Improving Semantic Expressiveness in Data Catalogs: Applying Ontological Patterns to Data Catalog Vocabulary Relations

Vânia Borges[1,*], Natália Queiroz de Oliveira[1], Maria Luiza Machado Campos[1], and Giseli Rabello Lopes[1]

[1] *Programa de Pós-Graduação em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil*

Abstract

On the Web, cataloged resources can be related in many ways. Complex relationships may be necessary to characterize the context in which the resources were created, allowing for tracing input data, the software used, and the agents and funders involved. However, describing these relationships expressively and formally, contributing to the semantics of the resources associated, and facilitating interoperability is still a challenge. To avoid inconsistencies and ambiguities between different interpretations of relations, it is essential to use common terminology. The Data Catalog Vocabulary (DCAT) is a W3C-recommended schema used for catalog interoperability, modeling the data structures of relevant resources along with their primary relationships. This work proposes applying ontological patterns based on the Unified Foundational Ontology (UFO) to improve the semantic expressiveness of domain-specific relationships that are not currently offered in DCAT. The use of these patterns offers additional details about interactions among those involved and, when needed, documents the evolution of relationships over time, contributing to understanding, reuse, and interoperability.

Keywords

Data Catalog Vocabulary, Relations, Relationships, Ontological Patterns.

1. Introduction

Data catalogs have been gaining prominence in the literature as a solution for increasing visibility and access to cataloged resources [1,2]. They are collections of metadata, combined with data management and search tools, that assist analysts and other data users in finding the data they need [3]. Serving as an inventory of available resources, they provide information to evaluate the fitness of data for intended uses [4]. Their metadata are organized into schemas, also known as metadata models, which capture information about various aspects of a cataloged resource [5], including how they relate to each other. These cataloged resources often originate from diverse data repositories, encompassing a wide array of datasets that require proper description and discoverability. Complex relationships may be necessary to describe the context in which these resources were created, enabling tracing of input data, the software used, and the agents and funders involved [6]. However, describing these relationships expressively and formally, contributing to the semantics of the associated resources, and facilitating interoperability remains a challenge.

In the catalog domain, Data Catalog Vocabulary (DCAT) is a W3C recommendation for catalog interoperability [7]. It has been used in different implementations such as: the DCAT Application Profile (DCAT-AP) [8] that serves as a standard for describing public sector datasets across Europe; the GeoDCAT-AP that represents geographic metadata in European data portals [9]; and the core

[1] *Proceedings of the 18th Seminar on Ontology Research in Brazil (ONTOBRAS 2025) and 9th Doctoral and Masters Consortium on Ontologies (WTDO 2025), São José dos Campos (SP), Brazil, September 29 – October 02, 2025.*

* Corresponding author.

vjborges30@ufrj.br (V. Borges); natalia.oliveira@ppgi.com.br (N. Q. Oliveira); mluiza@ppgi.com.br (M. L. M. Campos); giseli@ic.ufrj.br (G. R. Lopes)

0000-0002-6717-1168 (V Borges); 0000-0001-8371-142X (N. Q. Oliveira); 0000-0002-7930-612X (M. L. M. Campos); 0000-0003-2482-1826 (G. R. Lopes)

of the FAIR Data Point schema, which facilitates the publication of FAIR-compliant metadata for catalogued resources [10]. DCAT describes cataloged resources on the Web with an emphasis on datasets and data services. Thus, a publisher can describe their datasets and data services in a catalog using a standard model and vocabulary that facilitates the consumption and aggregation of metadata from various catalogs [7]. DCAT offers a set of relations between resources and proposes solutions for representing domain-specific relations, which include the use of *dcat:Relationship*. According to DCAT, *dcat:Relationship* applies to a specific set of relationships, i.e., those in which one resource plays a role with respect to another. This type of relation, known as role-playing relation, is common in social or organizational situations, where the dynamic between the parties is mediated by specific roles that each party plays [11]. This work focuses on using ontological patterns to enhance the expressiveness and formalism of these relations, providing mechanisms for catalogs to support the dynamic creation of these relationships at the schema level and, as a result, for reuse by resource publishers.

Patterns are instruments for encapsulating common knowledge that can contribute to the analysis of different concepts and types of relations, thereby improving representation and providing support for machine-actionability for catalogs. In the Software Engineering community, the term “pattern language” refers to a network of interrelated patterns together with a process for systematically solving coarse-grained software development problems [12]. This approach has been successfully applied in ontology engineering through the development of ontology patterns (OPs). OPs are an emerging approach that benefits the reuse of encoded experiences and good practices [13], giving rise to ontology pattern languages (OPLs). The literature highlights different contexts that explore OPs to enhance semantic expressiveness. For instance, in multidimensional models used in the representation of analytical data in data warehouses [14]; in OWL ontologies, with an alignment between the I-ADOPT framework and the OPs established by Measurement OPL (M-OPL) [15]; and in guiding the definition of exploratory questions for conceptual models, guaranteeing pragmatic explanations in relation to the model [16].

Recent work has presented a systematic analysis of truthmaking patterns (TMP) for relations, based on the ontological nature of their truthmakers, which are the entities responsible for the truth of the propositions arising from the relationships [11]. It presents several TMPs for relations with different levels of expressivity. As ontological patterns, these TMPs help define concepts and relationships, speeding up ontology development and encouraging reuse.

This work proposes the adoption of the TMP and powertype pattern for improving the semantics of the role-playing relations in DCAT. The goal is to enhance the semantics of DCAT by using TMP for descriptive relationships. Instead of relying on relationships with embedded semantics in the data, a pattern is suggested to guide the catalog administrators in creating domain-specific relations and their truthmakers (relationships). These relations, shown in the schema, become reusable by publishers. The truthmakers provide details about interactions among those involved and, when needed, document how these relationships evolve over time. Adding this information helps agents better understand relationships and improves interoperability.

This paper is organized as follows: Section 2 presents the background of the paper. Section 3 describes how DCAT handles relations and relationships. Section 4 applies the truthmaking pattern to *dcat:Relationship*. Section 5 addresses a simple implementation in OWL to explore the potential of the TMP. Finally, in Section 6, we conclude and list some future work.

2. Ontological patterns for relations in UFO

The terms "*relation*" and "*relationship*" are often used interchangeably in the literature. This paper considers the distinction proposed by Guarino and Guizzardi [17]. According to the authors, a *relation* holds because the *relationship* exists. In this case, the authors identify the relationship as the truthmaker of the relation, i.e., what establishes the truth of the propositions derived from this relationship.

