phd.unict.it (G. Gafà); daria.spampinato@cnr.it (D. Spampinato)

[1]https://digitalpavese.cnr.it/en/ontopavese-2/. Last accessed 2025/08/18
[2]The *WEB portal* is accessible at https://digitalpavese.cnr.it/. Last accessed 2025/08/18
[3]Text Encoding Initiative, the *de facto* international standard for digital scholarly editions of texts https://www.tei-c.org/. Last accessed 2025/08/18
[4]https://teipublisher.com/. Last accessed 2025/08/18
[5]A comprehensive overview of digital scholarly editions can be found in the catalogs of Franzini https://dig-ed-cat.acdh.oeaw.ac.at/ and Sahle https://v3.digitale-edition.de/. Last accessed 2025/08/18
[6]One example above all others can be found in the edition of *Quaderni di Paolo Bufalini* in [5].
[7]A good example of a bibliographic ontology is BiGrafo, relating to the works of Franco Fortini in https://github.com/DFCLAM/bigrafo. Last accessed 2025/08/18

Even to obtain answers to the aforementioned CQ, we have chosen to accompany the *PaveseInTesto* edition with a formal (RDF/OWL) bibliographic ontology. Furthermore, ontologies are effective in addressing several issues and aspects that arise in the specialized semantic representation of data, as is the case with OntoPavese:

- The provided information is often incomplete. For instance, the creation date of a work might be missing details (e.g., the day).
- Web publications are, by their own nature, open-ended works that allow for corrections, integrations, additions [3].
- Many of the available data are structured into complex semantic hierarchies. For instance, Pavese's drafts may contain either private writings or literary works; the latter can be further divided into essays and creative works, and within creative works a distinction can be made between texts in verse and in prose.
- Using an ontology enables the integration of data to enrich query results; moreover, the adoption of well-known standards for the semantic description of the entities within the edition-archive allows to share information with the scientific community and turning PAVES-e into a semantic digital edition.

The paper also outlines the process of constructing the knowledge graph using the XML files containing TEI annotations from the digital editions, along with other project materials. Additionally, the *O*ntoPavese*PathExplorer* tool is described, enabling the visual exploration of individual relationships within the ontology.[8]

The paper is divided into three main sections. Section 2 provide insights on the nature of OntoPavese, its underlying data and models, and other related features; in Section 3, the tool *O*ntoPavese*PathExplorer* is presented, along with a description of its main functionalities, coupled with some formal details about abstract structures underpinning them. Finally, Section 4 concerns with population of OntoPavese.[9]

2. OntoPavese's model

2.1. The data

The data available within the project concern a very significant portion of Pavese's textual production; more precisely: all known editions, in volumes and periodicals, containing any text by the author, in their first edition and in subsequent editions and reprints (when deemed appropriate by the domain experts); all works written by Pavese, including the unpublished ones, regardless of whether they are private writings or intended for the public; all manuscripts of the author archived at the *"Guido Gozzano — Cesare Pavese" Study Center* that contain texts of the author's main creative works in prose and poetry, selected by the domain experts.

All of these entities need to be represented within the ontology, which requires, first and foremost, the identification of the main information items–*entity attributes*–that characterize them. In fact, from a careful analysis of the available materials and the project targets, the following needs emerged:

- In the case of editions, it is necessary to represent (as a minimum) the attributes related to the bibliographic metadata.
- Concerning works, besides the title, the need is to also represent the *type* (e.g., "poem" or "story"), the place and date of writing, and, in the case of letters, the recipient.
- For manuscripts, it is necessary to represent the attributes pertaining to the archival metadata.

[8] Both the OntoPavese ontology and the tool *O*ntoPavese*PathExplorer* were introduced in [2]. This paper presents a number of advancements and also provides more detailed and comprehensive descriptions compared to [2].

[9] The work is the result of a constant collaboration among the authors during the phases of conception, planning, drafting, and revision. In particular, Section 1 is credited to Daria Spampinato, Section 2 to Giovanni Gafà, Section 3 to Salvatore Cristofaro and Section 4 to Giuseppe Arena.

The first step in the development of our ontology is the identification of the most appropriate standards to represent this information, which can be understood through three *key data-related perspectives*: the *bibliographic perspective*, that deals with the editions of Pavese's texts; the *philological perspective*, which, in our case, mainly concerns the creative and revising process of the works; and the *archival perspective*, that pertains to the preservation of the documents that transmit those works and texts. For instance, in the case of a poetry collection such as *Lavorare stanca* [9], we have to describe the various editions and the characteristics of the work and of the textual units (the poems) that compose it, as well as those of the documents containing the different manuscript drafts of these poems, complete with the relevant archival information.

2.2. Modelling the bibliographic and filological information

2.2.1. The LRM Model

LRM (*Library Reference Model*) [10] is the standard de facto for describing the semantics of bibliographic information, established by IFLA. LRM is a high-level conceptual model, currently available as a formal ontology–LRMoo [11].[10]

OntoPavese uses various entities from LRMoo and the formal CIDOC-CRM ontology[11] to describe not only the bibliographic level of information of PAVES-e, but also the philological one. In particular, the ontology adopts the core structure of LRM, inherited from the FRBR model [12], exploiting the four *abstraction levels Work*, *Expression*, *Manifestation* and *Item* (*WEMI*), implemented in LRMoo by means of the classes *F1_Work*, *F2_Expression*, *F3_Manifestation*, and *F4_Item*, respectively.

