OntoPoetry: Postdata Ontology for poetry domain.

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Elena González-Blanco[b], Omar Khalil[a] , Salvador Ros[a *], Mirella de Sisto[a] ,Laura Hernández[a], Javier de la Rosa[a], Alvaro Pérez[a] ,Aitor Diez[a], , Oscar Corcho[c]

[a] *POSTADATA Project. SSCC. Escuela Técnica Superior de Informática, UNED, Madrid, Spain*
[b] *POSTADATA Project .School of Human Sciences and Technology, IE University, Madrid, Spain*
[c] *Ontology Engineering Group. ETSI Informática, Universidad Politécnica de Madrid, Madrid, Spain*

Abstract. The idiosyncrasy of literary studies has been an obstacle to its technological improvement for years, especially to represent their knowledge in a machine-readable format. The richness, variety, and different study`s perspectives that scholars find in their studies make this task a highly complex challenge. This complexity is even more noticed in the poetry genre, where each poetic tradition has independently developed its analytical terminology and methodology. In this work, we have addressed the construction of a poetry ontology to express the scholar´s knowledge spread out in isolated databases or works. Ontopoetry ontology has been developed following Neon methodology, and it has been structured in three modules: a) core, b) poetic analysis and c) transmission, covering the essential aspects in a poetry literary study. Ontopoetry core module has been aligned with FRBRoo ontology guaranteeing its interoperability. This paper is focused on the description of the core module, its classes and relationships and the design decisions taken during the process. We also describe the proposed controlled vocabularies for this module and their relationship with the remaining modules.

Keywords: European Poetry, Standardization, Network of Ontologies, Interoperability, Linked Open Data

1. Introduction

The need for standardization has increased significantly in different research fields to understand and exchange information in their knowledge domains. Many scientific disciplines have established formal protocols and languages for quickly adapting to their problems. In the case of humanities and cultural fields, they have followed an independent path in which creativity and tradition play an essential role where standardization is a problematic matter. Literature, and especially poetry, is a clear reflection of this idiosyncrasy.

From the philological point of view, there is no uniform academic approach to analyze, classify, or study the different poetic manifestations. The divergence of theories is even more prominent when comparing poetry schools from other languages and periods. One of

* Corresponding author. E-mail:sros@scc.uned.es.

the most significant conceptual and terminological problems is that, even when a set of poetic works is formalized under a repertoire, each repertoire belongs to its poetic tradition, and each tradition has independently developed its analytical terminology, in some cases for centuries [1]. This uncoordinated evolution produces a bunch of terminologies to explain similar metrical phenomena through the different poetic systems whose correspondences have been hardly studied. For example, the same quatrain of dodecasyllables can be encoded in different ways depending on the philological tradition (i.e. 12A12A12A12A or 4x(7pp+7p) or 4aaaa) or even named with a different meaning: "alexandrine" means a 14-syllable line in Spanish but an only 12-syllables line in French [2]. As a result, if a researcher were looking for quatrains of dodecasyllables in different traditions, it would be necessary to visit each database independently and then carry out different searches adapting the query to the conventions of the repertories. Furthermore, the workload of those researchers who are not experts in databases could be increased in cases in which repertoire's interfaces do not provide a detailed enough query editor or because there is no tool to export the data to be managed in another way. In addition, the poetry research is usually conducted in an individual and isolated manner, and there is a certain lack of communication with other areas of knowledge. Therefore, these repertoires are based on stand-alone collected databases [3]–[8]. Interoperability among all these collections would be useful to perform comparative studies and move a step forward beyond the modern philological state-of-the-art, to explain phenomena like the origins of vernacular poetry or the evolution from accentual to syllabic rhythmical patterns.

Although the current technical infrastructures are prepared to harvest such collections and provide access to them by search engines, it is necessary to use metadata and vocabularies at a philological level to climb up the semantic layer and link data among different traditions [9]–[11].

In this context, the use of technologies applied to poetry is ground-breaking. This way of representing distributed literary collections as machine-readable repositories will open the door to pose new research questions and perform comparative philological analysis between heterogeneous poetic corpora with different formats.

All these difficulties and problems of access to poetic resources and, in short, the impossibility of having the means to process this information completely and efficiently have been the origin and the incentive for the conception of an ontology of poetry [11], [12]. For this purpose, we have extracted from a set of repertoires of different poetic traditions and periods, [13], [14], the concepts and relationships necessary to achieve a general representation of the poetry domain. This study identified the core concepts of poetic resources as cultural heritage objects and concepts related to other complementary areas related to poetic analysis and bibliographic information. The result of the whole process has led us to develop a complete ontology for poetry.

This work presents the methodology carried out to build the core ontology devoted to poetry, and it is part of the results of the Postdata (Poetry Standardization and Linked Open Data) ERC project, which aims to provide a means for poetry researchers to publish and consume semantically enriched data in the context of poetry.

The document is structured as follows. In section §2, we identify some previous results related to ontologies in the literature, especially in the domain of poetry. Section §3 describes the methodology used to develop OntoPoetry Ontology modules. Section §4. describes the modules that form the OntoPoetry ontological model. Section§5 presents a detailed description of the OntoPoetry core ontological model. Finally, the results and future works are presented in Section§6.

2. Related works

One of the consequences of the evolution of the Semantic Web and the progressive transformation of Humanities into "Digital Humanities" has been the implementation of new metadata mark-up languages and the great number of ontologies developed and published to describe traditional concepts with computer-readable languages (e.g., Text Encoding Initiative TEI-XML[1], Dublin Core[2] or CIDOC-CRM[3]). In this sense, libraries, museums, and archives [4]have led this process, and many projects are using these approaches in their developments (e.g., TexGrid[5], OpenEdition[6]

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

[2] https://www.dublincore.org/specifications/dublin-core/

[3] http://www.cidoc-crm.org/

[4] http://lodlam.net/

[5] https://textgrid.de/

[6] https://www.openedition.org/

or Scholar Digital Editions (SDE)[7]). However, as far as our knowledge extends, the application of these paradigms to computational literary studies focused on poetry is still limited [15], [16]

From the computational literary studies point of view, a poem can be considered a work representing an original idea. The ideas of the work can be expressed following an intellectual process (e.g., writing, translating, or editing a poem) that yields to the different editions or expressions of a work. In addition, each poem has a physical representation that conveys the expressed ideas (e.g., a manuscript or a book).

Considering these different levels of information related to a poem, we can find ontologies that cover some of the concepts associated with literary poetry studies.

One of the most known ontologies in the literature is the CIDOC Conceptual Reference Model (CIDOC-CRM)[8]. This ontology formally describes the concepts and relationships used to document cultural heritage. This model focuses on the representation of museums' heritage works and contains concepts for representing entities such as people and places associated with the works. Therefore, we could model a poetic work by considering it a cultural heritage item. Likely, this approach would have been enough for modelling a work, but we would have lost the information about other aspects of a work as its physical representation for which we can associate a bibliographic entry. To model these physical aspects, we find the Functional Requirements of Bibliographic Records (FRBR)[9] and its object-oriented version FRBRoo[10]. These ontologies offer a perspective on the structure and relationships of bibliographic and authority records [17]. The most significant entities in these ontologies are Work, Expression, Manifestation, and Item, representing the different ways of conceiving a literary work as a text or physical resource. FRBRoo model harmonizes information from museums, archives, libraries, and cultural heritage entities. Therefore, these ontologies can cover the descriptive aspects of the works and their forms of expression and manifestation.

However, modelling poetic resources as cultural heritage and bibliographic objects is not enough to represent their structure and analysis. In this sense, one of the first approaches to model the poetry domain was to define a set of metadata tags to annotate forms and structures of poetic works [5] as part of the Text Encoding Initiative (TEI) [18]. Although this set of tags captured part of a poem's metric domain knowledge, they definitely do not cover other metrical concepts related to literary analysis or prosody and do not follow semantic web technologies to provide the annotations as linked data resources.

Furthermore, making poetry specific information available as linked data will ease the expansion of poetry domain knowledge with other aspects related to linguistic or rhetorical concepts for which we find the following ontologies: a) the Lexicon Model for Ontologies (Lemon) [11] designed for modelling machine-readable dictionaries and lexicons covering aspects of lexical decomposition, sentence structure, syntax, variation, morphology, b) The Gold[12] ontology a complete ontology for descriptive linguistics, that describes the most basic categories and relationships used in the scientific description of human language, or c) The Rhetorical Annotation Ontology Project (RAOP)[13], a specific domain ontology, built for the annotation of speech figures and the rhetorical aspects of written and oral texts.

Beyond the analysis of well-known ontologies in Digital Humanities, we can find the first steps made in the Postdata project to define a poetry ontology. It started looking for nonontological resources from highly reliable sources. Twenty-five repertories described in [19], most of them available on the web, were selected by a set of experts. They represent different poetry traditions, languages, prosodic systems, and cultures [20], resulting from research projects. Hence, they contain information gathered or generated by experts, which improves the reliability and robustness of data, categories, and structure. Once the sources were selected, a complete European Poetry Domain Model, EPDM, was made by transforming the nonontological resources. During this process was necessary to address problems related to the consistency of the concepts and disagreements. These problems were faced by experts in the poetry domain [9]–[11]. The result was an EPDM that includes details about textual transmission and aspects related to prosody, literary and rhetorical analysis, the structures identified in the poems, significant elements for publications, and its relationships with music. [13], [14]. Starting from the EPDM, we built the first ontology version [21]. After a first review, the ontology lacked alignment with ontological resources. Therefore, we

[7] http://www.sd-editions.com/

[8] http://www.cidoc-crm.org/cidoc-crm

[9] http://purl.org/vocab/frbr/core#

[10] http://www.cidoc-crm.org/frbroo/home-0

[11] https://www.lemon-model.net/index.php

[12] http://purl.org/linguistics/gold/

[13] http://bakulf.github.io/raop/

built this new version fully aligned to the foundational ontologies, which supports better interoperability while providing more straightforward and useful models for the end-users.

3. Description of OntoPoetry Ontology.

Ontopoetry Ontology facilitates a set of concepts for describing poetic works (poems, poetic drama or plays written in verse and songs). It is the product of a homogenization effort that considers different literary traditions, periods, poetic genres, and authorship. Additionally, this will enable the comparison of the characteristics and data in this poetry and thus carry out invaluable research in Comparative Literature and Comparative Metrical Studies quantitatively.

3.1. Modularization Approach.

Given the excellent coverage of the Postdata Ontology V1.0 [21] and after a critical analysis of it made by experts on ontology engineering and poetry, it was concluded that it was cumbersome to manage a single artefact as the result of the development tasks. Therefore, a redefinition of the subdomains would improve the semantic description of the ontology, its interoperability, and ease of use. To solve this weakness, we opted for re-modularisation, intending to ease the maintenance and publication of the ontology [22]. Modularisation helps during the re-engineering and alignment tasks as engineers can work in parallel and not be overwhelmed with significant structures or many ontological elements. It also allows users to understand the ontology better and have more flexibility in deciding which modules to reuse [23] based on the identification of relevant use cases for the potential users of OntoPoetry Ontology after a previous analysis of the relevant applications or objectives of the ontology and by enforcing encapsulation and independence as modularisation criteria. Encapsulation involves the identification of elements that are related to a given sub-domain so that a module could be easily exchanged by another, while independence seeks the identification of self-contained modules and their potential reuse [22].

This analysis yielded the definition of two potential cases of use used to define the main OntoPoetry Ontology subdomains (i.e., modules):

- **Bibliographic information search and indexing**: tackling the requirements of the poetry scholar community researchers involves the intensive usage of bibliographic information, mainly from the librarian domain. This information is used to differentiate the different conceptual and physical properties of the works yielding to the definition of two modules:
  - OntoPoetry Core module represents the abstract or conceptual side of the bibliographic information. Abstract entities of works are required to differentiate between the original idea of a work and its different expressions. Repositories work with abstract notions of works that can be used to better differentiate between different editions or textual versions and to know more about the intention or original ideas of the authors. Relevant classes tackled by this module are Poeticwork, Ensemble and Redaction. Therefore, this module includes all the essential information that characterizes works and their expressions, irrespective of their physical materializations.
  - OntoPoetry Transmission module represents the more tangible side of bibliographic information related to poetic works. It extends the descriptions of the Postdata Core module by linking the abstract textual editions or instances of Redaction with appropriate sources, between which we distinguish BibliographicSources (which correspond to the conceptual notion of a bibliographic product) and PrimarySources (which correspond to unique physical sources such manuscripts). Other relevant classes include Item and Facsimile, and they are physical artefacts corresponding to exemplars of bibliographic products and reproductions of other physical artefacts. Hence, it provides ontological elements to describe the materializations and manifestations of expressions, including ownership, identification, integrity status, dimensions, and material, among others.
- **Poetic information annotation and searching**:
  - OntoPoetry Poetic Analysis module: this module represents different phenomena associated with metrics and prosody, including the textual elements or parts of a poem and the different metrical patterns that analyze those elements. Relevant classes related to this sub-domain include Line, Stanza, Rhyme, Foot and Syllable. The ontology allows for metrical information annotation using scansion symbols for each line and describes each element's analysis at a metrical syllable, foot or morae level. In

addition, we also include means to annotate the presence of literary devices such as enjambment, synalepha, and other relevant figures of speech.