The UFO relation taxonomy utilizes two orthogonal distinctions of relations that consider internal/external and descriptive/non-descriptive relations [11, 18]. Given the objective of this paper, we focus here on external-descriptive relations. In UFO, material relations are external descriptive relations that hold in virtue of at least one *relational moment* inhering in at least one *relatum* that is existentially dependent on another *relatum*. They can be single-sided whenever they hold in virtue of one or more moments inhering in just one *relatum* (e.g., <administratorOf>); or multi-sided whenever they hold in virtue of at least two moments, each inhering in a different *relatum* (e.g., <treatedIn> between a patient and a medical unit) [11].

Guarino et al. [11] propose a systematic analysis of truthmaking patterns for relations, based on the ontological nature of their truthmakers. In their work, they present a number of TMP for relations according to levels of expressivity. Before proceeding, it is essential to discuss an important notion, namely, the distinction between strong and weak truthmakers. A strong truthmaker is one whose existence is sufficient for a proposition to be true. In contrast, a weak truthmaker makes a proposition true not merely because of its existence, but because of the way it contingently is. In this paper, we focus on the TMP for external descriptive relationships, especially role-playing relations [11]. In this type of relation, an entity plays a relational role that emerges due to the existence of another entity. To illustrate, Figure 1 depicts the TMPs for this kind of descriptive relation, as represented in Figure 1(a). A weak TMP is illustrated in Figure 1(b), where the material relation <administratorOf> between an admin and a catalog is derived from the relator <Administration>. This relator accounts for the social commitments and obligations associated with the administrative role <Admin>, which depends on some catalog. Figure 1(c) shows the full TMP, which adds the event <AdministrationEvolution>. The event is a strong truthmaker and accounts for the period of time during which the role is played.

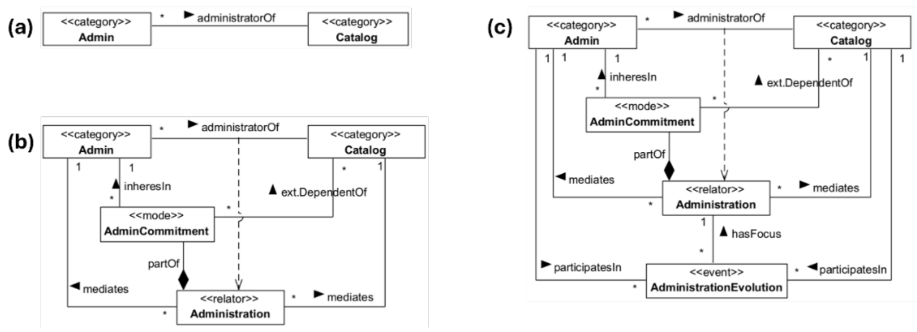

Figure 1: Weak and full truthmaking patterns for a single-sided relation.

The powertype pattern is a well-known pattern in the conceptual modeling community and is relevant to this research. This pattern addresses a phenomenon that occurs in various subject domains that require the handling of multiple classification levels [14, 19]. It is an example of an early approach for multi-level modeling in software engineering, used to model situations in which instances of a type (the power type) are specializations of a lower-level type (the base type), and both power types and base types appear as regular classes in the model. Multi-Level Theory (MLT) [20] is an axiomatic theory based on UFO that provides powertype support in UML [19]. It uses the <<instantiation>> stereotype to highlight the association between the base type and the higher-order type, establishing an instantiation semantics. According to the theory, the cardinality between the types involved establishes distinct behaviors for instances of the base type.

Figure 2 shows an example where each instance of <Agent> will be an instance of, at most, one instance of the higher-order type, in this case, an instance of <Person> or <Organization>. For a complete description of the approach, see [19].

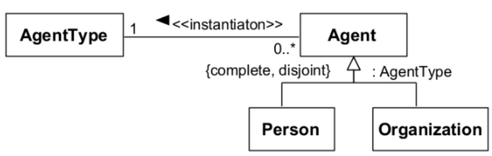

Figure 2: Relation between higher-order type and base type.

3. Relations and relationships in DCAT

DCAT is a metadata vocabulary implemented in OWL 2 that reuses terms from standardized vocabularies, such as Dublin Core (DC), Friend Of A Friend (FOAF), and Provenance Ontology (PROV-O). Additionally, it defines a minimal set of classes and properties of its own [21]. Figure 3 shows a UML diagram of the DCAT entities associated with the definition of qualified relationships. The model adopts the following prefixes: *dcat* for DCAT concepts, *foaf* for Friend Of A Friend vocabulary, *skos* for Simple Knowledge Organization System, and *dct* for Dublin Core Terms. In the figure, *dcat:Resource* represents the cataloged resources, i.e., resources published or curated by a single agent. According to DCAT, this class should not be instantiated, but rather its specializations.

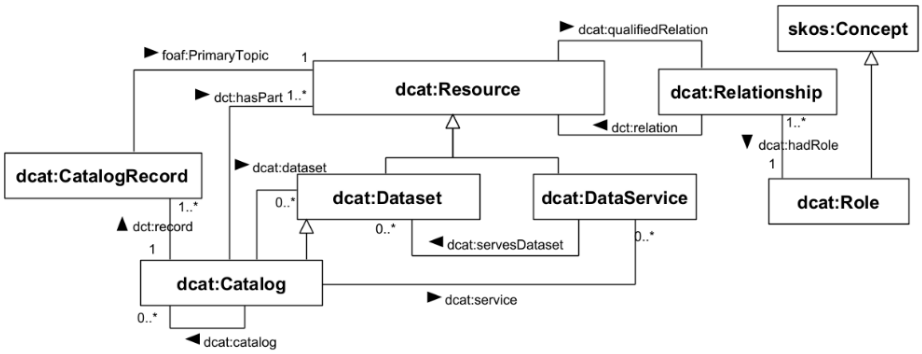

Figure 3: Simplified view of DCAT resources and qualified relationships.

In addition to the relations defined between entities, DCAT also includes classes to facilitate the creation of new ones. This flexibility is necessary because resources can be related in many ways. Furthermore, complex relationships may be needed to characterize the context in which the resources were created, for example, by tracing input data, the software used, and the agents and funders (sponsors/financiers) involved [6].

DCAT offers qualified relations to support complex non-binary relations not covered by PROV-O and DCAT properties. These qualified relations can also be convenient when relations are represented using known properties but have additional information needs that require a more sophisticated representation [6]. For example, one might want to describe the temporal dimension of a function, i.e., the period during which an individual or organization performed a certain function.