2.2.2. A Modelling Example: *Lavorare stanca*

The levels *Work* and *Expression* allow to model Pavese's work from a philological perspective with considerable expressive power. Let us return to the case of the collection *Lavorare stanca*, mentioned earlier, which raised several editorial issues, even in terms of the drafting of the individual poems and their inclusion and ordering within the collection [13]. Two significantly different versions of the collection exist: the one published in 1936 by the Solaria publishing house [14], and the one edited in 1943 [9] by Einaudi. These two editions essentially differ in what poems they include (the 1936 edition had to contend with censorship under Fascism), as well as in the internal ordering and textual contents of these poems (although the changes at this level are often marginal).

We can represent *Lavorare stanca* as a single *Work* (i.e., an instance of *F1_Work*), realized (property *R3_is_realised_in*) in two different *Expressions* (i.e., two instances of *F2_Expression*), to which there correspond (property *R4i_is_embodied_in*) the first editions of 1936 and 1943 (instances of *F3_Manifestation*), as well as all subsequent editions that refer to either of these *Expressions*. The collection, as a *Work*, is related to the individual *Works* representing the poems it contains via the property *R67_has_part*, just as the *Expressions* of these poems are related to those of *Lavorare stanca*, to which they belong, through property *R5i_is_component_of*. It is worth mentioning that the choice of using these two properties has required some debate; indeed, LRMoo provides, for *F1_Work*, the property *R74_uses_expression_of*, and, for *F2_Expression*, the corresponding property *R75_incorporates*, that could be employed to describe the situation where a *Work* includes texts that realize another, different, *Work*. However, the possibility of employing *R74_uses_expression_of* and *R75_incorporates* proved problematic for the following reasons, and we therefore decided not to adopt them–though we do not rule out revisiting this choice in the future. The creative act of conceiving a collection (of poems) would not just consist (to our opinion,) in the mere sum of the creative acts by which the individual poems were composed. From this perspective, using the above properties to define the relationship between a work and its components, could then be the more appropriate choice. On the other hand, the use case examples accompanying the LRMoo

[10] http://iflastandards.info/ns/lrm/lrmoo/. Last accessed 2025/08/18

[11] The classes and properties of LRMoo integrate with those of the formal CIDOC-CRM ontology: http://www.cidoc-crm.org/cidoc-crm/. Last accessed 2025/08/18

definitions of *R74_uses_expression_of* and *R75_incorporates* actually seem to point in the opposite direction, where a poetry collection can be viewed as the ordered set of poems it contains. In any case, since our choice suits our purposes well enough, there is no need to dwell further on such issues.

2.2.3. Modelling a Single Work and Its Drafts

If the collection takes on different forms over time, then, as we have seen, the same can also be said of the poems it contains. LRM understandably leaves a certain degree of freedom about the extent of change that must occur in a work's text for it to be considered a different *Expression* (see [11, p. 24]). Once again, the choice rests with the domain experts, who determine on a case-by-case basis when a form assumed by the text can be considered a new *Expression*.

Text modifications do not occur only within the poems published in the 1936 and 1943 editions of *Lavorare stanca*; there are indeed several drafts of each text, both handwritten and typewritten, that differ–sometimes significantly–from each other and from the printed version. In the special case of manuscript texts[12] we have decided to create as many *Expressions* as there are successive drafts of each poem. To further distinguish these drafts from those corresponding to the publication, we internally defined the class *ExpressionDraft* as a means of collecting them separately from other *Expressions*. We also decided not to explicitly model the relationship between successive drafts. LRMoo does in fact provide the property *R76_is_derivative_of*, the description of which suggests that it could be used, among other things, to represent a revision of a text. Now, it would certainly be legitimate to establish that Pavese's corrective work on a given draft–typically quite intense–constitutes an *Expression* derivative of the source text it revised. However, that would lead us to represent each draft through multiple *Expressions*, thus implying a descriptive level that delves into the content of the individual text, which is beyond the scope of our initial objectives; conversely, asserting that a given draft can be considered the source of some subsequent draft would require a case-by-case philological analysis of the variants, which did not fall within our purview. Therefore, we limite ourselves to indicating, via a dedicated *data property*, the sequential number of the drafts that (at least) orders them chronologically, while leaving open the possibility of using property *R76_is_derivative_of* in the future, should the domain experts decide to add this level of description.

Figure 1 presents the instantiation of the ontology with reference to *Lavorare stanca* and one of the poems it comprises, *Ulisse*.[13]

2.2.4. Modelling the Manuscripts

Manuscripts were given particular attention, as their pages are reproduced in facsimile format on the WEB portal of the project, within the diplomatic editions of the works, and there was thus a desire to represent them in the ontology. Drafts of texts have a status that is difficult to capture within the *WEMI* model[10], which–while allowing the description of the developmental stages of a work from its conception to the individual printed copy–is aimed at representing bibliographic information and thus published texts, not those preserved in archives. In a manuscript, the *Manifestation* is exemplified by just one *Item*. Thus, as noted by Elena Pierazzo[3], all the features of the *Manifestation* are also features of the *Item*, making the distinction irrelevant in this case. Hence, as we were not interested in describing the *Manifestation* level for manuscripts, we introduced a *shortcut* property, *hasInstantiation*, to link each instance of *ExpressionDraft* to its corresponding instance in *PhysicalDraft*–another internally defined subclass of *F4_Item* (see also Section 2.3 for further details). Instances of *PhysicalDraft* are composed of instances of *PhysicalPage* (property *P46_is_composed_of* from the CIDOC-CRM ontology), a subclass of *E24_Physical_Man-Made_Thing*, also defined in the CIDOC-CRM ontology.

[12]Typescripts do not fall within the scope of information to be represented in the ontology.