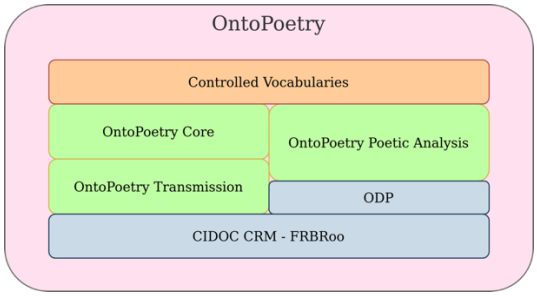

Fig. 1 . Encapsulation of OntoPoetry Ontology

Therefore, OntoPoetry Ontology comprises three ontology modules, each of them developed and serialized independently following OWL and RDF standards, Fig. 1.

Each module covers a specific sub-domain as much as possible, so we expect the ontology modules to exhibit a high module cohesion (degree of relatedness between the ontological elements defined in the same module). Another desired property of the ontology modules is related to the logical quality. In a modularisation scenario, we must ensure two properties for each module so that the complete set of ontologies is safe ("they do not produce unexpected results such as new inconsistencies or subsumptions between imported symbols" [24]).

- Local Correctness of the module: Ensure that any module does not entail any additional fact for the elements defined by the other modules than those entailed by the element's module.
- Local Completeness of the module: ensure that all the entailments of the global ontology about the interest elements are preserved in this module's entailments.

Core and Transmission modules are entirely founded on the FRBRoo model, while the Poetic Analysis module is partially based on the FRBRoo model and the selection of appropriate Ontology Design Patterns (ODPs). The partial alignment of Poetic Analysis to the FRBRoo model comes from the need to reuse the basic entities from CIDOC-CRM, such as events, time and agents.

All the ontology modules include properties that rely on the concepts of a set of controlled vocabularies or concept schemes that help users to better categorize or classify each of the elements described by the ontological modules at a deeper level than the one provided by the defined classes. This is a very useful step towards consolidating different poetic traditions as each tradition may consider different classifications (i.e., types of stanzas and lines, types of stress and scansion technique used, the objective of the poem, and the main theme of the poem.). While a complete analysis of all the possible concepts relevant for each tradition has not been thoroughly conducted, we have identified those aspects that vary across traditions [25]. Then, the main contribution in this sense is identifying discrepancies and providing initial concepts that serve as an example to guide the inclusion of further concepts as long as the project evolves. This means that most of the vocabularies are open to editions and subjected to modifications after deployment of the Postdata Knowledge Graph. In this work, for the sake of space, we will introduce the OntoPoetry Core module.

4. Methodology

Once we defined the three Ontopoetry modules, we followed Neon scenario six for building core and transmission modules and NeOn scenario seven for Poetic analysis module[26],[27] **(see** Fig. 2)

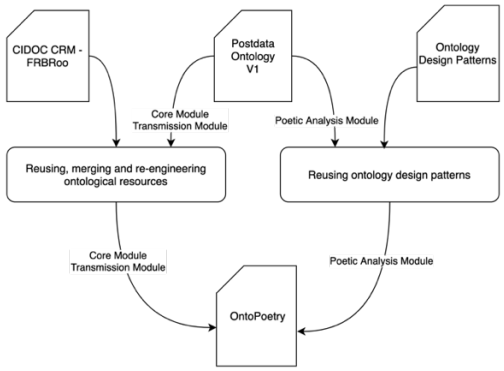

Fig. 2 NeOn scenarios six and seven applied to Ontopoetry Ontology.

4.1. Scenario six: Reusing, merging, and re-engineering ontological resources

Neon Scenario six [24] includes ontological resource reuse, reusing and merging ontological resources and ontological resource re-engineering. This scenario starts from the idea of reusing existing ontologies to take profit of its benefits (i.e., increase interoperability or reduce the development costs) [27], [29]. Ontology reuse is generally understood as adopting existing ontologies to solve different problems in an ontology development process [26]. The realization

of the reuse process can be seen as the consecution of various integration operations, such as the inclusion of a whole ontology or the introduction or removal of ontological elements or pieces of an ontology [30]. This scenario was adapted for building core and transmission Ontopoetry modules. (see Fig. 3).

These modules describe concepts related to the concepts described by bibliographic and cultural heritage ontologies.

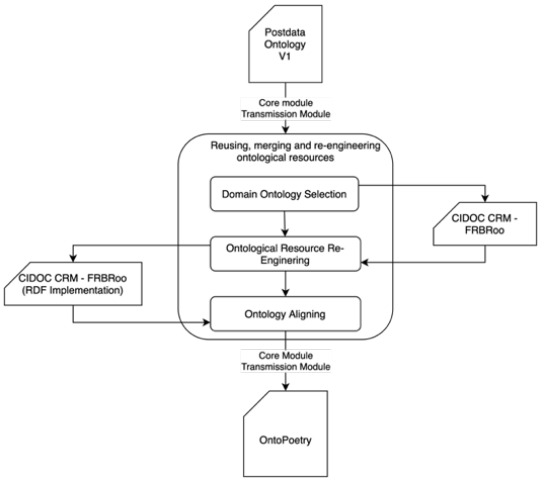

Fig. 3 Scenario six Adaptation.

Therefore, starting from the Postadata Ontology V1 [21], we selected the remarkable foundational ontologies that better fit the Postdata project's objective of providing an interoperable model by reusing well-known used ontologies in these fields.

While there was no poetic-related ontology that fitted the project purposes, several common standard ontologies in the librarian domain were helpful for our purposes. In our case, CIDOC CRM [31] acts as a foundational ontology that offers a solid philosophical foundation, so adopting its pragmatics and semantics is of great importance towards interoperability. FRBRoo is an extension of CIDOC CRM that facilitates a domain ontology for the librarian domain that is well thought to tackle different requirements or publication paths, so the syntax and semantics of FRBRoo [32], [33] are of great importance to identify those requirements that match our needs. We concluded to use CIDOC CRM and its extension FRBRoo. Then, the task of reusing ontologies was limited to reusing these ontologies.

Nevertheless, the alignment of the mentioned Postdata modules with FRBRoo models cannot be done straightforward for several reasons: (1) FRBRoo includes an RDF serialization using OWL semantics, but it could be considered as an abstract model that could be implemented according to any object-oriented representation; (2) CIDOC CRM provides elements that offer a higher level of description than the one we need and (3) Postdata Ontology V1 is the result of a translation of an abstract data model that does not comply with the modelling practices nor pragmatics or structure of CIDOC CRM model. For this reason, before reaching the ontology alignment task, it was necessary to re-engineer the ontologies.

Therefore, the process involved re-engineering or modification and merging or aligning the mentioned ontologies to facilitate an integrated ontology that conformed to the standard and included those elements that fall outside the standard. We adapted the activities of those processes according to our needs

4.2. Reusing ontology design patterns.

This task is performed by finding and applying appropriate Ontology Design Patterns. Ontology Design Patterns (ODPs) are modelling solutions that have been extensively used to solve specific ontology design problems, and for this reason, reusing them is being considered as a good practice for the development of ontologies[27], [34]. In this way, reusing ODPs facilitates an ontology design for domain experts and knowledge engineers. In addition, it also helps to reduce the efforts of ontology alignment or ontology integration, making the ontologies more reusable.

At this moment, several types of ODPs exist, but, in our case, we only consider Content Ontology Design Patterns or Content Patterns (CPs), [35], which propose modelling solutions for the domain classes and properties that define the ontology. We can reuse CPs from the ODP repository[14] or foundational or core ontology.

In this work, after searching and selecting appropriate CPs, we performed a strategy based on the integration of them by indirectly reusing the elements contained in the OWL files that represent the ODP template (see Fig. 4). This scenario was applied to build the Poetic analysis module.

[14] http://ontologydesignpatterns.org

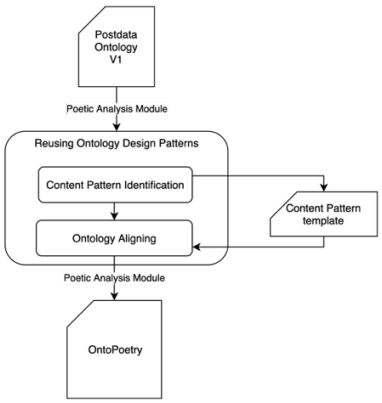

Fig. 4. Scenario seven Adaptation.

5. Ontopoetry Core Ontology Module

The core module resides at the heart of the OntoPoetry Ontology, which facilitates a set of classes and properties to describe general aspects of poetic work (i.e., a poem). The core module describes the original idea of poetic work and its related textual editions. Hence, it enables the representation of information related to the written language and various elements concerning the author, such as the literary period it can be ascribed to.

The core Ontology module is aligned and integrated with FRBRoo departing from the Postdata core Ontology V1 [21] by following the modularization criteria of Section §3, and the methodology explained in Section §4. We selected FRBRoo 2.4 [36] and CIDOC-CRM 6.0 [31] as the base ontologies for the alignment analysis. While CIDOC-CRM provides mechanisms to represent cultural objects according to a precise, logical foundation, FRBRoo extends the previous model by integrating requirements from the bibliographic domain. The alignment with FRBRoo is done by including classes and properties as subclasses or sub-properties whenever there is the need to specialize the ontology. We directly reuse the rest of the elements from FRBRoo whenever they cover the EPDM or Postdata Ontology V1 information needs by declaring equivalent classes or properties. Therefore, we will refer to FRBRoo classes or CIDOC CRM classes even if the former extends the latter model. While CIDOC CRM classes and properties identifiers include prefixes **E** and **P**, FRBRoo classes and properties include prefixes **F** and **R**.

[15] Entities belonging to CIDOC model will start with the prefix "E" and entities belonging to FRBRoo start with the prefix "F". All of these entities are highlighted in bold across the document.

As CIDOC-CRM and FRBRoo are conceptual models that do not completely follow the RDF data model, we had to modify the properties or classes of these models during the modelling process.

In the following subsections, we present a description of the main classes organized by their content or purpose. For each set of classes, we include how we modelled some properties of interest to the poetry domain, highlighting those that required the modification or specialization of the models since CIDOC-CRM is a foundational ontology and FRBRoo is a general domain ontology that does not provide elements specific to the poetry domain.

5.1. Poetic Works, Redaction, Ensemble and Excerpt Description

Postdata Core ontology module aims to model the knowledge related to a work. In this sense, it distinguishes between the abstract concept of a poem or set of poems and the textual realization of them. Therefore, Postdata ontology is modelled considering two primary levels of representation: The level for representing the ideas of a poetic work (i.e., work level) and the level to represent the textual signs of poetic works (i.e., expression level) and their relationships. Taking into consideration, we analysed FRBRoo ontology to get the best alignment[15].

In FRBRoo, we can find **`F1 Work`**. This class is an abstract class used to refer to the conceptual idea of a work that "appear in the course of the coherent evolution of an original idea into one or more expressions that are dominated by the original idea" [36]. Together with this definition, FRBRoo defines different useful types of works considered in the modelling process. These classes are used to model the work level.

Therefore, we have defined the class **`pdc:PoeticWork`** that represents an abstract concept of artistic creation. In the context of the Postdata project, only artistic works in verse are considered: poems, plays written in verse and songs. General data about the work such as title, subtitles, literary tradition or the purpose of the poetic work or redaction (e.g., to be sung, prayer, educational, spell) are included in this class. Being an abstract concept of artistic creation, it does not correspond to any material object that we can identify as the poetic work but to the original idea of the work. The class **`pdc:PoeticWork`** corresponds to an **`F14 Individual Work.`** In our case, each

poetic work corresponds to the idea or concept of a single poem, play or song, irrespective of its presentation or its contribution to a higher idea or concept. In this way, each **pdc:PoeticWork** is realized in only one self-contained expression in terms of FRBRoo, and do not have other works as parts, Fig. 5.