To create relations across resources, DCAT utilizes the *dcat:qualifiedRelations* property [6]. This property links the source resource to an instance of the *dcat:Relationship*. In the DCAT specification, *dcat:Relationship* is "an association class for attaching additional information to a relationship between DCAT Resources" [21]. It involves another resource referenced by the *dct:relation* property, which, in the context of a *dcat:Relationship*, must point to another *dcat:Dataset* or another cataloged resource [6]. The *dcat:Relationship* uses the *dcat:hadRole* property to link to *dcat:Role*. The *dcat:Role* class has two functions in the specification. It provides the meaning of the agent responsibility regarding the Resource and the role or function of an Entity concerning another provided Entity [6].

According to the current schema, new relationships are represented along with the instances (data), making it difficult to standardize and reuse relations in the model. According to Albertoni *et al.* [21], these expected relationships can be complex and must be addressed. For DCAT, associating a term from a semantic artifact is sufficient to provide the semantics of these relationships. However, it does not establish the nature of the relationship or provide a means of understanding it. The following sections explore the TMP for descriptive relations and the powertype pattern to assist in the implementation of new relations.

4. Applying the truthmaking pattern for descriptive relations

The *dcat:Relationship* entity was introduced into DCAT from its second version onwards to enable the representation of relationships between datasets and other resources. According to the DCAT, it applies to a specific set of relations, i.e., those in which one resource plays a role with respect to another. As aforementioned, this type of relation, known as role-playing relation, is common in social or organizational situations, where the dynamic between the parties is mediated by specific roles that each party plays [11].

One approach to representing role-playing relations involves using relators [11]. In this approach, the relator expresses the social commitments and obligations associated with the roles that individuals or entities play in a given context. Reification of the relator, as the truthmaker of these relationships, provides a more explicit structure for understanding the interactions. Additionally, explicitly defining the roles played by entities helps establish the cardinalities of the relationships, thereby avoiding ambiguities in conceptual modeling. Another important aspect is the dynamism that can occur in this type of relationship over time, depending on the circumstances and actions of the parties involved. In this case, as presented in the full TMP, events can be relevant to documenting the properties evolution of these relationships [11].

Based on the definitions of relation and relationship mentioned in Section 2, we consider that entities under the superclass *dcat:Relationship* are those whose instances serve as truthmakers for relations between resources. These entities are categorized as relators in UFO, corresponding to the mereological sum of external dependent aspects of at least one entity involved. As an association class[2], it also allows modelers to define relationship-specific properties. As a superclass, *dcat:Relationship* establishes mandatory properties for distinct relationship types in the domain. Accordingly, it is classified as a UFO category.

To standardize and define new relationships and relations across resource types at the schema level, it is essential that the value associated with *dcat:Role* be explicitly expressed in the model rather than alongside the data. In this way, entities that specialize in *dcat:Relationship* become carriers of this relational aspect that refers to one of the entities involved in the relationship. To achieve this, it is necessary to deal with *dcat:Role* as a higher-order type. It is worth noting that UFO distinguishes between first-order types (1stOTs) and high-order types (highOTs) based on their level of abstraction and categorization of entities represented. While 1stOTs represent individual concrete objects, highOTs represent categories of 1stOTs or other highOTs.

4.1. dcat:Role as high-order type

Based on the DCAT specification, *dcat:Role* instances are terms in a semantic artifact that specify the meaning of an entity role concerning another or an agent role regarding an entity. This definition corresponds to "the position or purpose that someone or something has in a situation, organization, society, or relationship" as stated in the Cambridge Dictionary[3]. In this context, it represents a relevant aspect of an entity that is externally dependent on another. Therefore, we consider *dcat:Role* an external dependent aspect representing a relational property of an agent or source resource that depends on another resource. External dependent aspects define the properties that an individual holds in the scope of a certain material relation [23].

Addressing semantic overload and distinguishing the ontological nature of *dcat:Role*, this paper considers it as a superclass to denote distinct roles and functions based on their application: *ResourceRole* and *AgentRole*. Each of these new quality types is linked to its own value space. These value spaces – referred to as quality structures - are abstract entities delimiting the range of possible values (qualia, singular quale) for qualities of a given quality type [24]. Therefore, each specialization of *dcat:Role* defines a conceptual space that must be managed by a nominal quality structure [25]. Each potential value can be a term with an independent meaning. Many semantic

[2] An Association that has a set of Features of its own [22].
[3] https://dictionary.cambridge.org/us/dictionary/english/role

artifacts, such as thesauri and taxonomies, have structural relations that software agents can utilize to identify synonyms and equivalents with other artifacts.

4.2. A pattern for relationship types

Based on the distinction between 1stOTs and highOTs, we propose a pattern for role-playing relations in DCAT using TMP, *dcat:Role* and *dcat:Relationship*. It should be noted that the use of entities such as *dcat:Role* to link terms that indicate the meaning of the role or function performed by another entity is not exclusive to DCAT. Other vocabularies, such as Bibliographic Framework Initiative[4] (BIBFRAME) and Organization Ontology[5] (ORG), use classes in similar ways.

Our pattern outlines the relations between *relata*, where at least one aspect of a *relatum* depends externally on another. Consequently, it permits the specification of a relationship between two or more resources where an aspect of one resource (*ResourceRole*) is externally dependent on another resource(s). The full TMP of interest comprises a relator and an event. The relator connects the involved entities (related resource types) and adds characteristics of the *relata* that are externally dependent on another, in this case, the role of the resource. This relator may also possess its own characteristics. One or more material relations between the entities involved will be derived from the instantiated relator. This way, the relator makes the semantics of the material relation explicit.

Here, we propose that entities aligned with the pattern are modeled as highOTs, aiding catalog modelers in defining new relationships and relations. These entities are shown in Figure 4. The models presented in this paper were implemented using the Visual Paradigm[6] tool version 17.2. This tool has a plugin[7] for OntoUML, an ontology-driven modeling language that incorporates the distinctions underlying UFO into UML class diagrams [24]. It introduces various stereotypes that correspond to the concepts defined in UFO, as well as grammatical formal constraints that reflect its axiomatization.