[13]For the sake of clarity, the diagram does not display all *Manifestations* of the works, nor all extant drafts of the poem. Elements shown in black correspond to the classes and properties defined in LRMoo, whereas those in blue indicate the internally defined subclasses and subproperties thereof.

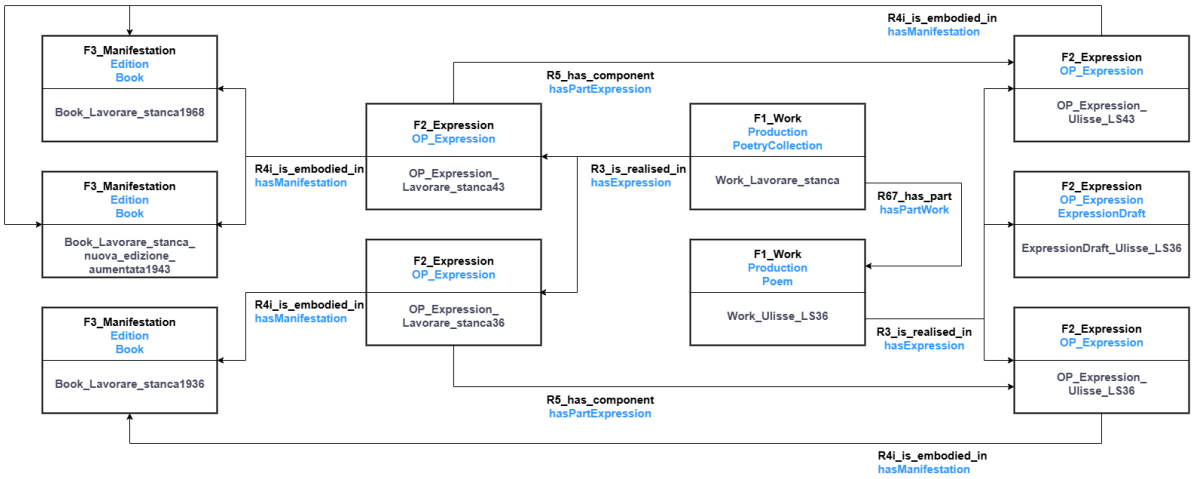

Figure 1: A segment of OntoPavese's knowledge graph, relating to the poetry collection *Lavorare stanca* and one of its constituent poems, *Ulisse*.

2.2.5. Modelling the Position and Sections of Texts

In regard to the bibliographic and philological information discussed so far, the need is also to represent the data about the position occupied by a text within a collection and its eventual belonging to a specific section of the work. In this case, it was necessary to adopt a compromise solution, for at least two reasons:

- On the one hand, property *R71_has_part*, which links an instance of *Manifestation* to its parts, at least according to its official description and related examples, seems to understand "parts" not as portions of a given volume, but only as individual publications within a set–such as a specific volume in a trilogy–and it has been used in OntoPavese in this way.
- On the other hand, information about the position of a poem in an anthology, as well as about the section containing it, would belong to the *Expression* level, as it results from the author's creative work. However, recording this information within the poem's *Expression* would be, if not a conceptual error, at least a representational stretch. This is because the poem can (and usually does) originate before, and independently of the decision to include it in a specific collection, and because it may appear in different collections and in different positions.

We therefore created the class *PartialEdition* that represents the texts comprising a specific publication and that allows, via the use of specific, dedicated *data properties*, the inclusion of information about the page numbers in which they appear and any sections to which they belong. The *Expression* of a poem included in a collection is thus published (property *is embodied in*) within the *Manifestation* of that collection, and specifically it is edited (property *hasPartialEdition*) in a part (class *PartialEdition*) of that volume, through which all the relevant data can be retrieved. Instances of *PartialEdition* are related to the instance of *F3_Manifestation* they belong to via another internally defined property (*isPartOf*).

2.2.6. Modelling the Genre of the Work

One of the projects requirements is the ability to perform searches by genre (or type) within Pavese's works. Information about the type of a work is not explicitly handled by LRMoo. The formal CIDOC-CRM ontology provides two possible mechanisms to categorize the described objects: the addition of subclass hierarchies to an existing, suitable class, or the use of the class *E55_Type* and the related property *P2_has_type*. Since the concepts we intended to represent are relatively stable–as indicated in the guideline of the CIDOC-CRM ontology–we preferred to define a system of subclasses within the *Work* class. This allows us to distinguish, first of all, between private texts (letters and diary notes) and

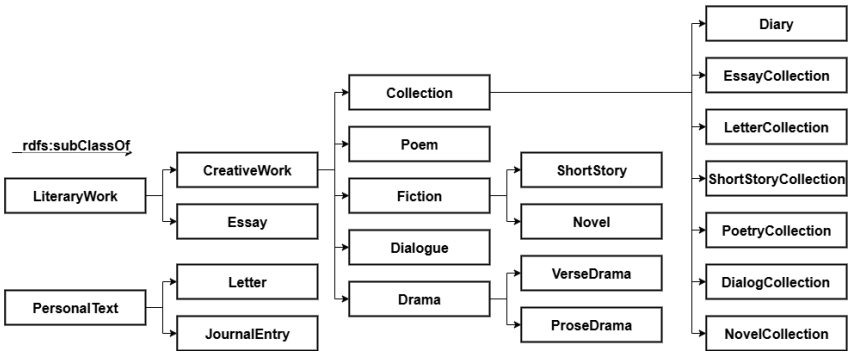

Figure 2: The *Work* subclass hierarchy defined within OntoPavese.