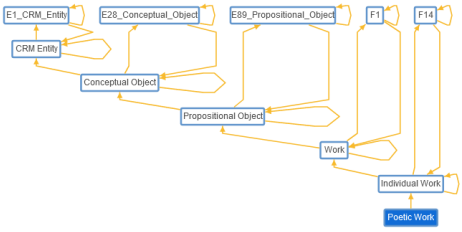

Fig. 5. Class hierarchy for **pdc:PoeticWork**

On the other hand, we can find in FRBRoo **F2 Expression** for modelling the expression level. This class represents the intellectual content of a work in some form or set of signs rather than concepts. An expression of a work refers to its signs, but it should not be considered a physical entity. For example, a work captures the idea of a poem, and the expression reflects the text or other tangible signs that express the work irrespective of its physical format or carrier.

We define the **pdc:Redaction** class to represent textual versions or editions of a **pdc:PoeticWork**. It represents the different textual contents or signs of a poetic work (i.e., particular poem, play or song) irrespective of layouts, fonts, material, and other elements that correspond to the physical or digital version of a poetic work (i.e., manifestation level in FRBRoo). The class **pdc:Redaction** is a subclass of **F22 Self Contained Expression**. Since the signs of a **pdc:Redaction** are textual, we considered it was helpful to reflect that **pdc:Redaction** is also a subclass of CIDOC-CRM **E33 Linguistic Object.** This allows us to indicate that the expressions we reflect in the redaction class correspond to natural language information objects, Fig. 6.

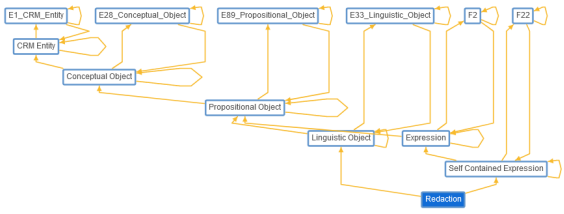

Fig. 6 Class hierarchy for **pdc:Redaction**

Therefore, **pdc:PoeticWork** always represents the ideas of a single poem or song and the class **pdc:Redaction** always carries out one **pdc:PoeticWork.** OntoPoetry can express compositions and derivations of ideas and which textual signs represent these ideas. This layered representation enables the retrieval of several editions or redactions for the same poetic work, which is particularly important to users conducting studies in the poetry domain. Sometimes a work can be considered as part of another work or derivation. To model these different situations, we have implemented two classes.

On the one hand, **pdc:Ensemble** is defined as an abstract work whose idea is the compilation of different textual editions or instances of **pdc:Redaction**, each corresponding to a **pdc:PoeticWork**(i.e. aggregation of expressions from other works). We defined **pdc:Ensemble** as an **F17 Aggregation Work**, Fig. 7. We mainly use **F22 Self Contained Expression** and **F23 Expression Fragment** to represent the expressions of interest.

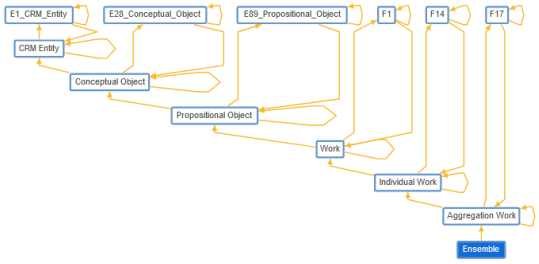

Fig. 7. Class hierarchy for **pdc:Ensemble.**

As an example, " La Antologia de la nueva Poesia" (i.e., New Poetry´s anthology) by José Batlló is an anthology of Spanish authors. Each of the poems can be considered a poetic work and the anthology a Self-contained Expression. Similarly, "Algunas obras de Herrera" (i.e., Some Herreras´s works) can be considered a Self-Contained Expression because it comprises a set of poems created and compiled by Fernando de Herrera. In both cases, the Self-Contained Expression realizing the Ensemble is comprised by the rest of the expressions compiled.

On the other hand, **pdc:ComplexWork** represents those cases in which several works contribute to the ideas of a given original work. This is the case when the original work's ideas are shared among a set of component works or when an original work has been derived (i.e., adaptation), giving rise to a different work even if it also contributes to the original work's ideas. We defined the class **pdc:ComplexWork** as a subclass of **F15 Complex Work**, Fig. 8.

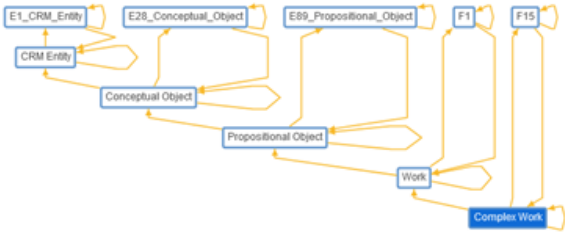

Fig. 8 Class hierarchy for **pdc:ComplexWork**

Hence, a **pdc:ComplexWork** should not be confused with a **pdc:Ensemble** that represents an aggregation work whose original idea is the compilation of expressions of already existing works. Aggregations may happen at different levels (i.e., abstract or physical). As the core module reflects the abstract notions of poetic works, here we will refer to aggregations at the work level. Aggregations at the work level or abstract level mean that the author's original idea is the compilation of works, thus acting as a compiler. This differs from the idea of aggregating several editions in a specific bibliographic source, which is a task performed by editorials and hence corresponds to aggregations at the source level that are covered by the OntoPoetry Transmission ontology module.

When a researcher is working with a work, sometimes needs to distinguish different parts of the work to be further described with some textual annotations. To represent those common parts of the work´s signs, we created a generic class **pdc:Excerpt** as subclasses of **F23 Expression Fragment** and **E33 LinguisticObject** to indicate that a **pdc:Redaction** can be broken down into pieces of textual signs smaller than the referred redaction, Fig. 9.

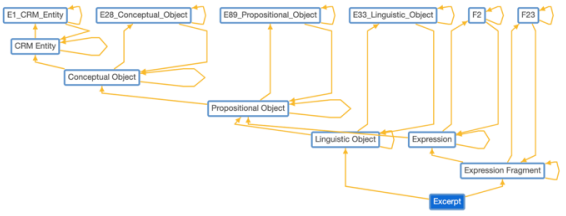

Fig. 9. Class hierarchy for **pdc:Excerpt**

Each fragment is characterized by a set of object properties subproperties of **pdc:hasFragment** as for example **pdc:hasCommentarty**, **pdc:hasExplicit**, **pdc:hasIncipit** and **pdc:hasSample** and their inverses **pdc:isCommentaryOf**, **pdc:Isexplicitof**, **pdc:isincipit** and **pdc:issample**. The content of each fragment is described by the datatype property **pdc:text**, subproperty of **P190 has symbolic content**, [37].

Once defined the main classes of the core module, we defined a set of properties to enable a detailed description of the relationships between the most relevant entities of this module: works and expressions.

5.1.1. Work-Work relationships

A work-work relationships express the different relations between the classes that are subclasses of **pdc:Work.** We can express three primary relationships.

In the first place, we have to model that a work has other works as members. This situation is given when we want to express different original poetic works that contribute to a poetic work because there is a relevant intellectual or artistic effort to modify another work. In this case, we reused the classes and properties using equivalence axioms to deal with the complex work case by declaring **pdc:isWorkOf** as comparable property of **R10 is member of** so we included **pdc:Work** and **pdc:ComplexWork** as domain and range for the property. In Fig. 10, we show how the work "El libro del Buen Amor"(The Book of the Good love) by Juan Ruíz could be modelled. The scholars mostly agree that in this work, Juan Ruíz used previously composed episodes to create the first version of the book. In this sense the lines 199 to 216 are a tale named "Enxiemplo de las ranas, en cómo demandavan rey a don Júpiter". This tale can be considered a poetic work part of "El Libro del buen amor" that will be considered a complex work.

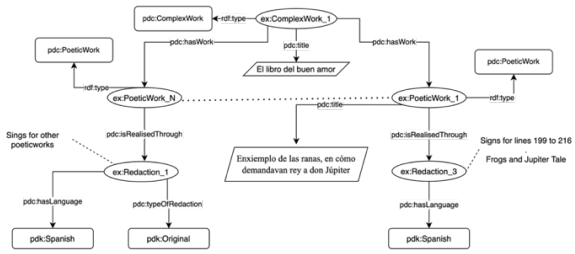

Fig. 10.Different Poetic Works and Complex Work.

Secondly, we can make explicit the case when several derivations or components (i.e., **pdc:poeticwork**) contribute to the same idea or concept. Those derivations can be expressed using a generic property **pdc:isDerivativeOf** to express derivation of works (i.e., a work is a synthesis or adaptation of another work) that we defined as equivalent to **R2 is derivative of**. For example "La vida acompañada" by Bartolomé de Carranza (RB,II/531,fols. 246v-248r) (i.e., VA poeticwork) is considered a contrafactum from Fray Luis de Leon´s

"Oda a la Vida retirada" (i.e., OVR poeticwork). That said, **VA pdc:poeticwork isDerivateOf OVR pdc:poeticwork**.

5.1.2. Work-Expression relationship

These relations model the use cases between a redaction and a poetic work. We can express that a **pdc:Redaction** realizes a **pdc:PoeticWork**, through two properties: **pdc:realizes** and its inverse **pdc:isRealisedThrough.** These properties have been established as a subproperty of **R9 is realized in Self-Contained Expression of Individual Work** and its inverse **R9i**. An example of poetic work realization is shown in Fig. 11. The simpler case of use is when a poetic work represents the ideas of a single poem or song, and we have only one textual expression of the poem. In this case, a **pdc:Redaction** realizes one **pdc:PoeticWork.** This example models the fact that the author Fernando Herrera wrote a poem named Soneto I. Its realization containing its signs is on **ex:Redaction_1,** written in Spanish.

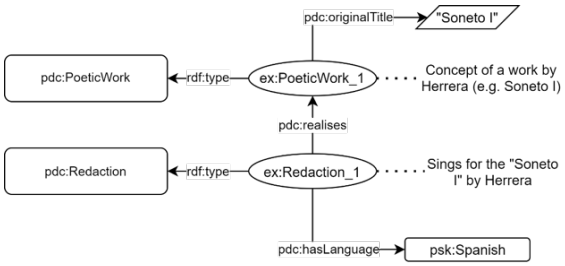

Fig. 11. Example of Poetic Work realization.

Another common situation in poetry is when we want to express that several **pdc:Redaction** realizing the creation of the same **pdc:PoeticWork.** This situation means more than one textual set of signs realizing the same idea or concept, Fig. 12. This will be the case whenever the type of intellectual process employed to create an original poetic work results in the production of a new expression. One example is when we have expressions in different languages that do not modify the original idea of the work. Other examples include a musical transcription or a revision or update on another text, for which we will consider different **pdc:Redaction** of the same **pd:PoeticWork**. In Fig. 12, is expressed the Sonet I from Herrera and two realizations corresponding to the Spanish and the English versions.

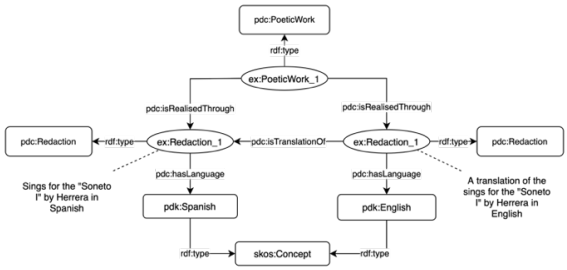

Fig. 12. Example of several Redactions for a single Poetic Work.

Finally, sometimes it is important to differentiate the most characteristic expression for a work. For this situation, we also reused **R40 has representative expression**, so we included **pdc:hasRepresentativeExpression** as equivalent to **R40** property.

5.1.3. Expression-Expression relationships

One of the most common relationships between expressions is when the research wants to include some information from a part of another redaction. To relate a **pdc:Excerpt** to a **pdc:Redaction,** we have defined the property **pdc:hasFragment,** which is aligned with **R15 has fragment** and its inverse **pdc:isFragmentOf** subproperty of **R15i**.

If we consider redactions as textual entities or textual editions of poetic works, we defined two properties to relate instances of **pdc:Redaction** that share textual characteristics. On the one hand, we defined **pdc:edits** as a subproperty of **P130 shows features of** to indicate textual editions of a given redaction, and **pdc:hasTranslation** as subproperty of **P73 has translation** to make translations explicit. Being a translation a special case of the edition, we made **pdc:hasTranslation** as subproperty of **pdc:isEditedIn** (inverse of **pdc:edits**).