Using the powertype pattern, the cataloged resource type (*ResourceType*) is powertype of *dcat:Resource* entity. As a result, all the specializations of this entity become instances of the *ResourceType*, and instances of *dcat:Resource* must be an instance of, at most, one instance of resource type. Related resource type (*RelatedResourceType*) and resource type by role (*ResourceTypeByRole)* are specializations of *ResourceType. ResourceTypeByRole* identifies domain resource types according to their role and categorizes *dcat:Resource*. In this context, its instances are a set of *dcat:Resource* specializations that assume a role in relation to other resources. The *RelatedResourceType* classifies specific resource types in the domain that are involved in relationships. When made explicit at the schema level, the instances of these highOTs are anti-rigid types whose contingent classification condition is relational. Therefore, they are categorized as <Rolemixin>.

RelationshipType is the powertype of *dcat:Relationship*. It is rigid and categorized as <Category>. *RelationshipTypeByRole* specializes *RelationshipType* and categorizes *dcat:Relationship*. Its instances are relators that mediate resource types and are truthmakers for role-playing relations between them. In defining relationships, *RelationshipTypeByRole* is a sortal, and its instances must be types classified as <Relator> that corresponds to the mereological sum of the external dependent moment of at least one resource type, including the *ResourceRole* instance.

[4] https://www.loc.gov/bibframe/
[5] https://www.w3.org/TR/vocab-org/
[6] https://www.visual-paradigm.com/
[7] https://github.com/OntoUML/ontouml-vp-plugin

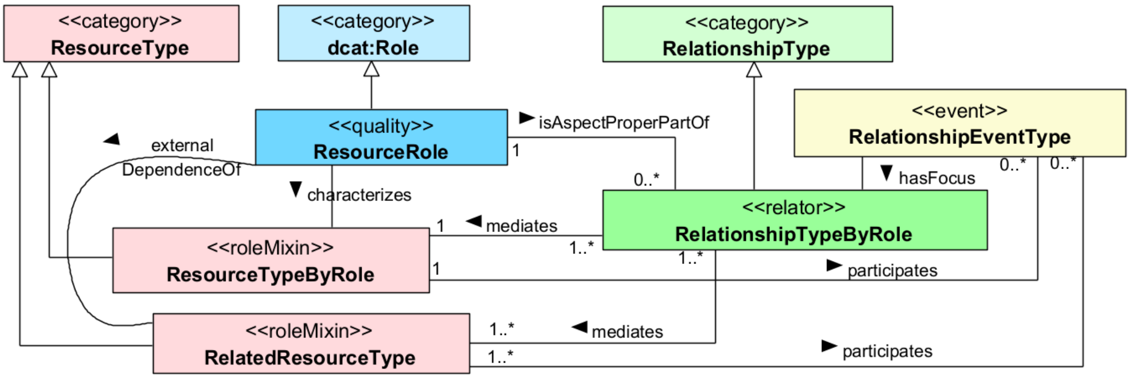

Figure 4: Full pattern for relationships among resource types in ML-DCAT.

The full TMP also employs events. For the DCAT relationships, the events can work as relational episodes [17]. In this context, focusing on the relator, an event allows us to follow the manifestation of certain aspects defined in the relationship [17]. Instances of its instances can, for example, follow the time period during which a function or role is performed. Thus, the *RelationshipTypeEvent* entity is defined as an <Event Type> that focuses on *RelationshipTypeByRole*. Consequently, it captures the creation of relationship types and the manifestation of specific properties (aspects) of those relationships at a given time. The way the episode reports participation varies based on the relationship specific properties. According to Guarino, Sales and Guizzardi [11], these aspects are considered the "focus" of the event, while the relationship itself is the focus of the relational episode. It is through the relationship that the event is understood and interpreted. Instances of *RelationshipEventType* are created only when changes in types are significant to the domain. They allow temporal and contextual metadata to be associated with the relationship, such as property changes over time in a domain relationship, as well as the responsible agent. This work has adopted behavior similar to that of highOT endurants for high-order events, including the use of intra- and cross-level structural relations [20]. Thus, the model incorporates *RelationshipEvent* as the base type of the *RelationshipEventType* powertype. Similar to other DCAT entities, its instance depends on the level of change monitoring that the catalog intends to employ.

The *ResourceRole* entity (*dcat:Role* specialization) is an external dependent moment, an aspect, within the pattern. It characterizes *ResourceTypeByRole* and depends on *RelatedResourceType*, both entities mediated by *RelationshipTypeByRole*, which formalizes and makes explicit this dependency as a mereological sum of externally dependent aspects. According to multi-level theory, the entity *ResourceRole* is a ***regularity property***, i.e., it is an attribute defined in a highOT that influences the intension of instances of this type [20]. Aligned with this vision, *ResourceRole* fulfills the DCAT proposal, establishing the meaning of the relationship, and contributes to the intension of the material relations dynamically defined at the schema level. In the *RelationshipTypeByRole* instances, the *ResourceRole* instances are standardized values listed in Quality Structures. They capture the role of the resource regarding another, offering the meaning for relationships and relations that will be instantiated. The modeler can also insert relevant characteristics about the resources that influence the relationship, as well as those specific to the relationship itself. These characteristics can also change over time, depending on the contextual circumstances of the relationship.

Adopting the pattern provides the modeler with additional resources to analyze the addressed concepts, offering tools to improve their representation. The pattern also allows the modeler to define grounded role-playing relations. Furthermore, it is possible to observe an evolution of the concept associated with the *dcat:Relationship* class. This class is no longer just a UML association class and is now treated as a relator. While the former is identical to an instance of an association, i.e., an objectified n-tuple, the relator grounds those n-tuples, in the sense that its instances represent what happens in reality whenever the association holds [26]. Associations, in turn, are represented by material relations derived from relators.

To define new material relations, the following steps must be taken: (i) identify the types of resources to be connected; (ii) choose a term from a standardized vocabulary that accurately represents the semantics of the resource role; (iii) clarify (made explicit) the resource role as a new type, if needed; (iv) model the relationship by specializing *dcat:Relationship*; (v) add valued attributes into the object facet of the relationship instance; (vi) define additional properties to the class facet, if needed; (vii) associate the involved resources; (viii) link the resource types, defining one or more material relations; and (ix) evaluate whether an event that works as a mechanism for recording and monitoring the evolution of the relationship over time is needed. These relations must be linked to the relator from which they derive. It is important to note that external dependence relations and their reified relators address cardinality issues and the ambiguity between association specialization, subsetting and redefinition [27, 28].