texts intended for publication, and then to identify work types within them such as poetry, dialogue, short story, etc. Figure 2 provides an overview of the currently defined hierarchy.[14]

2.3. Modelling Archival Information

Once the philological and bibliographic levels were defined, the next step was to choose a standard for representing archival materials. We adopted the authorative standard established by the ICA (*International Council on Archives*) with RiC-CM (*Records in Context Conceptual Model*),[15] whose latest version, 1.0, was recently released and includes significant updates. OntoPavese incorporates various entities from the formal RiC-o ontology, which was published alongside the conceptual model.[16]

One of the core classes of RiC-o is *RiC-E02_Record_Resource* (corresponding to the *Record Resource* entity in RiC-CM), which denotes, in the most general way, an archival resource, or, more precisely, its informational content, independently of any of its physical manifestation. Within this class, three subclasses are defined: *RiC-E04_Record*, which represents an informational object–*Record*–whose identity derives from the object itself, and which we use to represent individual folders; *RiC-E03_Record_Set*, which represents collections–*Record Set*s–of informational objects whose identity depends on their members, and which we employ for describing the fonds, and the series and subseries they contain; and *RiC-E05_Record_Part*, which represents parts of a *Record*–*Record Part*s–whose identity depends on the latter, and which we adopt to describe the witnesses[17] of a work found within a folder. We also rely on the class *RiC-E06_Instantiation*, which designates the inscription of the informational content of a *Record Resource* on a physical carrier.

As previously noted, our goal was to represent perspectives on certain objects. This point becomes clearer here. The witnesses of a work contained in a folder represent one or more drafts of that work. However, we already reference those drafts in the ontology as distinct *Expressions* in the LRM model. Presenting both archival and philological perspectives could therefore lead to a significant multiplication of entities in the ontology: the same set of manuscript pages would have to be described both as a *Record Part*, as well as an *Expression* (more precisely, an instance of class *ExpressionDraft*, see Section 2.2.3).

After consulting the authors of RiC we concluded that the nature of an *Expression* in the LRM model and that of a *Record Resource* are similar: both refer to informational content independently of the physical form that may contain it. Likewise, *Items* and *Instantiation*s describe physical objects with comparable characteristics. It therefore seemed legitimate to consider an individual manuscript draft of a work both as an *Expression* and as a *Record Part*, and we hence decided to define *ExpressionDraft* as a

[14] Note that, while individual diary pages belong to class *PersonalText*, the diary as a whole is classified as a creative work, in accordance with the domain experts' recommendation: in fact, Pavese had planned its publication and even gave it a title [15, p. LXXII].

[15] https://www.ica.org/resource/records-in-contexts-ontology/. Last accessed 2025/08/18

[16] The version of the RiC-o ontology used is 1.02, available at: https://www.ica.org/standards/RiC/RiC-O_1-0-2.html. Last accessed 2025/08/18. We are considering integrating the new version 1.1, which was recently released.

[17] In philology, a witness is any manuscript or printed source that transmits a text, i.e., a physical copy through which the text has been preserved and is accessible today.

subclass of the two corresponding classes. The same applies to *PhysicalDraft*, that we characterized as a subclass of the classes *F4_Item* and *RiC-E06_Instantiation*. The property *hasInstantiation* (see Section 2.2.4) was then defined as a subproperty of the property *hasOrHadInstantiation* in RiC-o.[18]

2.4. Modelling Dates

In this section we illustrate our proposal to represent dates related to specific types of individuals described in the ontology: the date when a *Work* was conceived;[19] the date when a specific draft of a work was created; and the date when a *Manifestation* was published. This task was particularly delicate, as the available material presents a number of different scenarios for dates (or date intervals): some are fully documented (day, month, and year), but more often they are partially documented (e.g., day and month are missing), or entirely absent. Incomplete or missing dates could be be simply considered as uncertain; however, in some cases, the editor of the work has reconstructed the missing information with a reasonable degree of confidence.

Both CIDOC-CRM and RiC include classes and properties for managing dates. Additionally, there exists the OWL-Time ontology,[20] developed by the *SDWWG* (*Spatial Data on the Web Working Group*) for the *W3C* and *OGC* (*Open Geospatial Consortium*). Each of these models has its own strengths and weaknesses. However, we chose not to explicitly incorporate any of them into our ontology. Instead, we followed a different, more direct approach (see below), as it fulfills our aims, particularly in addressing the competency questions we formulated (cf. Section 1), while avoiding unnecessary complexity. In fact, we realized that the most effective way to represent dates and support the variety of queries we envisioned, was to create the three dedicated classes *Day*, *Month*, and *Year* typing the day, month and year items of a date, respectively. An individual of the class *Date* is linked to its components (day, month, year) through specific properties, which also allow us to specify whether each component is certain, reconstructed by the editor, or simply unknown.

Works, drafts, and editions are always linked to their relevant creation-date interval by means of the two properties *hasCreationDateFrom* (start of the interval) and *hasCreationDateTo* (end of the interval). The case in which the creation date is not an interval is handled by simply treating the start and end of the interval as equal objects.

3. Visually exploring individual relationships within OntoPavese

As explained in previous sections, OntoPavese is a large-sized ontology that takes care of various aspects of Pavese's works. Various string processing and data retrieval tools have been devised and used to (partially) populate the ontology, particularly in terms of *class assertions* and *object* and *data property assertions*,[21] where the XML files containing the TEI annotations of Pavese's works , as well as other textual sources, are suitably processed in order to identify and extract the relevant data items to be inserted into the ontology, which are subsequently translated (by means of XSLT transformations and the like) into RDF format.[22] Currently, OntoPavese has been populated with 3,434 class assertions, 18,662

[18] To our knowledge, there is currently no attempt to harmonize the RiC model with CIDOC-CRM and LRM, and our project thus positions itself as an initial step towards this direction.