Finally, a special case of expression-expression relationships is when we need to establish the relationships to explain how we relate an aggregation work, represented by **pdc:Ensemble** with the works it compiles (e.g., the expression of an anthology of poems, realizing an ensemble). According to FRBRoo recommendations [37], instead of using a property to relate the ensemble with the rest of poetic works, it is necessary to create a new expression for the ensemble that will be linked to the expressions of the poems the work compiles (i.e., **pdc:redaction**). We defined **pdc:isRealisedInSelfContainedExpression** as equivalent to **R3 is realized in Self-Contained Expression of Work** to relate an aggregation work to its expressions, and

pdc:incorporates as equivalent to **R14 incorporates** to reflect the fact that the expression of an ensemble incorporates the expressions of other works.

Fig. 13 shows the hypothesis that Fernando de Herrera's original idea for the work "Algunas obras de Herrera" (Some Herreras´s works) was to create an anthology of poems, an aggregation at work level rather than creating independent poetic works, to be delivered to King Felipe II. This hypothesis is reflected in the example by using an instance of **pdc:Ensemble** to represent the aggregation work. Following the FRBRoo alignment, we can relate the expression of this aggregation work with the redactions of all the poetic works that represent the signs and ideas of the poems compiled using **pdc:incorporates.**

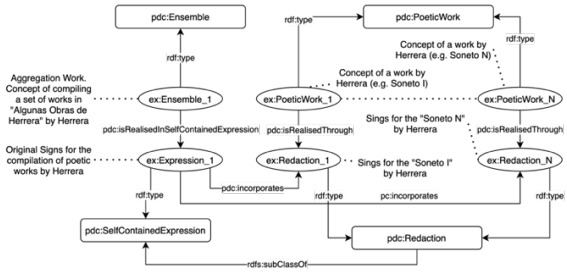

Fig. 13. Example of an Ensemble

5.2. Events, Agents and Roles

Since Postdata Ontology V2 has been aligned with FRBRoo follows an event-centric approach in which all the endeavours (instances of **pdc:PoeticWork, pdc:Ensemble** and **pdc:Redaction**) are the outcome of creation or modification events. These events are related to the participating agents, where this participation can be associated with a particular role (e.g., creator, editor transcriber). This section introduces the description of the fundamental classes used to deal with events, agents and roles and the ontological engineering decisions taken for aligning.

5.2.1. Events

The use of an event-centric approach in FRBRoo implies to model sets of phenomena and cultural manifestations occurring in time and space in which actors participate and produce endeavours. These are considered events and can be described with information about a place and a time.

In Ontopoetry Ontology, we identified two principal types of events: a) certainty b) death and birth.

a) Certainty is a property needed for giving reliability to the events when there is some sort of imprecision or certainty about a fact. For example, we may not be sure about the actual date of creation of a poetic work, or we could be very sure about the authorship of a given poetic work. Those cases are specific cases of a certainty assessment. To reflect certainty, it is possible to use a solution based on CRMInf extension model classes, but it would introduce the need to use named graphs in the RDF representation, which could increase the solution's complexity [38]. Instead of this solution, following CIDOC CRM recommendations is to define certainty as a subclass of **E16 Measurement**, so it would be possible to relate the thing being measured with the value of the measurement (the certainty value), [38]. This solution represents certainty even if the issue to represent belief and certainty or fuzziness is not closed is proposed, Fig. 14.

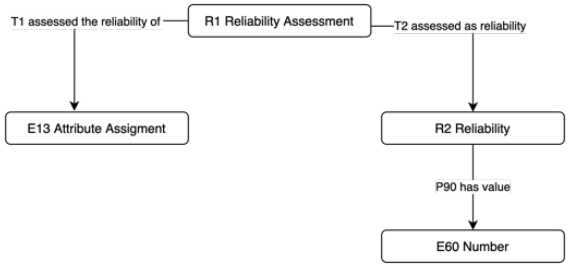

Fig. 14 Niccolucci's proposal to model certainty [Expressing reliability with CIDOC CRM]

Taking as starting point this model, we need to create a new class to represent the act of assessing certainty (subclass of **E16 Measurement**) and a class to represent the attribute assignment whose certainty is being assessed (subclass of **E13 Attribute Assignment**). We also need new properties to relate the entity and the activity being assessed with the attribute value (subproperties of **P141 assigned** and **P140 assigned attribute** respectively) and the attribute value to the class representing the certainty assessment (subproperty of **P39 measured**) since this solution to model fuzziness of attribute assignments, considers the usage of a numeric value to represent how sure is an assignment. As we do not assign continuous values but discrete values, we used a controlled vocabulary (e.g. probably not, probably, definitely not, definitely and possibly) and, therefore, property **P40 observed Dimension.** This

mechanism enables us to assess a certainty to any arbitrary attribute assignment.

Therefore, certainty is finally described using class **pdc:CertaintyAssessment**, (equivalent to **E16 Measurement)** to reflect some actor`s activity by declaring the certainty on a specific attribute assignment defined by **pdc:AttributeAssignment** class (equivalent to **E13 Attribute Assignment**).The properties **pdc:hasCertaintyAssessment** / **pdc:isCertaintyAssessmentOf** (subproperty of **P39 measured**) related these two classes. To assign a discrete value to the certainty we use **pdc:assessedCertainty** / **pdc:isCertaintyAssessedBy** properties (subproperty of **P40 observed Dimension**). Finally, we create properties **pdc:assignedAttributeTo (**subproperty of **140 assigned attribute)** to link an attribute assignment with the attribute entity and **pdc:assigned (**subproperty of **141 assigned)** to relate the attribute assignment to the entity being attributed, Fig. 15.

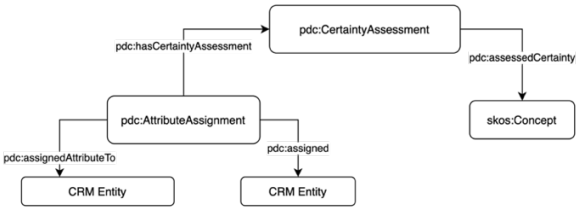

Fig. 15 Adopted structure to describe the certainty

In the example of Fig. 16 we show how to express that we have the certainty that the person (i.e., **ex:Person_1**) who is supposed to be the creator (i.e., **ex:AgentRole_1**) of the signs of a poetic work (i.e., **ex:Redaction_1**), definitely is not. For expressing the certainty of an arbitrary fact we have used the classes **pdc:AttributeAssignement** and **pdc:CertaintyAssessment.**The certainty of this assignment is assessed, and this fact is reflected using **pdc:assessedCertainty,** whose value corresponds to a concept that indicates we are sure that the assignment is not correct, that is, that is a wrong attribution to indicate that **ex:Person_1** plays the role of the creator with **ex:AgentRole_1**.

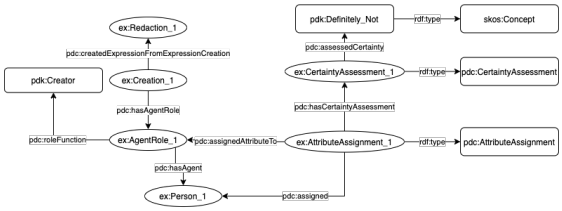

Fig. 16. Associated certainty for a Creator Role.

b) Other events of great importance in the domain of poetry are the birth and death of the authors. To express a person´s date of birth/death according to FRBRoo, we defined **pdc:Birth** and **pdc:Death** as equivalent to these classes **E67 Birth/ E69 Death**. In these events is necessary to identify at least the place, the date and the person affected by the event. Birth/death events are related to a place with the property, **pdc:tookPlaceAt** equivalent property of **P7 took place at.** The person affected by one of these events is related to it via the property **pdc:broughtIntoLive** equivalent property of **P98 brought into life** for a birth event and **pdc:wasDeathOf** equivalent property of **P100_was_death_of** in the case of death event. Finally, we express the date relation using the property **pdc:hasTimeSpan** equivalent to **P4 has time-span**. Fig. 17 shows an example of a birth event.

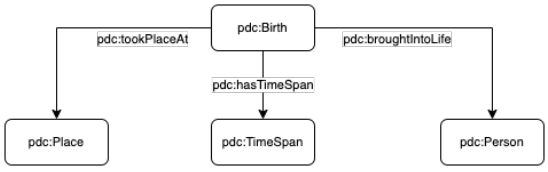

Fig. 17. Example of a birth event

5.2.2. Agents: Person, organization,

In the poetry domain knowledge, the concept of person and organization has a relevant use because they are related to the information of the author of the works. These two concepts are considered agents. FRBRoo makes specializations of CIDOC CRM classes to represent "real persons who lived or are assumed to have lived". (i.e., **F10 Person**). This description matches the semantic content in Postdata ontology, so we have defined a class **pdc:Person** as equivalent to **F10 Person,** Fig. 18.

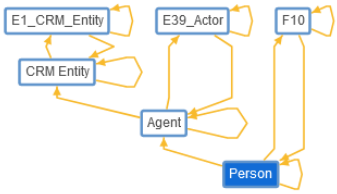

Fig. 18. Class hierarchy for **pdc:Person.**

The concept person has associated some data and object properties to describe the whole meaning in the poetry field. The properties supported in the ontology and their alignment with FRBRoo are in

On the other hand, FRBRoo also makes a specialization of CIDOC CRM classes to represent individuals that are groups of people and exhibit organizational characteristics (i.e., **F11 Corporate Body**). This meaning is not enough for the organizations represented in Postdata since we also have organizations like libraries. Therefore, our class **pdc: Organization**, a subclass of **E40 legal Body,** covers the definition of **F11 Corporate Body** and **F44 Bibliographic Agency**. To adopt these two semantic contents, we finally decided to make **pdc:Organisation** a subclass of **E74 Group**, Fig. 19.

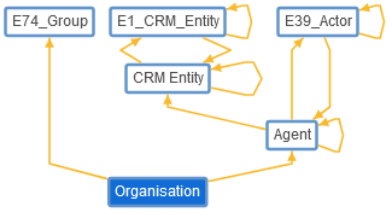

Fig. 19. Class hierarchy of **pdc: Organisation.**

A person's participation in one organization is expressed by the property **pdc:isMemberOf,** an equivalent property of **P107i_is_current_or_forer_member_of**.

Finally, both Persons and Organizations can participate as agents involving some action over an endeavour (e.g., creation, modification). Thus, we state a new class, **pdc:Agent** equivalent to **E39 Actor.** This class allows us to establish the right relations regarding the roles of agents in OntoPoetry Ontology.

5.2.3. Roles

In the poetry, ontology is needed to reflect the participation of some agent (i.e., a person or a group of people) in producing a specific element (i.e., poetic work) with a specific role. In this sense, the core module defines the class **pdc:AgentRole**. Since FRBRoo adopts an event-centric approach, the endeavours are the outcome of an event and are performed by actors. FRBRoo specification suggests the use of properties of properties to associate roles to the participants of the production events (i.e., creators). This approach is shown in Fig. 20 where an actor is associated to the creation by using **P14 carried out by** and in the same way, express roles following **P14.1 in the role of,** that relates and **E7 Activity** with **E55 Type** (the value of a controlled vocabulary specifying the type of role).

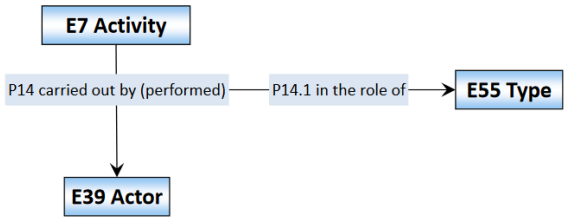

Fig. 20. Properties of properties to represent roles in CIDOC CRM.

However, this solution is not allowed in RDF [39]. Instead of using a property of properties, CIDOC CRM proposed two official modelling solutions to solve this issue in RDF [39]: a) to create one subproperty of **P14 carried out by** for each of the roles, or b) to create one intermediate class that links **P14 carried out by** with an instance of **E55 Type.** CIDOC CRM recommendation is to use the first approach because it is more aligned with the model and relaxes query complexity and runtime. However, when having several values for roles in a concept scheme, this solution is not useful since it is necessary to know beforehand all the possible values to create the subproperties. Therefore, for this situation is more useful to adopt the second approach and specify a class that represents the role assignation, Fig. 21.