5. Employing Well-founded Relations among Resources

To illustrate the use of the pattern for role-playing relations, we present part of a model developed for a healthcare catalog that describes research datasets on patient clinical data and its operational ontology. These datasets were generated from electronic medical records and processed to fit into a research case record form. To ensure provenance, metadata from the workflow applied in the process was created and packaged in a complementary dataset. Similarly, metadata from the extraction, processing, and conversion to the form was also produced and exported to another dataset. These datasets contain, respectively, the prospective and retrospective provenance metadata of the clinical data dataset. To publish these datasets and the relationships between them, the catalog uses DCAT and the pattern presented in Section 4.

In this section, we present the conceptual model implemented to represent the relationship and its transformation to an OWL ontology, which functions as a well-founded semantic model for the catalog. This semantic model is published in the same triplestore where the dataset descriptors are published, providing different agents with relevant information about the model and the existing data. For this implementation, we used the Systematic Approach for Building Ontologies (SABiO), a methodology that enables the development of operational ontologies from a reference ontology — a conceptual model that clearly and accurately describes the entities in the domain [29]. Following this methodology, the conceptual model is developed using Visual Paradigm and the OntoUML plugin. With the plugin, the conceptual model is translated into a structure defined by gentle UFO (gUFO), a lightweight version of UFO implemented in OWL 2 to support the design of well-founded operational ontologies [30]. It is then exported as a serialized OWL file in Turtle (TTL). To assist modelers, two applications implemented in Jupyter Notebook are used to adjust the ontology. These applications will be discussed in a future paper. The ontology is imported into Protégé[8] tool, where adjustments and validations with plugins and reasoners are performed. Once completed, it is published in the GraphDB[9], a triplestore used to store (meta)data in triples.

5.1. Conceptually modeling new relationships

Figure 5 illustrates a UML model with a relationship where one dataset works as the provenance dataset for the others, meaning its data represents the provenance metadata for the related datasets. According to the figure, a dataset can have more than one dataset with provenance, covering prospective and retrospective provenance metadata. On the other hand, a dataset can contain provenance metadata from more than one dataset. The highOTs are represented with the stereotype <<type>> for enhanced visibility. In the model, DCAT classes retain the prefix *dcat*; classes referring to the relation pattern are associated with the prefix *mldcat*, in reference to multi-level entities for DCAT. Domain-specific relationships use the prefix *mycat*. The prefix *datacite_voc* refers to the Datacite[10] vocabulary. The provenance dataset is explicitly defined as a specialization

[8] https://protege.stanford.edu/
[9] https://graphdb.ontotext.com/
[10] http://purl.org/datacite/v4.4/

of the dataset. *mycat:ProvenanceRelationship* represents the *mldcat:RelationshipTypeByRole* instance, with *datacite_voc:isMetadataFor* outlining the relationship intension. The term means that the data in a dataset is metadata for another resource, serving to represent the semantics of the *mycat:ProvenanceDataset*. The *mycat:isProvenanceMetadataFor* is a material relation derived from this relator. Based on the pattern, the term refers to the provenance dataset (intrinsic aspect) and is externally dependent on the existence of the dataset to which it relates. Attributes can be defined in the relator. Therefore, the relationship can indicate that *dataset* A partially covers the provenance of *dataset* B and fully covers that of *dataset* C, while also recording the level of confidence in that provenance.

In the model, the event for documenting changes adds the agents involved. It enables tracking changes to provenance metadata about the datasets, documenting updates to the catalog, such as changes to the confidence level of the provenance and the agent responsible for them. Similarly, the same treatment can be offered to improve agent attributions regarding resources in DCAT. This topic has not been explored here due to the limited number of pages.

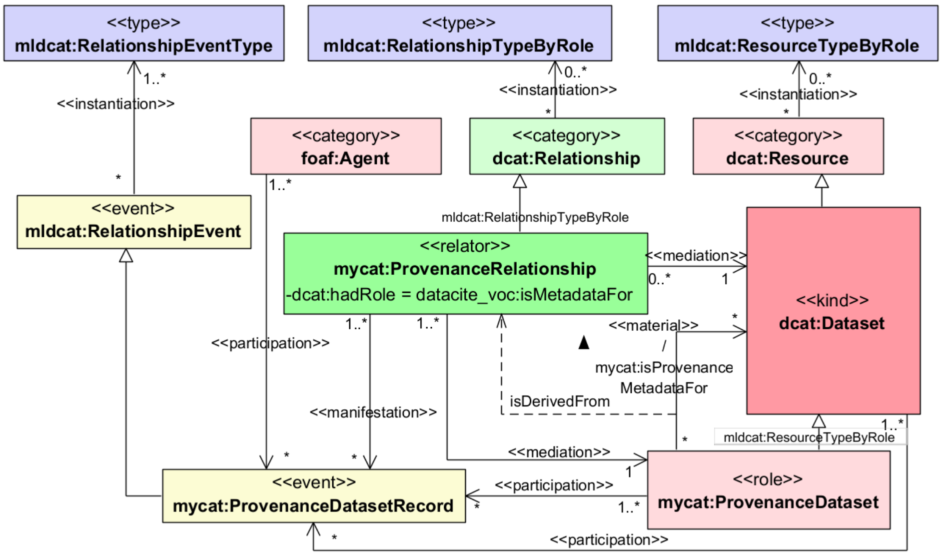

Figure 5: Example of Material Relation across Datasets using Truthmaking Pattern.

The pattern also applies to other domains, such as portals like OASIS BR[11] — a Brazilian portal that brings together scientific production and research data in open access, published on different digital platforms. In this case, it can define the relationship between the aggregator catalog (Portal) and the platforms it refers to. This relationship can include attributes such as periodicity and harvest type, which may change over time, such as switching the harvest periodicity from monthly to fortnightly. In this case, the event serves as a record mechanism to track these changes.

It should be noted that accidental or contingent roles played by types enhance the model semantics when made explicit. These types can be suppressed in the design phase to ensure the metadata schema fulfills non-functional requirements. To aid this process, UFO offers a method that establishes rules for abstraction [31].

5.2. Publishing the Semantic Model and Descriptors in the Catalog Triplestore

After validations made in Protégé, the ontology file (semantic model) is imported into the GraphDB triplestore along with the descriptors (cataloged resources), instances of the model. Figure 6 presents a simplified graph view of the relationship and material relation illustrated in Figure 5, published in a GraphDB triplestore. The categories of a foundational ontology, such as UFO, establish a common language and a referential model that can be used to describe the schema types and their relations [32]. Based on a solid theoretical foundation, these definitions are accessible through SPARQL queries, contributing to a clear and coherent representation of knowledge.