[19] We deemed it reasonable to follow the recommendation of LRM, and refer to the creation date of its first corresponding *Expression*

[20] https://www.w3.org/TR/2022/CRD-owl-time-20221115/. Last accessed 2025/08/18

[21] For completeness, we recall here that a class assertion formally consists of a double (i.e., an ordered pair) (x, C), where x is an *individual* (i.e., *class instance*) and C is a *class*, whereas an object property assertion (resp., a data property assertion) is a triple $\langle s, P, o\rangle$, where s–the *subject*–is an individual, P–the *predicate*–is an object property (resp., a data property), and o–the *object*–is an individual (resp., a literal). Object and data property assertions are collectively called property assertions. (For more on these notions, and other related ones used in this section, we refer to [16] and [17].) If s, P and o are the subject, predicate and object of a property assertion, respectively, then we say that s *is related to* o *by* P. Moreover, if x and y are individuals, then y *is reachable from* x, if there is a sequence of individuals starting at x and ending at y, and such that any individual in the sequence is related to the next by an object property. The *reachable individuals* are thus precisely the objects of (object) property assertions.

[22] See Section 4 for a more detailed description of the population processes of OntoPavese.

object property assertions, and 7,678 data property assertions, based on a (newly created) *signature* comprising 2,984 individuals, 30 classes, 30 object properties, and 15 data properties. As the project moves forward, these numbers are definitely expected to grow. To assist and guide users in exploring this plethora of *ontological entities*, a dedicated web tool–*O*ntoPavese *PathExplorer*–is being developed. *O*ntoPavese *PathExplorer* allows for an interactively queryable, visual exploration of *(relationship) paths*, where a path is (in the present context) a sequence of individuals (the *vertices* of the path), each related to the next by means of an object property. Vertices within a path can be explored in any order, freely moving from one vertex to the next or vice versa. Moreover, for each individual x traversed (as a path vertex) during exploration, *O*ntoPavese *PathExplorer* provides information–precomputed and easily accessible via the user interface–on the object properties that relate x to other individuals and, in particular, on the *depths* and *heights* of x: namely, the lengths of the longest *simple paths* (i.e., paths without repeated vertices) that start and end at x. Thus, after selecting a *target individual* of interest, one can potentially explore all paths that start and end at that individual, while continuously accessing *positional information*–depths and heights–of the vertices traversed along the way, thereby enabling a more comprehensive exploration. This provides an easy and intuitive mechanism for readily accessing useful structural information about the semantic organization of the ontology at the level of individual relationships.[23]

From a conceptual point of view, *O*ntoPavese *PathExplorer* organizes the individuals of the ontology hierarchically in a tree-like structure–the *path-tree*–, arranged vertically along *branches* of increasing levels—corresponding to their shared depths—and grouped horizontally based on the object properties that relate individuals at the preceding level to them. More formally, the path-tree consists of the abstract *edge-labelled tree* $\mathcal{T}$ (over the family of individual sets), defined recursively by the following conditions:

(1) The *root* of $\mathcal{T}$ is the set of all individuals y such that, for no property assertion $\langle s, P, o\rangle$ in the ontology, it is the case that $y = o$ (i.e., y is not a reachable individual; see footnote 21).[24]
(2) For any *node* N of $\mathcal{T}$, consisting of the set $\{x_1, x_2, \ldots, x_n\}$ of individuals, and any object property P, the (possibly empty) set of all individuals y such that, for some $1 \leqslant i \leqslant n$, $\langle x_i, P, y\rangle$ is a property assertion in the ontology, is a *child-node* of N, *connected* to N by an (single) *edge* labelled P.
(3) No node of $\mathcal{T}$ has child-nodes other than those given by clause (2).

Then, once a *target set* S of individuals is selected, *O*ntoPavese *PathExplorer* constructs (using information stored within the nodes of $\mathcal{T}$) and displays two separate collections, $\boldsymbol{L}'$ and $\boldsymbol{L}''$, consisting, respectively, of all object properties P' and all object properties P'', such that any individual s' in S is related to some individual o' by P' (in which case we say that o' *is entailed by* S *via application of* P'), and for any individual s'' in S, there exists some individual o'' related to s'' by P'' (in which case we say that o'' *is entailed by* S *via inverse application of* P'');[25] and in fact, after choosing an object property from one of these two collections, *O*ntoPavese *PathExplorer* computes the corresponding set $\Delta(S)$ of individuals that are entailed by S. This process can be iterated starting from any previously entailed set, leading to the sequence of individual sets $S, \Delta(S), \Delta(\Delta(S)), \ldots$ that the user hence successively explores and that *O*ntoPavese *PathExplorer* in fact visualizes along with the object properties used during entailments. Note that these individual sets $S, \Delta(S), \Delta(\Delta(S)), \ldots$ can also be stored, as they are computed, within a dedicated area, from which they can be subsequently selected by the user and combined by means of the usual *set-theoretic operations* of *union*, *intersection* and *complementation*. The

[23] Note that information on properties that relate ontology individuals can be easily retrieved, e.g., by means of simple SPARQL queries. However, SPARQL does not allow the direct retrieval of positional information.