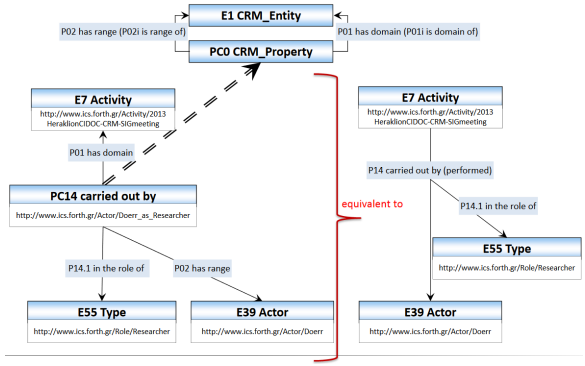

Fig. 21. Adopted solution to model roles in CIDOC CRM

The proposal is to create a **PC0 CRM_Property** class to include the properties **P02 has range,**

and **P01 has domain** that will be used to associate the actor of an activity (i.e., a creation activity) and the associated expression or work. The class **PC14 carried out by**, which is a subclass of **PCO CRM_Property** class expresses the role-actor relationship allowing to use the property **P14.1 in the role of** to associate the role-actor class to the specific role (i.e., a controlled vocabulary). This adaptation to RDF of the FRBRoo model, following the CIDOC CRM proposal, facilitates mapping our ontological elements.

So far, this solution allows us to relate events to agents and assign them a role, and it is also necessary to relate the events to works and expressions. Therefore, we used the property **R16** to relate **F27 Work conception** with **F1 Work**, **R17** to relate **F28 Expression Creation** to **F2 Expression** and **R19** to related **F28 Expression Creation** to **F1 Work**, Fig. 22.

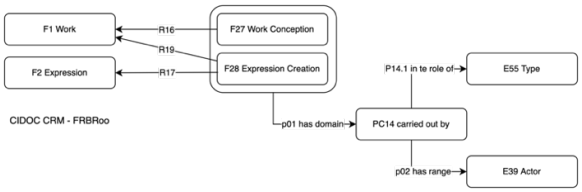

Fig. 22 CIDOC CRM solution to model roles and the relation with F1 Work and F2 Expression

The final alignment of the classes and properties used to describe the participation roles of agents in events that produced an endeavour is:

- The class **pdc:AgentRole** is a subclass of **PC14 carried out by**, which can be related to any activity that produced an outcome through **pdc:hasAgentRole** or the inverse **pdc:isAgentRoleOf**.
- Properties **pdc:roleFunction** and **pdc:hasAgent** are used to describe the role and the agent associated and are subproperties of **P14.1 in the role of** and **P02 has agent**.
- To express an activity related to a work, we defined **pdc:WorkConception** equivalent class of **F27 Work Conception**.
- For expressing activities related to an expression, we defined **pdc:ExpressionCreation** class, equivalent class of **F26 Expression Creation**.

Fig. 23 shows the domain and ranges of the classes used to express the agent-role structure.

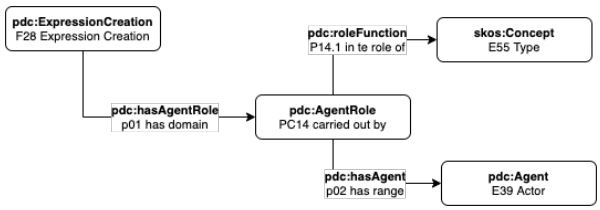

Fig. 23. Alignment of Agent Role properties for Expression Creation Event.

The relation between endeavours and events are modelled using:

- **pdc:initiated** is a subproperty of **R16 initiated**. Hence, it is used to relate the **pdc:WorkConception** activity with the instance of **pdc:Work** that was conceived or initiated. This is a way to reflect that the original idea of the work preceded its realization.
- **pdc:createdExpressionFromExpressionCreation** subproperty of **R17 created Expression from Expression Creation.** Similar to **pdc:initiated**, it relates an activity with the element produced during that activity. In this case, the **pdc:ExpressionCreation** with the instance of **pdc:Expression** produced. If the expression creation does not use **pdc:createdWorkByExpressionCreation**, this reflects a realization of a preconcepted work.
- **pdc:createdWorkByExpression-Creation** subproperty of **R19 created a realization of Work by Expression Creation.** Relates an instance of **pdc:ExpressionCreation** with an instance of **pdc:Work** that was concepted while the ideas of the work were performed. Using this property, we are making explicit the assumption that the ideas of the work and its signs are being produced simultaneously.

In those cases where the **pdc:roleFunction** of a **pdc:AgentRole** is a creator, we consider to be an instance of a more specific class**, pdc:Creator-Role**, which provides additional properties to describe authorship elements. We reflect this fact by expressing the following in the ontology

$$CreatorRole \equiv AgentRole \sqcap \exists roleFunction.Creator$$

For example, we can state that a **pdc:CreatorRole** is anonymous by using the object property **pdc:authorship** and indicating the appropriate concept of the authorship concept scheme. When the agent associated with a **pdc:CreatorRole** is a person, we could also indicate its education level in the context of the creation using **pdc:authorEducationLevel** and the interested concept of the appropriate concept scheme, Fig. 24.

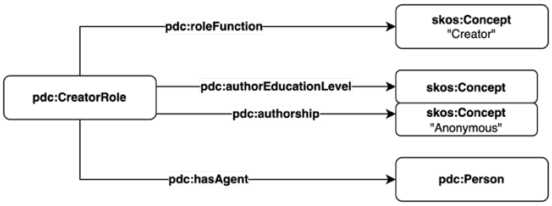

Fig. 24. Use of **pdc:creatorRole.**

5.3. Places.

Another relevant concept in the Postdata ontology is Place. The place is a concept used to locate both author and work and other properties related to poetry (i.e. literary text). In FRBRoo, the class to **F9 Place** has a similar semantic of the concept place used in our ontology. Therefore, a simpler alignment is possible. We defined the class **pdc:Place**, for which we stated to be equivalent to **F9 Place**, Fig. 25

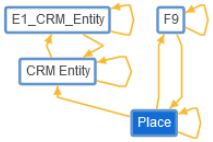

Fig. 25. Class hierarchy for **pdc:Place.**

5.4. Time expressions

Some properties related to events need to express information about time. In the ontology, we have considered two different approximations: a) when we want to express a period (i.e., time-span) and b) when one might just want to state a particular date of happening for some event.

In FRBroo is possible to use the class E52 Time-span to express the time-span of events or activities, but for its use in RDF, an adaptation is necessary [39]. Time-spam in CIDOC-CRM is considered and has two boundaries because they are treated as intervals, and this is something that cannot be represented using a single data type in RDF. For this reason, the CIDOC CRM RDF implementation guidelines [39] propose to use two properties to relate the time-span of an event with two different primitive values for dates of type **xsd:dateTime**:

- **P81 ongoing throughout** broken down into two properties to support RDF implementation:
  - **P81a_end_of_the_begin** should be instantiated as the earliest point in time the user is sure that the respective temporal phenomenon is indeed ongoing.
  - **P81b_begin_of_the_end** should be instantiated as the latest point in time the user is sure that the respective temporal phenomenon is indeed ongoing.
- **P82 at some time within** broken down into two properties to support RDF implementation:
  - **P82a_begin_of_the_begin** should be instantiated as the latest point in time the user is sure that the respective temporal phenomenon is indeed not yet happening.
  - **P82b_end_of_the_end should** be instantiated as the earliest point in time the user is sure that the respective temporal phenomenon is indeed no longer ongoing.

It may be highlighted that it may be necessary to round dates (i.e., 1971 = Jan 1 00:00:00 – Dec 31 23:59:59) up for the "a" subproperties and round down the "b" subproperties since the use of **xsd:dateTime** is recommended.

If we want to state a particular date of happening for some event, there is no need to make time inference, and it would be enough to state a date using **P78 is identified by**, which associates a time-span with a date appellation.

For these situations, we have defined the class **pdc:TimeSpan** as equivalent to **E52 Time-Spam** and the properties **pdc:hasTimeSpan**, **pdc:endOfTheBegin** and **pdc:beginOfTheEnd** as equivalent to **P4 has Time Span**, **P81a_end_of_the_begin** and, **P81b_begin_of_the_end.** For the case of a unique date, we have defined **pdc:date** property as equivalent to **P78 is identified by** property (see Fig. 26).

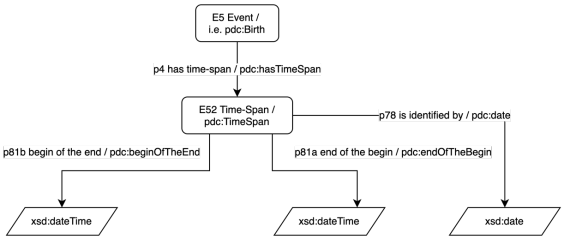

Fig. 26. Alignment of time-span and date in Postdata core.

An example of the usage of the time expressions is shown in Fig. 27. The concept we want to express is the existence of a poetic work, "Soneto I", created by Herrera in 1582, and we include the original signs of the poem (i.e., text) in Spanish.

Therefore, this schema shows that Soneto I (i.e., **ex:PoeticWork_1)** was conceived (i.e., **ex:AgentRole_1**) by Fernando de Herrera (i.e., **ex:Person_1**) and was produced (i.e., **ex:Creation_1)** at 1582 (i.e., **ex:TimeSpan_1**) in a given place (i.e., **ex:Place_1)**. The original text (i.e., **ex:Redaction_1**) is in Spanish.

We can say that this example is a expression creation activity identified by **ex:Creation_1**. The activity is located in time and space using **pdc:hasTimeSpan** and **pdc:tookPlaceAt** object properties. During this activity, "Fernando de Herrera" played the role of creator, which is modelled using an agent role structure connecting **ex:AgentRole_1** resource to the person via **pdc:hasAgent** and to the creator role using **pdc:roleFunction**. In addition, we have made use of a controlled vocabulary to reflect the associated role of the agent in the participation (i.e., **pdc:AgentRole**) through the property **pdc:roleFunction.** Several individuals of type **pdc:AgentRole** may be related to a creation event to reflect different participants and their associated roles.

Being **ex:Creation_1** an expression creation activity, we can make assumptions explicit using **pdc:createdWorkByExpressionCreation** and **pdc:createdExpression**. In this case, both previous object properties are used to indicate that the poetic work and its realization, identified by **ex:PoeticWork_1** and **ex:Redaction_1**, were produced simultaneously. Then, this yielded to the realization of the textual signs of the original manuscript, information that can also be used to categorize the redaction resource using **pdc:typeOfRedaction**.

5.5. Textual annotations related to Entities

In literary knowledge domain and, therefore, in poetry, it is necessary to get information about certain entities in a text or work or that a given text or work are related to an entity (i.e., poems inside an expression). A general way to model this is to create a subproperty of **P67 refers to,** to express that a text refers to another entity. It relates a redaction or another part of the redaction to the mentioned entity. We enable this representation by assigning **E33_Linguistic Object** as the domain of the property and **E1 CRM Entity** as the range. To this purpose, we reused these classes **E33 LinguisticObject**, and **E1 CRM_Entity** by using equivalence properties (**owl:equivalentProperty**).

The properties aligned as subproperties of **P67_refers_to** are:

- **pdc:mentions** to state that a redaction mentions an event, person, organization or place.
- **pdc:dedicatedTo** relates a **pdc:Redaction** to **a pdc:Person** to whom the current text was dedicated.
- **pdc:hasCharacter** relates a **pdc:Redaction** to **pdc:Character** equivalent to **F38 Charater**, depicted in the redaction's textual content.
- **pdc:hasNarrativeLocation** relates a **pdc:PoeticWork** to a **pdc:Place** in which the narrative of the poetic work takes place.
- **pdc:isIntendedFor** relates a **pdc:PoeticWork** or **pdc:Redaction** to a given agent that is the intended readership or receptor of the endeavour.
- **pdc:narrates** relates **a pdc:Excerpt** to the **pdc:Event** described in that precise text fragment.

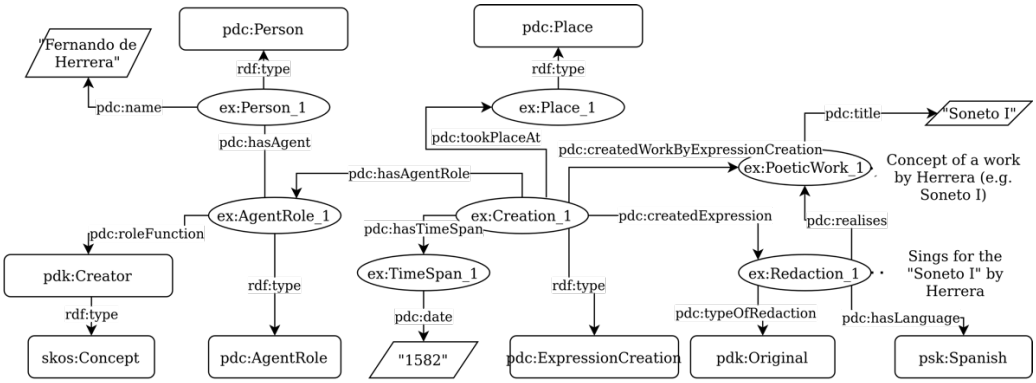

Fig. 27. Creation events and agent-roles

5.6. Concepts and Controlled vocabularies in the OntoPoetry-core ontology

Regarding the expression of concepts, FRBRoo – CIDOC CRM suggests using concepts from Knowledge Organisation Systems to classify entities according to several characteristics to indicate values of controlled vocabularies. For this reason, it is enough to indicate that **`skos:Concept`** is a subclass of **`E55 Type`**, Fig. 28

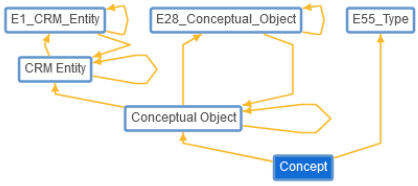

Fig. 28. Class hierarchy for **`skos:Concept`**.