[11] https://oasisbr.ibict.br/vufind/

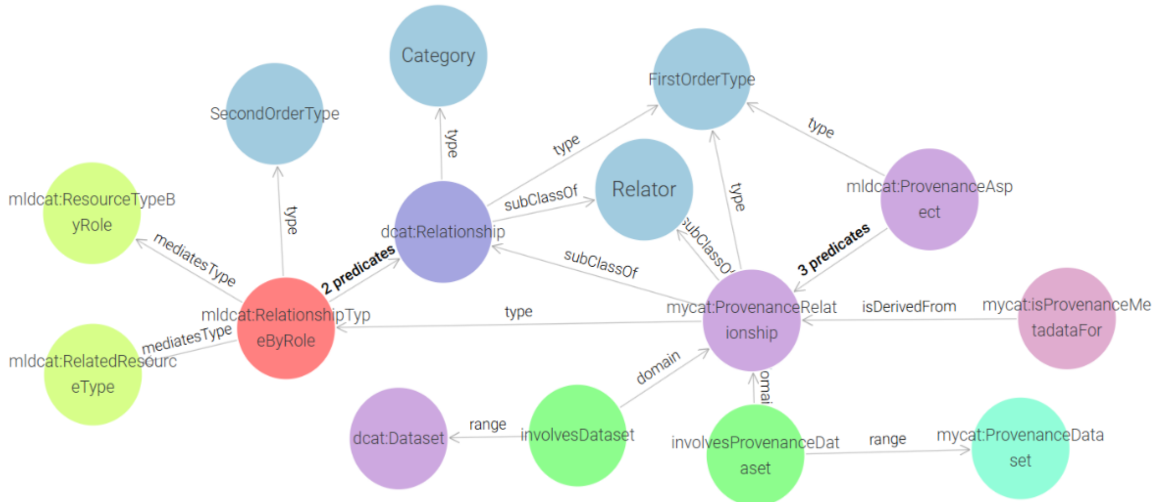

Figure 6: Relation and Relationship graph view using OWL.

SPARQL queries can be used to describe domain-specific relations, following the pattern for external descriptive relations. The formalism of the pattern enables the description of the entities involved, the relation, and its truthmaker. Figure 7 shows a SPARQL query that collects the elements involved in the pattern and the query results. This way, the schema, as a semantic model, provides relevant information for understanding domain-specific relations.

```
SELECT DISTINCT ?relationship ?entityByRole ?relatedEntity ?roleMeaning
?materialRelation ?inverseRelation
  WHERE {
    ?relationship rdf:type/rdfs:subClassOf* mldcat:RelationshipTypeByRole .
    ?resourceRole mldcat:resourceRole ?relationship;
                  mldcat:hadResourceRoleValue ?roleMeaning .
    ?materialRelation gufo:isDerivedFrom ?relationship ;
                      rdfs:domain ?entityByRole;
                       rdfs:range ?relatedEntity .
    OPTIONAL
         {?inverseRelation owl:inverseOf ?materialRelation}
  } ORDER BY ?relationship
```

	relationship	entityByRole	relatedEntity	roleMeaning	materialRelation	inverseR...
1	mycat:ProvenanceRelationship	mycat:ProvenanceDataset	dcat:Dataset	datacite_voc:isMetadataFor	mycat:isProvenanceMetadataFor	

Figure 7: SPARQL Query to list domain-specific relations and relationships.

Recent research has experimented with the use of ontological patterns to explain the elements of ontology-driven conceptual models [16, 32]. Similarly, it is possible to extend the use of patterns to provide a view of the entities in the catalog schema. Therefore, by combining the information from the results in Figure 7 and using the pattern as a foundation, it is possible to design a SPARQL query to explain the material relations in the model. Figure 8 illustrates this example. Access to this information is useful for publishers who can select, from the relationships offered by the catalog, the one that best suits their needs or even ask the administrator to create a new one.

	materialRelation	explanation	inverseRelation
1	mycat:isProvenanceMetadataFor	"The domain-specific relation mycat:isProvenanceMetadataFor is a material relation among resource types derived from mycat:ProvenanceRelationship. In this relation, instances of mycat:ProvenanceDataset plays the role/function of <datacite_voc:isMetadataFor> for instances of dcat:Dataset"	

Figure 8: Explaining material relations in a Catalog.

The explanations can be extended to published descriptors. Thus, descriptors that assume a certain role because they are involved in domain-specific relationships may have an explanation associated with them, as shown in Figure 9. The figure shows an explanation for the workflow provenance dataset (Hospital_Dataset_WS_Provenance) and for the data transformation process provenance dataset (Hospital_Dataset_ETL_Provenance). Based on the pattern, it can be inferred that these resources are datasets that perform a specific role, defined as *mycat:ProvenanceDataset* for another dataset that contains patient clinical data according to the case record form (Hospital_Dataset_CRF). This function is represented by the relationship *mycat:isProvenanceMetadataFor*, derived from the relationship mycat:ProvenanceRelationship. Change logs stored as events could also be presented, demonstrating changes in provenance records over time.

	resource	explanation
1	ex:Hospital_Dataset_WS_Provenance	"The resource <Hospital_Dataset_WS_Provenance_Metadata> is a <dcat:Dataset> that plays the role of <mycat:ProvenanceDataset> for <Hospital_Dataset_CRF> through the <mycat:isProvenanceMetadataFor> relation, derived from <mycat:ProvenanceRelationship> relationship."
2	ex:Hospital_Dataset_ETL_Provenance	"The resource <Hospital_Dataset_ETL_Provenance_Metadata> is a <dcat:Dataset> that plays the role of <mycat:ProvenanceDataset> for <Hospital_Dataset_CRF> through the <mycat:isProvenanceMetadataFor> relation, derived from <mycat:ProvenanceRelationship> relationship."

Figure 9: Explaining dataset instances according to their role domain-specific relations.

The additional information provided by patterns offers insights into the semantic model of the catalog for agents. In particular, the truthmaker of relations gives a more precise understanding of the resources involved. Using SPARQL queries as semantic mechanisms, it is possible to validate new relations, ensuring the schema's consistency. These relations, formally and semantically expressed, can be reused by different resource publishers.

6. Conclusion and future work

This paper presented truthmaking and powertype patterns to enhance DCAT role-playing relations, thereby improving its semantics. As a result, instead of relying on embedded semantics at the instance level, a pattern was introduced to guide catalog administrators in creating dynamically domain-specific relations and their truthmakers at the schema level, contributing to the accurate representation of resources.