[24] Observe that, if there were an individual x such that x is reachable from itself, then the root of $\mathcal{T}$ would have actually to be a *maximal* (however chosen) set of individuals, any two of which are not reachable one from the other. (This indeed guarantees that all individuals defined in the ontology are included within the nodes of $\mathcal{T}$). However, in the case at hand of the ontology OntoPavese, such an individual x reachable from itself does not de facto exist, and thus the provided definition of the root of $\mathcal{T}$ clearly makes it a maximal set as above.

[25] Note that the properties P' in $\boldsymbol{L}'$ are grouped based on the depths of the individuals to which individuals s in S are related by P', whereas properties P'' in $\boldsymbol{L}''$ are grouped based on the heights of the individuals that are related to s by P''.

combined sets can then be used in the *entailment process* as well. These mechanisms allow for even complex semantic queryings on individual relationships within the ontology.[26]

The path-tree $\mathcal{T}$ is built-up by using the *OWL Functional-style Syntax representation* (see https://www.w3.org/TR/owl2-syntax/) of the ontology, from which the object property assertions are extracted first, and then organized into a *relation matrix* which is subsequently used to construct $\mathcal{T}$.

Observe that the graphical interface of *O*ntoPavese*PathExplorer* presents a list of individuals defined within the ontology, searchable by IRI or label, from which the user can select the desired ones to include in the target sets (see Figure. 3). Notice also that, besides the functionalities described above,

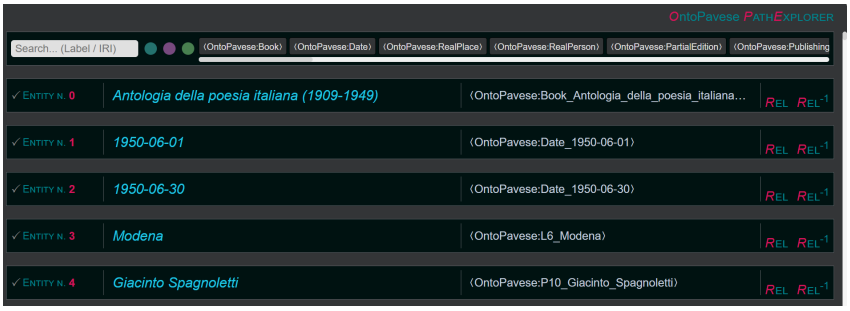

Figure 3: Graphical interface of *O*ntoPavese*PathExplorer*

for any individual x selected by the user, *O*ntoPavese*PathExplorer* can even visualize, upon request, the collection $\mathfrak{A}(x)$ of all *complete-paths* starting at x, linearly arranged in consecutive rows, where a complete-path starting at x is a maximal length sequence of property assertions such that the object of any property assertion in the sequence equals the subject of the next property assertion, and the subject of first property assertion is x.[27] In order to facilitate the view of the hierarchical structure of vertices in these complete paths (i.e., how the subjects and objects of the property assertions are related each other), an option is provided to visualize the entire collection $\mathfrak{A}(x)$ in tree form (see Figure. 4).

In concluding the section, it is worth mentioning that, originally, *O*ntoPavese*PathExplorer* was intended to provide a faithful tree-like visual representation of the path-tree in its whole, i.e., with all of its nodes and edges collectively painted on screen (cif. [2]); however, as the project moved forward, a number of issues arose concerning the huge amount of data items to be fully represented, as well as, and in particular, issues related to user interactions during set entailments, that turned out somewhat awkward to manage visually due to the tree-like visual representation of the path-tree. The version of *O*ntoPavese*PathExplorer* described here is lighter, and more intuitive compared to the original one.[28]

As a final remark, we also mention that, although *O*ntoPavese*PathExplorer* has been developed for the OntoPavese ontology, the software implementing it is actually designed to accept in input the *OWL Functional-style Syntax representation* of any OWL ontology, from which *O*ntoPavese*PathExplorer* is automatically constructed. In fact, the software has also been tested on a specialized ontology, currently under development within the COVerLeSS project (see [18, 19]), that deals with the representation of *Italian Verism terms*, and for which the use of *O*ntoPavese*PathExplorer* has indeed proven fruitful, leading to the discovery of useful relationships between the terms.

4. On the (semi-automatic) population of OntoPavese

The population of OntoPavese has been carried out semi-automatically through a structured process of extracting, transforming, and loading bibliographic data from heterogeneous sources.

[26] Actually, "individual *vs* literal" relationships are also accounted for by *O*ntoPavese*PathExplorer*, where literals just are treated as individuals, and data properties as object properties.

[27] By footnote 24, such maximal length sequences of property assertions are clearly well defined.

[28] However, the original version has not been definitively abandoned, as it may still be reconsidered and integrated with the current one.

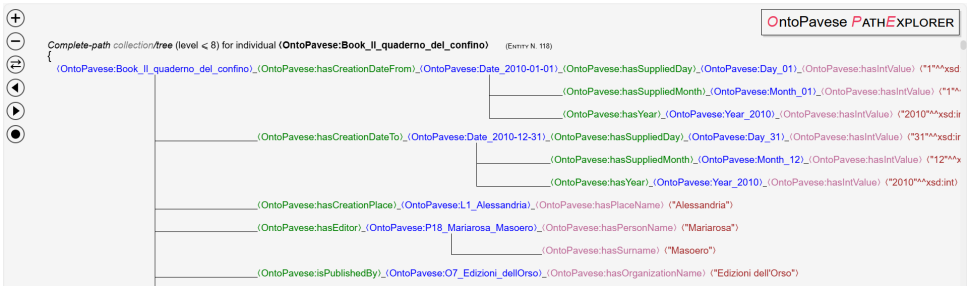

Figure 4: Tree-like visual representation of complete-paths

In the first extraction phase, these data were manually extracted and normalized from a PDF bibliographic catalog such as [20] and from *Pavese's Opera Poetica* [21]. Human intervention was required to interpret layout structures, distinguishing author names, titles, and edition details. For XML/TEI files (letters, diaries, poems, essays), automatic XSLT transformations were used to extract relevant metadata (such as filenames, authors, dates and, for letters, sender, recipient and place of origin).[29]