The scope of the Postdata project is not to cover all the possible concepts of all the categories but to define the different categorizations. All these can be modelled using sub properties of **`P2 has type`** except **`pdc:hasLanguage`** that is modelled by using **`P72 is language of`**. It should consider that the object type does not need to be a **`skos:Concept,`** but we could use another URI that represents a class, for example, the Wikidata URI for a specific ethnicity.

5.7. Ontopoetry core module Controlled vocabularies

During the model process for the poetry ontology, we identified some data properties that, based on their content, we considered to define them as controlled vocabularies. The main goal of defining controlled vocabularies is unified terminology related to their content. In the Ontopoetry ontology core module, we have identified controlled vocabularies for the classes, PoeticWork, Redaction, Person, AgentRole and CreatorRole. The controlled vocabularies which refer to a PoeticWork are **genre**, **poeticType** and **authorEducationLevel**. These vocabularies help define, with a unified terminology, characteristics of the Poetic work. The vocabulary **poeticType**[16] aims to register the information about the major poetic genre of the text in verse, whether it is a poetic or dramatic text in verse. With this purpose, we have reused these two elements from the existent vocabulary "Literary forms and genres" from the UNESCO Thesaurus[17].

Then, the **genre** controlled vocabulary delimitates the specific subgenre to which the PoeticWork belongs. To build this vocabulary and we reused the "Tesauro de poesía medieval castellana"[18]. However, this existent vocabulary is restricted to a concrete period and literary tradition. For this reason, we have developed twenty-five new elements to include representation of most established poetic genres in both Spanish and English tradition and cover all the periods.

[16] https://github.com/linhd-postdata/skos-codelists/blob/master/poetic-type.ttl

[17] See the UNESCO vocabulary here: http://vocabularies.unesco.org/browser/thesaurus/en/page/concept327 (accessed: 3/1/2022)

The Postdata skos is available here: https://github.com/linhd-postdata/skos-codelists/blob/master/poetic-type.ttl (accessed: 3/1/2022)

[18] See this controlled vocabulary at: http://vocabularios.caicyt.gov.ar/pmc/index.php?tema=12&/tipo-de-poema-segun-forma (accessed: 3/1/2022)

Table 1. List of principal concepts in Ontopoetry.

Data Property	Subproperty	Class Domain
typeOfCharacter	P2_has_type	Character
function	P2_has_type	ConceptualObject
authorEducationLevel	P2_has_type	CreatorRole
typeOfDesignation	P2_has_type	CreatorRole
typeOfEvent	P2_has_type	Event
typeOfTextualElement	P2_has_type	LinguisticObject
ethnicity	P2_has_type	Person
floruit	P2_has_type	Person
literaryPeriod	P2_has_type	Person
nationality	P2_has_type	Person
socialStatus	P2_has_type	Person
gender	P2_has_type	Person
religiousAffiliation	P2_has_type	Person
typeOfPlace	P2_has_type	Place
literaryTradition	P2_has_type	PoeticWork
authorship	P2_has_type	PoeticWork
poeticType	P2_has_type	PoeticWork
audicenceEducationLevel	P2_has_type	Redaction
typeOfRedaction	P2_has_type	Redaction
audience	P2_has_type	Redaction
hasLanguage	P72 is language	Linguist Object

The properties **authorEducationLevel** and **audienceEducationLevel** describe the educational background of the author and audience. Subsequently, since they can be defined following the same parameters, we have defined one single vocabulary, namely **EducationLevelScheme**, for both. This controlled vocabulary[19] has four defined terms: higher education, Early childhood education, Primary education, and Secondary education; it is based on the UNESCO thesaurus of Education level.

If we move to the Redaction class, only one controlled vocabulary refers to it: typeOfTextualElement[20]. This vocabulary refers to the types of textual elements under consideration, namely, whether the text is a poem, a stanza, a chapter, an epilogue or a prologue.

The class Person has five defined, controlled vocabularies, namely **gender, literaryPeriod, socialStatus,** and **religiousAffiliation**. These properties enrich the profile of the Person. The use of pre-determined terms enables easily linking different persons based on characterizing elements as, for instance, which literary period they are part of. The first one, **gender**, defines the gender of the person with five possible choices[21]. The second one, **literaryPeriod**, indicates the contextualization of the author from the point of view of the Literature History, offering nine possible aesthetic trends[22]. The following one, socialStatus includes eleven social positions[23]. The next one, **religiousAffiliation**, classifies the person according to his or her religious beliefs. For this particular controlled vocabulary, we have reused the "Religious groups" controlled vocabulary from the UNESCO Thesaurus, which offers thirteen alternatives[24].

In the AgentRole class, three vocabularies were designated, namely, **roleFunction, typeOfCharacter** and **typeOfDesignation**. The first one, **roleFunction**, delimitates the specific function performed by some agent on a work, with six different ones[25]. For **typeOfCharacter**, there are many existent classifications. However, one of the most accepted in Literature Studies is the classic theory of Vladimir Propp (1968). For this reason, we decided to use this classification. According to it, seven terms have been defined for the **typeOfCharacter**[26] vocabulary. **typeOfDesignation** covers different types of names to include pseudonyms, heteronyms, allonym or orthonym variants, counting them with four elements[27].

Finally, we have defined the **certainty** controlled vocabulary related to events. **Certainty** is defined with five options that relate to how certain given information is, in other words, the degree of certainty[28]. Previously, another controlled vocabulary was

[19] https://github.com/linhd-postdata/skos-codelists/blob/master/author-education-level.ttl

[20] https://github.com/linhd-postdata/skos-codelists/blob/master/textual-element-type.ttl

[21] See the complete skos at our Github: https://github.com/linhd-postdata/skos-codelists/blob/master/gender.ttl (accessed: 3/1/2022).

[22] See the complete skos at: https://github.com/linhd-postdata/skos-codelists/blob/master/literary-period.ttl (accessed: 3/1/2022)

[23] See the complete skos at: https://github.com/linhd-postdata/skos-codelists/blob/master/social-status.ttl (accessed: 3/1/2022)

[24] See the UNESCO vocabulary at: http://vocabularies.unesco.org/browser/thesaurus/en/page/concept293 (accessed: 3/1/2022). See Postdata skos at our GitHub: https://github.com/linhd-postdata/skos-codelists/blob/master/religious-affiliation.ttl (accessed: 3/1/2022)

[25] See the complete skos at our GitHub: https://github.com/linhd-postdata/skos-codelists/blob/master/role-function.ttl (accessed: 3/1/2022)

[26] The complete list of terms is available at: https://github.com/linhd-postdata/skos-codelists/blob/master/character-type.ttl (accessed: 3/1/2022)

[27] See the complete skos at our GitHub: https://github.com/linhd-postdata/skos-codelists/blob/master/designation-type.ttl (accessed: 3/1/2022)

[28] https://github.com/linhd-postdata/skos-codelists/blob/master/certainty.ttl

included, namely," school", but it was, removed, at a later stage because it partially overlaps with the Literary Period and the affiliation of an author to a school rather than another can be an open discussion in literary criticism or even be controversial.

We illustrate the use of these controlled vocabularies using the example of Sonnet I by Fernando Herrera. From the poetic work point of view, this work is a "Poetry" poeticType, its genre is "sonnet", and the authorEducationLevel is "Higher education" since Fernando Herrera was considered an intellectual of this time. Its redaction is a "Poem" typeOfTextualElement since it includes the signs of the sonnet.

From the person point of view, Fernando Herrera was a male (i.e. gender), and his literaryPeriod would be Renaissance, socialStatus would be Priest, religiousAffiliation would be Christians. By having this information defined with a controlled vocabulary, it will be possible to easily connect Fernando de Herrera, for example, with other authors of the Renaissance, whether they have been characterized with the term "Renaissance" or with "Renacimiento".

In the case of this poem, the version published in Versos de Fernando de Herrera (1619) would have Francisco Pacheco with roleFunction collector, as he was the one who compiled the poems in the edition. Regarding typeOfCharacter vocabulary, there are not many characters in Herrera's poetry, but we can consider his lover, the countess Leonor, the protagonist of many poems. In typeOfCharacter, she would have the role of the princess or prize for the masculine hero.

5.8. Notes and Text

Notes provide definitions or explicative notes, and some elements may have different associated names or titles. We used a subproperty of **`P3 has note`** to associate an informal description about an entity. We also defined **`pdc:portrait`** property a subproperty of **`P3 has note`** to associate a person with an image denoted by an URL. We use **`xsd:anyURI`** as the range of this datatype property. We also defined the property **`pdc:biography`** to include a textual description of a person's life. Finally, we defined **`pdc:text`** to include a textual description of a redaction or a redaction fragment as a subpoperty of the **`P190_has_symbolic_content`**.

Table 2. Notes properties.

Property	Subproperty	Class Domain
text	P190_has_symbolic_content	Propositional object
portrait	P3_has_note	Person
biography	P3_has_note	Person

5.9. Datatype Properties

Here we show the modelling decisions we made to cover the datatype properties. We also grouped them according to how they are modelled or their function.

Datatype properties related to appellations

All the properties in this group are used as identifiers or signs to refer to instances. CIDOC CRM offers subclasses of **`E41 Appellation`** to reflect the assignment of an identifier to an entity. Entities are then related to appellations using **`P1 identifies,`** and appellations have textual content. In RDF, this results in having a URI to identify the textual element, which in most cases will result in the addition of an unnecessary entity, as is the case of this project. As we do not need to make special assertions about the textual entities themselves, but we are just interested in the textual content, we can get rid of the class appellation and make direct use of the subproperties of **`P1 identifies`** directly with literals,[39].

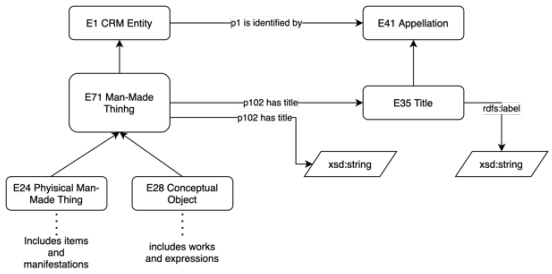

Fig. 29. Appellations

In Fig. 29 we can appreciate the assignation of appellations to entities following CIDOC CRM conceptual model, which employs a class to represent the title itself; and following the RDF implementation guidelines, which suggests we could use direct datatype properties instead if we consider it. As we do not further describe titles, we finally opted for the direct use of literals, but we would need to change the range of the properties to include literals and appellations, which does not make conceptual sense. For this reason, we redefined **`P1 is identified by`** as a datatype property that will act as the super-property of the rest of the datatype properties of this set of properties. The hierarchy of datatype properties used to provide names, titles, and date will also contain equivalent datatype properties of the needed subproperties of **`P1`**: **`P131 is identified by`** for persons, Table 3, **`P102 has title`** and **`P149 is identified by`** for endeavours, Table 4, **`P87 is identified`**

by for places, Table 5 and **P78 is identified by** for dates, Table 6.

Table 3. Data Properties for Person.