The introduction of multi-level types and an ontological foundation enhances the expressiveness and formalism of relations expressed using DCAT. Together, they provide means for defining, validating, explaining, and comparing domain-specific relationships. It is worth mentioning the treatment of highOTs as endurants, i.e., as entities with modal properties that can change qualitatively while remaining the same [33]. For instance, this enables the identification of resource types that play specific roles in relation to others. Software agents can handle the complexity introduced, making it transparent to researchers, catalog managers, and other users, as demonstrated with SPARQL queries.

A case in the area of clinical health data was used to demonstrate the use of an ontology that extends DCAT, adding highOTs that allow the definition of new relations and their relationships with their respective associated meanings. The publication of the data, along with the schema, in a triplestore such as GraphDB enabled us to illustrate the potential uses of ontological foundations and the adoption of ontological patterns in metadata schemas for catalogs. According to Auer [34], publishing the semantic model alongside the data represents an improvement in semantics, aiding in its contextualization. Additionally, the ontological foundation provided by UFO and its ontological patterns offers information about resource types and their relationships that is independent of any specific domain. This helps agents contextualize the domain and its data. The example demonstrates the potential for improving the quality of metadata schemas, which now have mechanisms to support communication with different agents. Using the semantic model, SPARQL queries can serve different human agents, from managers (e.g., catalog administrators) to data publishers and consumers.

This work, which focuses on improving relationships, is part of a research project aimed at enhancing the semantic expressiveness of DCAT by employing an ontological foundation and multi-level principles [35, 36]. This enhancement is essential for semantic interoperability, as shown by other studies in the field. In [37], for example, the authors incorporate key concepts from DCAT into the structure of the Elementary Multiperspective Materials Ontology (EMMO), a domain-specific ontology, to advance semantics for data sharing and use from the perspective of industry commons. In [38], the authors introduce the Data Catalog, Provenance, and Access Control (DCPAC) ontology. This ontology has DCAT and PROV-O at its core and combines other standardized ontologies and vocabularies to add a semantic layer to data lakes.

In the specific case of data catalogs that curate resource descriptors hosted on other platforms, the entities in the metadata schema classified as relators and events play a crucial role. They enable the management of changes that occur in descriptors over time. More than just a record as provided in *dcat:CatalogRecord*, which acts as a log for resources in the catalog, they establish a context for the changes, providing a provenance for the descriptors. In the future, this work will be expanded to address other types of relations according to the typology offered by UFO, further improving the expressiveness of the DCAT.

Acknowledgements

This work has been partially supported with research grants from RNP, CAPES (Process number 88887.613048/2021-00), and FINEP/DCT/FAPEB (nº 2904/20-01.20.0272.00).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] J. Kropshofer, J. Schrott, W. Wöß, L. Ehrlinger. A Survey on the Functionalities of Data Catalog Tools. IEEE (2025). doi: 10.1109/ACCESS.2025.3568542.

[2] H. Sheridan, A. J. Dellureficio, M. A. Ratajeski, S. Mannheimer, T. R. Wheeler. Data Curation through Catalogs: A Repository-Independent Model for Data Discovery. JeSLIB (2021). doi: 10.7191/jeslib.2021.1203.

[3] S. C. Guptill. Metadata and data catalogues. Geographical information systems. (1999) .Vol. 2:677−92. URL: https://www.geos.ed.ac.uk/~gisteac/gis_book_abridged/files/ch49.pdf

[4] D. Wells. Introduction to Data Catalogs. 2019. URL: https://www.ecintegrators.com.au/wp-content/uploads/2022/08/dave-wells-intro-to-data-catalogs-alation-2.pdf

[5] M. P. Satija, M. Bagchi, D. Martinez-Avila. Metadata management and application. LIBRARY HERALD. (2020). doi: 10.5958/0976-2469.2020.00030.2.

[6] R. Albertoni, D. Browning, S. Cox, A. N. Beltran, A. Perego, P. Winstanley. The W3C Data Catalog Vocabulary, Version 2: Rationale, Design Principles, and Uptake. Data Intelligence. (2023). doi: 10.1162/dint_a_00241.

[7] O. Rouchon, E. Kraaikamp, E. Gonzalez, A. S. F. Kjeldgaard, N. P. Tenderup, J. Davidson, et al. D6.2 - Core metadata schema for legal interoperability. Zenodo (2024). doi: 10.5281/zenodo.11104269

[8] B. Van Nuffelen. Data Catalog Application Profile (DCAT-AP) - Version 3.0 [Internet]. SEMICEU (2024). URL: https://semiceu.github.io/DCAT-AP/releases/3.0.0/.

[9] A. Perego, B. van Nuffelen, GeoDCAT-AP - Version 2.0.0: A geospatial extension for the DCAT application profile for data portals in Europe, SEMIC Recommendation, European Commission, 2020. URL: https://semiceu.github.io/GeoDCAT-AP/releases/2.0.0/.