In the second transformation phase, the collected data from primary bibliographic entries, their content and data extracted from XML files were (re)organized into three groups of Excel spreadsheets, structured around seven main ontological classes (*Work*, *Expression*, *Manifestation*, *Person*, *Place*, *Organization* and *Date*). Each individual is identified by a unique ID, which, in combination with the namespace of the ontology, allows the automatic generation of an IRI that conforms to Semantic Web standards. The population process were automated through a Python script that employs the pandas[30] and RDFLib[31] libraries to map (Excel) table columns to OntoPavese's properties, distinguishing between data properties, for literal values, and object properties for relationships between individuals.

In the third loading phase, after initializing the RDF graph and loading the reference ontology, the code performs a syntactic cleanup of IRIs (using the *clean_uri* function), and sets dictionaries for mapping between table columns and RDF properties. Processing occurs systematically for each of the seven ontology spreadsheets: each row generates a resource identified by an IRI, properties and readable labels. The system also supports the presence of multiple relationships, allows the coexistence of distinct types for the same resource, and incorporates control features to handle missing values. The resulting RDF data are exported as an RDF/XML file, which is subsequently imported into GraphDB.

The workflow was guided by specific methodological principles. To begin with, each resource (including journals and collected publications) has to be represented in its entirety within the three LRMoo levels *Work*, *Expression*, and *Manifestation* (unless it is included in the *PartialEdition* class, in which case it is sufficient to know the range of the relevant pages). Moreover, a *Work* needs to be associated with a single *Expression*, except in those situations where textual variants (as in the case of the two editions of *Lavorare stanca* of 1936 and 1943), or interpretive doubts suggest the creation of multiple *Expressions* to preserve potentially relevant information. Furthermore, in order to manage information redundancy, a suitable criterion is adopted, by which such key information such as title, author, and language are replicated in the three LRMoo levels. Finally, in the case of works published under different titles but referable to the same conceptual entity (for instance, the Pavese's essay *The Spoon River Anthology* reprinted with a new title as a part of the posthumous collection of essays *La Letteratura Americana e altri saggi*), in order to safeguard the intellectual identity of the resource, it is necessary to keep separate *Expressions* that refer to the same *Work*.

Knowledge extraction is a crucial step in the construction of knowledge graphs, which typically relies on techniques such as Named Entity Recognition, Natural Language Processing, and Machine

[29]This is because Pavese's production is very extensive and, at this stage of the PAVES-e project, only part of the texts has been encoded in XML/TEI, while OntoPavese already contains bibliographic information relating to his entire work. As a result, some of the data needed to populate the ontology was already available in XML/TEI, while other data were collected from various sources.

[30]https://pandas.pydata.org/docs/. Last accessed 2025/08/18

[31]https://rdflib.readthedocs.io/en/stable/. Last accessed 2025/08/18

Learning, of increasing relevance in the field of Digital Humanities [22, 23]. Declarative mapping languages such as R2RML and RML offer powerful means to integrate heterogeneous data sources into RDF [24]; yet their applicability is limited in contexts characterized by fragmented and incrementally expanding datasets, and they were therefore not adopted in this project. This decision was based not only on the high heterogeneity of the source formats, ranging from XML/TEI files to bibliographic data in PDF, but also on the need for precise domain-specific disambiguation, which could not be reliably automated. Instead, the chosen combination of Excel, RDFLib, and XSLT supports incremental data entry and editorial revision while avoiding the rigidity of an early fixed database structure. Although it may introduce some risk of inconsistency, semantic coherence is ensured through reasoning and RDF visualization tools during the post-processing stages.

5. Conclusions and future work

This paper presents the OntoPavese ontology, developed as part of the PAVES-e project, which focuses on the literary production of Cesare Pavese. It also introduces the *O*ntoPavese *PathExplorer* tool, designed to explore individual relationships within the ontology. Although both the ontology and the visualization tool are in an advanced stage of development, they are not yet fully complete. In the near future, we plan to finalize the remaining components and test the functionality of both the ontology and the visualization tool in practical scenarios. We also intend to engage external users in the testing process, providing them with tailored guides and usage tips, especially for the *O*ntoPavese*PathExplorer* tool.

Acknowledgments

The PRIN 2022 PAVES-e project, developed in collaboration with the University of Catania (PI: Antonio Sichera), the University of Turin (AI: Laura Nay) and the CNR (AI: Daria Spampinato), is funded by the European Union – Next Generation EU, M4C2 (CUP B53D23022270006). The authors also thank the PRIN PNRR 2022 COVerLeSS project, funded by the European Union – Next Generation EU, M4C2 (CUP B53D23029310001). Special thanks are also due to Florence Clavaud, member of the *Expert Group on Archival Description* of the *International Council on Archives*, for her generous assistance, and to Laura Mazzagufo, engaged in the design of the *PaveseInTesto* interface with TEI Publisher, for her suggestions.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] C. D'Agata, A. M. Del Grosso, L. Nay, G. Palazzolo, A. Sichera, D. Spampinato, PAVES-e: Per una Hyperedizione dell'opera di Cesare Pavese, in: A. D. Silvestro, D. Spampinato (Eds.), AIUCD 2024 Me.Te. Digitali. Mediterraneo in rete tra testi e contesti, Proceedings del XIII Convegno Annuale AIUCD2024, 2024, pp. 191–196. doi:10.6092/unibo/amsacta/7927.