Property	Subproperty	Domain
additionalName	P131_is_identified_by	Person
alternativeName	P131_is_identified_by	Person
forename	P131_is_identified_by	Person
genName	P131_is_identified_by	Person
name	P131_is_identified_by	Person
nameLink	P131_is_identified_by	Person
originalName	P131_is_identified_by	Person
penName	P131_is_identified_by	Person
positionName	P131_is_identified_by	Person
surname	P131_is_identified_by	Person

Table 4. Data Properties for Endeavours

Property	Subproperty	Domain
alternativeTitle	P102_has_title	Propositional Object
originalTitle	P102_has_title	Propositional Object
subtitle	P102_has_title	Propositional Object
title	P102_has_title	Propositional Object
redactionNumber	P149_is_identified_by	Redaction

Table 5. Data Properties for Place

Property	Subproperty	Domain
address	P87_is_identified_by	Place
coordinates	P87_is_identified_by	Place

Table 6. Data Property for dates

Property	Subproperty	Domain
date	P78_is_identified_by	TimeSpam

Datatype properties that refer to quantities

These can be modelled using instances of **E54 Dimension,** and **P90 has value** datatype property to associate the value of the dimension as illustrated:

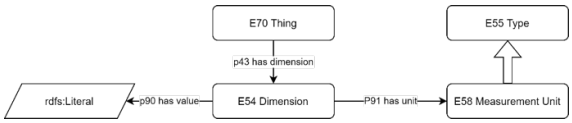

Fig. 30. Modelling dimensions in CIDOC CRM

In CIDOC CRM, a class can further describe the dimension characteristics. Instead, as we do not describe further the quantity measured, we will redefine **P43 has dimension** as a datatype property, following the same intuitions for the case of the appellations and the RDF Implementation Guidelines from CIDOC CRM. Hence, we defined **pdc:numberOfLines**, **pdc:numberOfPlays** and **pdc:numberOfPoems** datatype properties to indicate what we are "measuring" for an entity of type **pdc:Ensemble**.

5.10. Properties linking to Poetic Analysis Module

The OntoPoetry poetic analysis module represents different phenomena associated with metrics and prosody. This information is considered in the poetic analysis module as an event performed by an agent and produces a result. To model this event, we have used the class **pdp:ScansionProcess**. Therefore, all the scansion processes must be related to an expression of a poem (i.e. class **pdc:Redaction)** used as input for the scansion process. These relationships are realized using the property ***pdp:usedAsInput/*** **pdp:wasInputFor**.

5.11. Properties linking to Transmission Module

Finally, Ontopoetry deals with the relation between the conceptual or abstract work and a specific edition or manifestation. This is accomplished in the Transmission Module. Redactions are embodied in bibliographic products representing the ideal characteristics of the artefacts that result from a publication process and are also information objects carried by physical objects such as manuscripts, reproductions, or particular exemplars of a publication product. To link Ontopoetry core and transmission modules, we use the properties **pdt:embodies / pdt:isEmbodiedIn** that allow us to relate a Redaction with the Bibliographic Source that embodies its signs and **pdt:carries / pdt:isCarriedBy** that let us reflect which physical carrier (i.e. InformationCarrier, including Primary Source, Item and Facsimile) contains the information of a particular Redaction.

6. Conclusions and future work

This paper presents the methodology and the description of Ontopoetry Ontology, an ontology devoted to poetry developed in the Postdata (Poetry Standardization and Linked Open Data) ERC project, which aims to provide a means for poetry researchers to publish and consume semantically-enriched data in the context of European poetry. After a critical analysis of Postdata Ontology V1.0 (i.e., the first version of the Ontopoetry based on the analysis of 25 repertories) made by experts on ontology engineering and poetry,

we built Ontopoetry ontology, a new poetry ontology completely aligned with FRBRoo ontology by Reusing, merging, and re-engineering ontological resources and the Reuse of ontology design patterns process based on the NeOn Methodology. Ontopoetry consists of three modules: a) the core module, b) the poetic analysis module, and c) the transmission module. In this work, we have presented the core module.

This module presents the abstract or conceptual side of the bibliographic information and includes all the essential information that characterizes works and their expressions, irrespective of their physical materializations. The choice of the main classes of this module, Poeticwork, Ensemble and Redaction and their relationships, allows us to model the complexity derived from the knowledge that it is needed to preserve in a literary study. A poem is considered a work, but it is represented using different expressions and, even depending on the original author´s idea, could be interpreted as part of another work. Therefore, the options are wide, and the expressivity of the ontology must capture it. Together with the ontology definition, we have defined a set of controlled vocabularies that will help better semantic comprehension and give some uniformity in the descriptions of some concepts. The core module is connected by some object properties to the poetic analysis module and transmission module, conforming these three modules the complete description of the Ontopoetry ontology. The last modules will be described in future work.

Ontopoetry ontology has been used to build an extent poetry knowledge graph containing corpora in Czech, English, French, Italian, Portuguese, and Spanish. A total of twelve corpora with 3,847,739 verses are available with different levels of granularity, but all of them are annotated to a certain extent. This knowledge graph could be consulted after we finish the populate and enrichment process and the front-end development before the end of the project.

Besides developing the Ontopoetry ontology, the Postdata project is also developing the PoetryLab API. PoetryLab API is an extensible open-source API for poetry analysis. At this moment, it can perform the syllabification, scansion (extraction of stress patterns), enjambment detection (syntactical units split into two lines), rhyme detection, and medieval named entity recognition for Spanish poetry. This API achieves the state of the art performance in the tasks for which reproducible alternatives exist. Moreover, it is designed to aggregate as many tools as needed for poetry analysis. PoetryLab API uses Ontopoetry ontology as its underlying data model, and it uses RDF triples according to this ontology as source and generates new RDF triples as results. So Ontopoetry ontology is currently the data persistence and interoperability base of PoetryLab API.

The description of Ontopoetry ontology is available at https://postdata.linhd.uned.es/results/ontopoetry-v2-0/.

7. Acknowledgements

The authors acknowledge the support of the Starting Grant research project Poetry Standardization and Linked Open Data: Postdata (ERC-2015-STG-679528). This project is funded by the European Research Council (ERC) under the research and innovation program Horizon2020 of the European Union.

8. References

[1] E. González-Blanco and L. Seláf, “Megarep: A comprehensive research tool in medieval and renaissance poetic and metrical repertoires,” *Humanitats a la xarxa: món medieval/Humanities on the Web: The Medieval World*, pp. 321–332, 2014.

[2] E. González-Blanco García, M. Manailescu, and S. Ros, “From syllables, lines and stanzas to linked open data: standardization, interoperability and multilingual challenges for digital humanities,” 2016, Accessed: January. 08, 2022. [Online]. Available: http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez11

[3] L. Leonardi, “Repertorio della tradizione poetica italiana dai Siciliani a Petrarca.” http://www.mirabileweb.it/ (Accessed: January. 08, 2022).

[4] L. Grijp, “Dutch Song Database.” http://www.liederenbank.nl/index.php?lan=en (Accessed: January. 08, 2022).

[5] E. González-Blanco García and J. L. Rodríguez, “ReMetCa: a TEI based digital repertory on Medieval Spanish poetry,” Jan. 2013, Accessed: January. 08, 2022. [Online]. Available: http://e-spacio.uned.es/fez/view/bibliuned:365-Egonzalez6

[6] M. Colombi, “Pedecerto,” *Pedecerto. Metrica Latina Digitale*.

http://www.lutessa.it/pedecerto/public/ (Accessed: January. 08, 2022).
[7] M. Brea, "MedDB - Inicio," *MedDB: Base de datos da Lírica profana galego-portuguesa*. http://bernal.cirp.gal/ords/f?p=MEDDB3:2 (a Accessed: January. 08, 2022).
[8] S. Asperti and F. Zinelli, "Bibliografia Elettronica dei Trovatori - v.2.5 - 2012." http://www.bedt.it/BEdT_04_25/index.aspx (Accessed: January. 08, 2022).
[9] E. González-Blanco, G. del Rio Riande, and C. Martínez Cantón, "Linked Open Data To Represent Multilingual Poetry Collections. A Proposal To Solve Interoperability Issues Between Poetic Repertoires," in *Proceedings 5th Workshop on Linked Data in Linguistic*, Paris, Jun. 2016, pp. 77–80. doi: 10.5281/zenodo.60767.
[10] H. Bermúdez-Sabel, M. Curado Malta, and E. Gonzalez-Blanco, "Towards Interoperability in the European Poetry Community: The Standardization of Philological Concepts," in *Language, Data, and Knowledge*, 2017, pp. 156–165.
[11] H. Bermúdez-Sabel, M. L. Díez Platas, S. Ros, and E. González-Blanco, "Towards a common model for European Poetry: Challenges and solutions," *Digital Scholarship in the Humanities*, Dec. 2021, doi: 10.1093/llc/fqab106.
[12] E. Gonzalez-Blanco and S. Ros, "POSTDATA – Poetry Standardization and Linked Open Data," *Postdata*, 2016. http://postdata.linhd.es/ (a Accessed: January. 08, 2022).
[13] M. Curado Malta, E. Gonzalez-Blanco, C. Martinez Cantón, and G. del Rio Riande, "A Common Conceptual Model for the Study of Poetry in the Digital Humanities," in *Digital Humanities 2017. Conference Abstracts*, Montréal, Canada: McGill University / Université de Montréal, 2017, pp. 210–212. Accessed: January. 08, 2022 [Online]. Available: https://dh2017.adho.org/abstracts/148/148.pdf
[14] M. Curado Malta, E. González-Blanco, C. I. Martínez Cantón, and G. del Rio Riande, "Digital repertoires of poetry metrics: towards a linked open data ecosystem," in *Digital Humanities and Digital Curation. Proceedings of the First Workshop on Digital Humanities and Digital Curation co-located with the 10th Conference on Metadata and Semantics Research (MTSR 2016).*, 2016. Accessed: January. 08, 2022 [Online]. Available: http://ceur-ws.org/Vol-1764/1.pdf
[15] P. Bootz and S. Szoniecky, "Toward an ontology of the Field of Digital Poetry," Sep. 2008, Accessed: January. 08, 2022. [Online]. Available: https://hal-univ-paris8.archives-ouvertes.fr/hal-01011128
[16] "DHQ: Digital Humanities Quarterly: Ontologies and Logic Reasoning as Tools in Humanities?" http://www.digitalhumanities.org/dhq/vol/3/4/000068/000068.html (Accessed: January. 08, 2022).
[17] D. B. Tillett, "What is FRBR? A conceptual model for the bibliographic universe," *The Australian Library Journal*, vol. 54, no. 1, pp. 24–30, Feb. 2005, doi: 10.1080/00049670.2005.10721710.
[18] "6 Verse - The TEI Guidelines." https://www.tei-c.org/release/doc/tei-p5-doc/en/html/VE.html (Accessed: January. 08, 2022).
[19] Postdata ERC project, "TOWARDS A NETWORK OF ONTOLOGIES FOR THE EUROPEAN POETRY - POSTDATA." http://postdata-prototype.linhd.uned.es/methodology.php (Accessed: January. 08, 2022).
[20] Postdata ERC project, "POSTDATA Repertories," *Google My Maps*. https://www.google.com/maps/d/viewer?mid=15MAs3lVHlOkeWUfBWXBP_prHbE (Accessed: January. 08, 2022).
[21] Ma. L. Diez Platas *et al.*, "Description of Postdata Poetry Ontology V1.0," *Plotting Poetry:Plotting Poetry: On Mechanically-Enhanced Reading*, to publish 2021.
[22] M. d'Aquin, A. Schlicht, H. Stuckenschmidt, and M. Sabou, "Criteria and Evaluation for Ontology Modularization Techniques," in *Modular Ontologies*, vol. 5445, H. Stuckenschmidt, C. Parent, and S. Spaccapietra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 67–89. doi: 10.1007/978-3-642-01907-4_4.
[23] P. Doran, V. Tamma, and L. Iannone, "Ontology Module Extraction for Ontology Reuse: An Ontology Engineering Perspective," in *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, New York, NY, USA, 2007, pp. 61–70. doi: 10.1145/1321440.1321451.
[24] B. C. Grau, "A Logical Framework for Modularity of Ontologies.," presented at the International Joint Conferences on Artificial Intelligence i, 2007. Accessed: January. 08, 2022 [Online]. Available: http://ijcai.org/Proceedings/07/Papers/046.pdf
[25] H. Bermúdez Sabel, "Towards interoperability in the European poetry community," presented at the Shaping Data in Digital Humanities,