[10] L. O. B. S. Santos, K. Burger, R. Kaliyaperumal, M. D. Wilkinson. FAIR Data Point: A FAIR-Oriented Approach for Metadata Publication. Data Intelligence (2023); p. 163–183. doi: https://doi.org/10.1162/dint_a_00160.
[11] N. Guarino, T. P. Sales, G. Guizzardi. Reification and Truthmaking Patterns. Conceptual Modeling (2018). Cham: Springer International Publishing. pp. 151-65. doi: 10.1007/978-3-030-00847-5_13.
[12] F. Buschmann, K. Henney. Pattern-Oriented Software Architecture (2008). URL: http://www.dre.vanderbilt.edu/~schmidt/POSA-tutorial.pdf
[13] R. A. Falbo, G. Guizzardi, A. Gangemi, V. Presutti. Ontology patterns: clarifying concepts and terminology. In: Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns (2013) - Volume 1188. Aachen, p. 14–26.
[14] G. Amaral, G. Guizzardi. On the Application of Ontological Patterns for Conceptual Modeling in Multidimensional Models. Advances in Databases and Information Systems (2019). Cham: Springer International Publishing. doi: 10.1007/978-3-030-28730-6_14.
[15] N. Q. Oliveira, V. Borges, M. L. M. Campos. Analysis of Observational Variables from an Ontological Patterns Perspective. Proceedings of the 15th Seminar on Ontology Research in Brazil (ONTOBRAS) and 6th Doctoral and Masters Consortium on Ontologies (WTDO) (2022); URL: https://ceur-ws.org/Vol-3346/Paper5.pdf.
[16] E. Romanenko, D. Calvanese, G. Guizzardi. A Pattern-Based Approach for Explaining Ontology-Driven Conceptual Models. Intelligent Information Systems (2025). Springer Nature Switzerland, p. 137–44. (Lecture Notes in Business Information Processing; vol. 557). doi: 10.1007/978-3-031-94590-8_17
[17] N. Guarino, G. Guizzardi. "We Need to Discuss the Relationship": Revisiting Relationships as Modeling Constructs. Advanced Information Systems Engineering (2015). Springer International Publishing, p. 279–94. (Lecture Notes in Computer Science; vol. 9097). doi: 10.1007/978-3-319-19069-3_18.
[18] C. M. Fonseca, D. Porello, G. Guizzardi, J. P. A. Almeida, N. Guarino. Relations in Ontology-Driven Conceptual Modeling. Conceptual Modeling (2019). Springer International Publishing, p. 28–42. (Lecture Notes in Computer Science; vol. 11788). URL: http://link.springer.com/10.1007/978-3-030-33223-5_4.
[19] V. A. Carvalho, J. P. A. Almeida, G. Guizzardi. Using a Well-Founded Multi-level Theory to Support the Analysis and Representation of the Powertype Pattern in Conceptual Modeling. Advanced Information Systems Engineering (2016). p. 309–24. (Lecture Notes in Computer Science; vol. 9694). URL: https://link.springer.com/10.1007/978-3-319-39696-5_19.
[20] V. A. Carvalho, J. P. A. Almeida. Toward a well-founded theory for multi-level conceptual modeling. Softw Syst Model (2018) Feb;17(1):205–31. doi: 10.1007/s10270-016-0538-9.
[21] R. Albertoni, D. Browning, S. J. D. Cox, A. G. Beltran, A. Perego, P. Winstanley. Data Catalog Vocabulary (DCAT) - Version 3 [Internet]. W3C (2024). URL: https://www.w3.org/TR/vocab-dcat-3/.
[22] Unified Modeling Language, v2.5.1 [Internet]. Object Management Group (OMG); 2017. URL: https://www.omg.org/spec/UML/2.5.1/PDF.
[23] G. Guizzardi, G. Wagner. What's in a Relationship: An Ontological Analysis. Conceptual Modeling - ER (2008). p. 83–97. URL: http://link.springer.com/10.1007/978-3-540-87877-3_8.
[24] G. Guizzardi, A. B. Benevides, C. M. Fonseca, D. Porello, J. P. A. Almeida, T. P. Sales. UFO: Unified Foundational Ontology. Borgo S, Galton A, Kutz O, editors. AO (2022) Mar 15;17(1):167–210. doi: 10.3233/AO-210256.
[25] A. Albuquerque, G. Guizzardi. An ontological foundation for conceptual modeling datatypes based on semantic reference spaces. IEEE (2013). pp. 1–12. URL: http://ieeexplore.ieee.org/document/6577693/
[26] G. Guizzardi, N. Guarino N. Explanation, semantics, and ontology. Data & Knowledge Engineering (2024). doi: 10.1016/j.datak.2024.102325.

[27] D. Costal, C. Gómez, G. Guizzardi. Formal Semantics and Ontological Analysis for Understanding Subsetting, Specialization and Redefinition of Associations in UML. Conceptual Modeling – ER (2011). p. 189–203. doi: 10.1007/978-3-642-24606-7_15

[28] G. Guizzardi. Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling. Conceptual Modeling (2014). p. 13–27. (Lecture Notes in Computer Science; vol. 8824). URL: http://link.springer.com/10.1007/978-3-319-12206-9_2

[29] R. A. Falbo. SABiO: Systematic approach for building ontologies. ONTO.COM/ODISE (2014). URL: https://ceur-ws.org/Vol-1301/ontocomodise2014_2.pdf.

[30] P. P. F. Barcelos, T. P. Sales, E. Romanenko, J. P. A. Almeida, G. Engelberg, D. Klein, G. Guizzardi. Inferring Ontological Categories of OWL Classes Using Foundational Rules. FOIS 2023. doi:10.3233/FAIA231122.

[31] G. L. Guidoni, J. P. A. Almeida, G. Guizzardi. Forward Engineering Relational Schemas and High-Level Data Access from Conceptual Models. Conceptual Modeling (2021). p. 133–48. (Lecture Notes in Computer Science; vol. 13011). URL: https://link.springer.com/10.1007/978-3-030-89022-3_12

[32] R. Confalonieri, G. Guizzardi. On the multiple roles of ontologies in explanations for neuro-symbolic AI. Mileo A, editor. Neurosymbolic Artificial Intelligence. (2025) Mar;1:NAI-240754. doi: 10.3233/NAI-240754.

[33] C. M. Fonseca, G. Guizzardi, J. P. A. Almeida, T. P. Sales, D. Porello. Incorporating types of types in ontology-driven conceptual modeling. Conceptual Modeling (2022). doi:10.1007/978-3-031-17995-2_2.

[34] S. Auer. Semantic Integration and Interoperability. Designing Data Spaces (2022). Cham: Springer International Publishing; p. 195–210. URL: https://link.springer.com/10.1007/978-3-030-93975-5_12

[35] V. Borges, E. M. Prata, M. L. M. Campos. Ontology-Driven Multi-Level Conceptual Modeling for Dataset and Distributions Descriptions. Frontiers in Artificial Intelligence and Applications (2023), doi: 10.3233/FAIA231132.

[36] V. Borges, N. Q. Oliveira, M. L. M. Campos. A Multi-level Ontology-based Approach for Descriptors of Catalogued Resources. Proceedings of the 15th Seminar on Ontology Research in Brazil (ONTOBRAS) and 6th Doctoral and Masters Consortium on Ontologies (WTDO), 2022; URL: https://ceur-ws.org/Vol-3346/Paper4.pdf.

[37] E. Ghedini, A. Hashibon, J. Friis. Semantic data exchange ontology. (2022). Zenodo. URL: https://doi.org/10.5281/zenodo.7784934.

[38] H. Dibowski, S. Schmid, Y. Svetashova, C. Henson, T. Tran. Using Semantic Technologies to Manage a Data Lake: Data Catalog, Provenance and Access Control. SSWS@ ISWC (2020). URL:http://www.ssws-ws.org/SSWS2020/SSWS2020_paper5.pdf