[2] L. Mazzagufo, S. Cristofaro, C. D'Agata, A. M. Del Grosso, P. Sichera, A. Sichera, D. Spampinato, Moving towards a semantic archival edition: the paves-e project, DH2025 Book of Abstracts (in press).

[3] E. Pierazzo, Digital Scholarly Editing: Theories, Models and Methods, Routledge, London New York, 2016. doi:10.4324/9781315577227.

[4] F. Tomasi, Organizzare la conoscenza: digital humanities e web semantico: un percorso tra archivi, biblioteche e musei, volume 39 of *Biblioteconomia e scienza dell'informazione*, Editrice Bibliografica, Milano, 2022. URL: https://doi.org/10.53134/9788893573573.

[5] M. Daquino, F. Giovannetti, F. Tomasi, Linked Data per le edizioni scientifiche digitali. Il workflow di pubblicazione dell'edizione semantica del quaderno di appunti di Paolo Bufalini, Umanistica Digitale 3 (2019). doi:10.6092/issn.2532-8816/9091.
[6] M. T. Biagetti (Ed.), Le ontologie bibliografiche: modelli concettuali e vocabolari condivisi per l'universo bibliografico, Bulzoni, Roma, 2022.
[7] M. T. Biagetti, A Comparative analysis and evaluation of bibliographic ontologies, in: F. Ribeiro, M. E. Cerveira (Eds.), Challenges and Opportunities for Knowledge Organization. The Digital Age. Proceedings of the Fifteenth International ISKO Conference 9-11 July 2018 Porto, Portugal, Baden Baden, Ergon, 2018, pp. 501–510.
[8] S. Peroni, Semantic Web Technologies and Legal Scholarly Publishing, number 15 in Law, Governance and Technology Series, Springer International Publishing, Cham, 2014. URL: https://doi.org/10.1007/978-3-319-04777-5.
[9] C. Pavese, Lavorare stanca - nuova edizione aumentata, Einaudi, Torino, 1943.
[10] P. Riva, P. Le Bœuf, M. Žumer, A conceptual model for bibliographic information, IFLA Library Reference Model, Netherlands (2017).
[11] T. Aalberg, P. Riva, M. Žumer, LRMoo: object-oriented definition and mapping from the IFLA Library Reference Model (2024). URL: https://repository.ifla.org/handle/20.500.14598/3677, publisher: International Federation of Library Associations and Institutions (IFLA).
[12] K. Coyle, FRBR, before and after: a look at our bibliographic models, ALA Editions, an imprint of the American Library Association, Chicago, 2016.
[13] L. P. Barbarino, Il primo "Lavorare stanca" di Pavese (1936). Edizione critica, Sinestesie, Avellino, 2020.
[14] C. Pavese, Lavorare stanca, Solaria, Firenze, 1936.
[15] C. Pavese, C. Segre, Il mestiere di vivere: 1935-1950, number 716 in Einaudi tascabili ET Scrittori, Einaudi, Torino, 2000.
[16] D. Allemang, J. Hendler, F. Gandon, Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS, and OWL, volume 33, 3 ed., Association for Computing Machinery, New York, NY, USA, 2020.
[17] S. Rudolph, Foundations of Description Logics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 76–136. doi:10.1007/978-3-642-23032-5_2.
[18] D. Bruno, G. Canzoneri, A. D. Silvestro, D. Spampinato, A. Zammataro, Un corpus online della letteratura secondaria (1872- 1890) del Verismo italiano, in: A. D. Silvestro, D. Spampinato (Eds.), AIUCD 2024 Me.Te. Digitali. Mediterraneo in rete tra testi e contesti, Proceedings del XIII Convegno Annuale AIUCD2024, 2024, pp. 232–239. doi:10.6092/unibo/amsacta/7927.
[19] D. Bruno, G. Canzoneri, S. Cristofaro, A. Di Silvestro, L. Mazzagufo, D. Spampinato, Reading the works of verism through the magazines and critics of xixth century. text editions, lexical maps and thematic dictionaries in a user-oriented portal, Umanistica Digitale 9 (2025) 1–30. doi:10.6092/issn.2532-8816/21302.
[20] L. Mesiano, Cesare Pavese di carta e di parole: bibliografia ragionata e analitica, I libri di "Levia gravia", Edizioni dell'Orso, Alessandria, 2007.
[21] C. Pavese, L'Opera poetica. Testi editi, inediti e traduzioni, Mondadori, Milano, 2021.
[22] M. Ehrmann, A. Hamdi, E. L. Pontes, M. Romanello, A. Doucet, Named Entity Recognition and Classification in Historical Documents: A Survey, ACM Comput. Surv. 56 (2023) 27:1–27:47. doi:10.1145/3604931.
[23] L. Giagnolini, A. Schimmenti, P. Bonora, F. Tomasi, Expliciting Contexts: Semantic Knowledge Extraction from Traditional Archival Descriptions, Umanistica Digitale (2025) 115–144. doi:10.6092/issn.2532-8816/21229.
[24] D. Van Assche, T. Delva, G. Haesendonck, P. Heyvaert, B. De Meester, A. Dimou, Declarative RDF graph generation from heterogeneous (semi-)structured data: A systematic literature review, Journal of Web Semantics 75 (2023) 100753. doi:10.1016/j.websem.2022.100753.