Denmark, Apr. 20, 2018. doi: 10.5281/zenodo.2020302.
[26] M. C. Suárez-Figueroa, "NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse," phd, Facultad de Informática (UPM), 2010. Accessed: January. 08, 2022. [Online]. Available: http://oa.upm.es/3879/
[27] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology for Ontology Engineering," in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–34. doi: 10.1007/978-3-642-24794-1_2.
[28] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology for Ontology Engineering," in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–34. doi: 10.1007/978-3-642-24794-1_2.
[29] V. A. Carriero *et al.*, "The Landscape of Ontology Reuse Approaches," in *Studies on the Semantic Web*, G. Cota, M. Daquino, and G. L. Pozzato, Eds. IOS Press, 2020. doi: 10.3233/SSW200033.
[30] H. Sofia Pinto and J. P. Martins, "Reusing Ontologies," in *AAAI 2000 spring symposium series*, Palo Alto, CA, 2000, pp. 77–84. [Online]. Available: Accessed: January. 08, 2022 https://link.springer.com/content/pdf/10.1007/s10115-003-0138-1.pdf
[31] CIDOC, "CIDOC CRM," Accessed: January. 08, 2022. http://www.cidoc-crm.org/.
[32] FRBRoo, "FRBRoo," *FRBRoo Funtional Requirements for Biliografic Records*, 2021. http://www.cidoc-crm.org/frbroo/home-0 (Accessed: January. 08, 2022).
[33] M. Doerr and C. Bekiari, "FRBROO, A CONCEPTUAL MODEL FOR PERFORMING ARTS," Accessed: January,7, 2022. https://projects.ics.forth.gr/_publications/drfile.2008-06-42.pdf. 2008.
[34] E. Montiel-Ponsoda, G. Aguado de Cea, A. Gómez-Pérez, and M. C. Suárez-Figueroa, "Helping Naive Users to Reuse Ontology Design Patterns," in *The Semantic Web: Research and Applications: 5th European Semantic Web Conference, ESWC 2008*, Tenerife, España, Jun. 2008, vol. Lectur. Accessed: January. 08, 2022. [Online]. Available: http://oa.upm.es/5193/
[35] V. Presutti *et al.*, "D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies.," http://neon-project.org/deliverables/WP2/NeOn_2008_D2.5.1.pdf. (Accessed: January. 08, 2022)2008.
[36] "FRBRoo Specification." http://www.cidoc-crm.org/frbroo/sites/default/files/FRBRoo_V2.4.pdf Model at http://www.cidoc-crm.org/frbroo/sites/default/files/FRBR2.4-draft.rdfs (Accessed: January. 08, 2022).
[37] M. Doerr, "Issue 395: Symbolic Content," *CIDOC CRM*, Oct. 06, 2017. Accessed: January. 08, 2022 http://www.cidoc-crm.org/Issue/ID-395-symbolic-content
[38] R. CIDOC CRM, "Certainty and belief values issue CIDOC CRM." 2017. Accessed: Accessed: January. 08, 2022. [Online]. Available: http://www.cidoc-crm.org/Issue/ID-349-belief-values
[39] M. Doerr, R. Light, and G. Hiebel, "Implementing the CIDOC Conceptual Reference Model in RDF," Accessed: January,7,2022. [Online]. Available: https://cidoc-crm.org/sites/default/files/issue%20443%20-%20Implementing%20CIDOC%20CRM%20in%20RDF%20v1.1.pdf. Oct. 2020.

9. ANNEX 1

Data Properties, Object properties and Classes OntoPoetry Core Module.

Table 7. OntoPoetry Core Module Classes.

Class	Type Alignment	Alignment
Activity	equivalentClass	E7_Activity
Agent	equivalentClass	E39_Actor
AgentRole	equivalentClass	PC14_Carried_Out_By
Ensemble	equivalentClass	F17 AggregationWork
AttributeAssignment	equivalentClass	E13_Attribute_Assignment
Birth	equivalentClass	E67_Birth
Character	equivalentClass	F38
ComplexWork	equivalentClass	F15
ConceptualObject	equivalentClass	E28_Conceptual_Object
CRM_Entity	equivalentClass	E1_CRM_Entity
Death	equivalentClass	E69_Death
Event	equivalentClass	E5_Event
Expression	equivalentClass	F2
ExpressionCreation	equivalentClass	F28
ExpressionFragment	equivalentClass	F23
poeticwork	equivalentClass	F14 IndividualWork
LinguisticObject	equivalentClass	E33_Linguistic_Object
Measurement	equivalentClass	E16_Measurement
Organisation	subClassOf	E74_Group
Period	equivalentClass	E4_Period
Person	equivalentClass	F10
Place	equivalentClass	F9
PropositionalObject	equivalentClass	E89_Propositional_Object
redaction	equivalentClass	F22 SelfContainedExpression
TimeSpan	equivalentClass	E52_Time-Span
Work	equivalentClass	F1
WorkConception	equivalentClass	F27
skosConcept	subClassOf	E55 concept
creatorrole	subClassOf	AgentRole
excerpt	subClassOf	LinguisticObject, ExpressionFragment
CertaintyAssement	subClassOf	E16_Measurement

Table 8. Ensemble Properties

Property	Subproperty	Domain	Range
numberOfLines	P43_has_dimension	Ensemble	integer
numberOfPlays	P43_has_dimension	Ensemble	integer
numberOfPoems	P43_has_dimension	Ensemble	integer

Table 9. Person Properties

Property	Subproperty	Domain	Range
additionalName	P131_is_identified_by	Person	string
alternativeName	P131_is_identified_by	Person	string
forename	P131_is_identified_by	Person	string
genName	P131_is_identified_by	Person	string
name	P131_is_identified_by	Person	string
nameLink	P131_is_identified_by	Person	string
originalName	P131_is_identified_by	Person	string

penName	P131_is_identified_by	Person	string
positionName	P131_is_identified_by	Person	string
surname	P131_is_identified_by	Person	string
biography	P3_has_note	Person	string
portrait	P3_has_note	Person	anyURI

Table 10. Place Properties

Property	Subproperty	Domain	Range
address	P87_is_identified_by	Place	string
coordinates	P87_is_identified_by	Place	string

Table 11. Propositional Objects Properties

Property	Subproperty	Domain	Range
alternativeTitle	P102_has_title	PropositionalObject	string
originalTitle	P102_has_title	PropositionalObject	string
subtitle	P102_has_title	PropositionalObject	string
title	P102_has_title	PropositionalObject	string
text	P190_has_symbolic_content	LinguisticObject	string

Table 12. Time Spam properties

Property	Subproperty	Domain	Range
date	P78_is_identified_by	TimeSpan	date
beginOfTheEnd	P81b_begin_of_the_end	TimeSpan	dateTime
endOfTheBegin	P81a_end_of_the_begin	TimeSpan	dateTime

Table 13. Agent Role Properties.

Property	Subproperty	Domain	Range
hasAgent	P02 has range	AgentRole	Agent
isagentof	P02 has range	Agent	AgentRole
isAgentRoleOf	P01_has_domain	AgentRole	Activity
roleFunction	P14.1_in_the_role_of	AgentRole	skosConcept
hasAgentRole	P01i_is_domain_of	Activity	AgentRole
authorEducationLevel	P2_has_type	CreatorRole	skosConcept
typeOfDesignation	P2_has_type	CreatorRole	Concept

Table 14 Event Properties.

Property	Subproperty	Domain	Range
broughtIntoLife	P98_brought_into_life	Birth	Person
wasDeathOf	P100_was_death_of	Death	Person
tookPlaceAt	P7_took_place_at	Period	Place
narrates	P67_refers_to	Excerpt	Event
narratedBy	P67i_is_referred_to_by	Event	Excerpt
typeOfEvent	P2_has_type	Event	skosConcept

Table 15 Linguistic Object properties.

Property	Subproperty	Domain	Range
hasLanguage	P72_has_language	LinguisticObject	Concept
isLanguageOf	P72i_is_language_of	Concept	LinguisticObject
typeOfTextualElement	P2_has_type	LinguisticObject	Concept
mentions	P67_refers_to	LinguisticObject	CRM_Entity
isMentionedIn	P67i_is_referred_to_by	CRM_Entity	LinguisticObject

Table 16. Propositional Object.

Property	Subproperty	Domain	Range
isAbout	P129_is_about	PropositionalObject	CRM_Entity
isSubjectOf	P129i_is_subject_of	CRM_Entity	PropositionalObject
isIntendedFor	P67_refers_to	PropositionalObject	Agent
isTargetAudienceIn	P67i_is_referred_to_by	Agent	PropositionalObject

Table 17. Conceptual Object property

Property	Subproperty	Domain	Range
function	P2_has_type	ConceptualObject	Concept

Table 18. Person Object Properties

Property	Subproperty	Domain	Range
gender	P2_has_type	Person	skosConcept
religiousAffiliation	P2_has_type	Person	skosConcept
ethnicity	P2_has_type	Person	skosConcept
floruit	P2_has_type	Person	skosConcept
literaryPeriod	P2_has_type	Person	skosConcept
nationality	P2_has_type	Person	skosConcept
socialStatus	P2_has_type	Person	skosConcept
diedIn	P100i_died_in	Person	Death
isMemberOf	P107i_is_current_or_former_member_of	Person	Organization
hasDedication	P67i_is_referred_to_by	Person	Redaction
wasBorn	P98i_was_born	Person	Birth

Table 19. Place Object Properties

Property	Subproperty	Domain	Range
witnessed	P7i_witnessed	Place	Period
isNarrativeLocationOf	P67i_is_referred_to_by	Place	PoeticWork
typeOfPlace	P2_has_type	Place	skosConcept

Table 20. PoeticWorks Object Properties

Property	Subproperty	Domain	Range
hasNarrativeLocation	P67_refers_to	PoeticWork	Place
isSynthesisOf	R2	PoeticWork	PoeticWork
hasSynthesis	R2i	PoeticWork	PoeticWork
literaryTradition	P2_has_type	PoeticWork	Concept
authorship	P2_has_type	PoeticWork	skosConcept
poeticType	P2_has_type	PoeticWork	skosConcept

Table 21. Redaction Object Properties

Property	Subproperty	Domain	Range
hasCharacter	P67_refers_to	Redaction	Character
dedicatedTo	P67_refers_to	Redaction	Person
typeOfRedaction	P2_has_type	Redaction	skosConcept
audience	P2_has_type	Redaction	skosConcept
audienceEducationLevel	P2_has_type	Redaction	Concept

Table 22. Time Object Properties.

Property	Subproperty	Domain	Range
hasTimeSpan	P4_has_time-span	Period	TimeSpan
isTimeSpanOf	P4i_is_time-span_of	TimeSpan	Period

Table 23. Work-Work object properties

Property	Subproperty		Domain	Range
hasWork	R10	ComplexWork	Work	
isWorkOf	R10i	Work	ComplexWork	
isDerivativeOf	R2	Work	Work	
hasDerivative	R2i	Work	Work	

Table 24. Work-Expression Object properties

Property	Subproperty	Domain	Range
isRealisedInSelfContainedExpression	R3	Work	SelfContainedExpression
realisesWork	R3i	SelfContainedExpression	Work
hasRepresentativeExpression	R40	Work	SelfContainedExpression
isRepresentativeExpressionFor	R40i	SelfContainedExpression	Work
isRealisedThrough	R9	PoeticWork	Redaction
realizes	R9i	Redaction	PoeticWork

Table 25. Expression-Expression Object Properties.

Property	Subproperty	Domain	Range
edits	P130 shows features of	redaction	redaction
isEditedIn	P130 shows features of	redaction	redaction
hasFragment	R15	Redaction	ExpressionFragment
isFragmentOf	R15i	ExpressionFragment	Redaction
hasTranslation	P73_has_translation	Redaction	Redaction
isTranslationOf	P73i_is_translation_of	Redaction	Redaction
incorporates	R14	SelfContainedExpression	Redaction
isIncorporatedIn	R14i	Redaction	SelfContainedExpression

Table 26. Certainty Object Properties.

Property	Subproperty	Domain	Range
assignedAtributeTo	P140 assigned attribute	AttributeAssignment	CRM entity
wasAtributedby	P140i assigned attribute	CRM entity	AttributeAssignment
assigned	P141_assigned	AttributeAssignment	CRM_Entity
wasAssignedBy	P141i_was_assigned_by	CRM_Entity	AttributeAssignment
isCertaintyAssessmentOf	P39_measured	CertaintyAssessment	AttributeAssignment
hasCertaintyAssessment	P39i_was_measured_by	AttributeAssignment	CertaintyAssessment
assessedCertainty	P40_observed_dimension	CertaintyAssessment	skosConcept
isCertaintyAssessedBy	P40i_was_observed_in	skosConcept	CertaintyAssessment

Table 27. Organization Object Property

Property	Subproperty	Domain	Range
hasMember	P107_has_current_or_former_member	Organization	Person

Table 28. Character Object Properties.

Property	Subproperty	Domain	Range
typeOfCharacter	P2_has_type	Character	skosConcept

characterIn	P67i_is_referred_to_by	Character	Redaction

Table 29. Event-Work-Expression Object Properties.

Property	Subproperty	Domain	Range
createdExpression	R17	ExpressionCreation	Expression
wasCreatedByExpressionCreationForExpression	R17i	Expression	ExpressionCreation
initiated	R16	WorkConception	Work
wasInitiatedBy	R16i	Work	WorkConception
createdWork	R19	ExpressionCreation	Work
realisedThroughExpressionCreation	R19i	Work	ExpressionCreation