Formal Ontology in Information Systems
N. Aussenac-Gilles et al. (Eds.)

doi:10.3233/FAIA231132

Ontology-Driven Multi-Level Conceptual Modeling for Dataset and Distributions Descriptions

Vânia BORGES[a,1] , Eduardo M. PRATA[a] and Maria Luiza M. CAMPOS[a]
[a] *Universidade Federal do Rio de Janeiro*
ORCiD ID: Vânia Borges https://orcid.org/0000-0002-6717-1168
ORCiD ID: Eduardo M. Prata https://orcid.org/0000-0001-7572-8336
ORCiD ID: Maria Luiza M. Campos https://orcid.org/0000-0002-7930-612X

Abstract. This paper presents an improvement proposal for an ontology-driven multi-level conceptual model for the data catalogue domain. Data catalogues gather metadata that describe resources in different and heterogeneous digital platforms (repositories). They are supported by Information Systems (IS) that use these descriptors to provide visibility and support resources exploration and analysis. Domain ontologies are essential to promote quality ISs, as they are developed to reflect the intended reality. The proposed conceptual model is well-founded on the Unified Foundational Ontology and the Multi-Level Theory, based on the widely used DCAT vocabulary, a standardized metadata schema for describing datasets and data services. The resulting model addresses ambiguities and contemplates high-level types contributing to the conformance of domain concepts and relationships. In addition, they provide knowledge about the different types of resource descriptors and relationships contained in a specific catalogue, favoring its management. The paper enhances the previous model by extending it to handle descriptors representing a dataset according to the data equivalence across multiple distributions. We also demonstrate the model by describing a dataset with no data equivalence in its distributions, taken from a real-world scenario, thus providing a structured representation to manage metadata sets in the data catalogue domain.

Keywords. Ontology-driven conceptual model, multi-level, data catalogue, metadata management.

1. Introduction

Data repositories are used by research institutions and government agencies to make data available on the Web. Implemented on heterogeneous digital platforms, and autonomously developed [1], these repositories are responsible for storing, curating, and accessing these data. However, bringing together data distributed in these different information silos to answer relevant questions from various domains promptly requires significant effort [1, 2].

To assist this task, catalogues have been used with repositories to increase visibility and access to catalogued resources [3]. A data catalogue is “a collection of metadata, combined with data management and search tools, that helps analysts and other data

[1] Corresponding Author: Vânia Borges, vjborges30@ufrj.br.

users to find the data they need, serves as an inventory of available data, and provides information to evaluate the fitness of data for intended uses.” [4]. In this configuration, repositories are responsible for the resource storage and curation, while catalogues are responsible for the metadata curation that describes those resources.

The dissemination of FAIR principles, aiming at **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable data, has emphasized the importance of semantically rich, readable, and agent-actionable (by human and machine) metadata [5]. They provide visibility to catalogued resources by describing them and presenting relevant information for their use. This machine-actionability, evidenced by the FAIR principles, is achieved with proper metadata treatment and implementation of digital infrastructures oriented by these principles [6].

Under the GO FAIR initiative, metadata treatment is observed with metadata schemas and standardized values to generate metadata records in triples, using the Resource Description Framework (RDF) model. In particular, for catalogues, we highlight the Data Catalog Vocabulary (DCAT), a metadata schema recommended by W3C for describing catalogued resources providing semantics and context [7]. It was developed in Web Ontology Language (OWL) to facilitate interoperability between catalogues on the Web [7].

The implementation of digital infrastructures may benefit from ontology-based and software engineering strategies. Thus, Information Systems (IS) that support data catalogues should be developed from well-founded domain ontologies, handling the intended reality to be represented, i.e., the domain of catalogued resources descriptors, providing elements (constructs) for managing these descriptors.

In order to support catalogue machine-actionability, an ontology-driven multi-level conceptual model for the data catalogue domain has been developed based on DCAT [8]. This model was developed using ontology analysis and a well-known foundation ontology, assigning metaproperties to existing concepts and relationships. Furthermore, this analysis allowed for ambiguities and higher-order types to be identified and dealt with. The latter contributes to (i) native conformance of domain concepts and relationships, and (ii) knowledge about the different resource descriptors and relationships in a specific catalogue, favoring their management. Despite improving understanding and addressing ambiguities, the model did not deal with implicit aspects of DCAT concepts that are handled independently by the ISs which support catalogues and repositories.

As a contribution, this work extends the previous model, providing richer descriptors for datasets, according to data equivalence across their multiple distributions. For DCAT, the implemented ISs are responsible for this distinction, respecting the community they serve [7]. By representing these descriptors, we aim at a core ontology for the catalogues domain with improved expressiveness for dataset description. This ontology leads to the standardization of distinct types of dataset organizations and compositions, improving the ISs quality, and supporting interoperability approaches among catalogues.

This paper is organized as follows: Section 2 discusses the relation between metadata and the data catalogue domain; Section 3 highlights the benefits of ontology-driven conceptual modeling; Section 4 presents the original model implemented from DCAT; Section 5 extends the model to describe datasets according to the data equivalence in their distributions; Section 6 explores the model to describe a dataset with non-equivalent distributions, addressing its benefits; and Section 7 concludes this paper and presents work in progress.

2. Metadata and the Data Catalogue Domain

According to Sheridan et al. [3], a data catalogue “is a curated collection of metadata records that describe and point to data products of interest.” To achieve these purposes, they should be supported by an IS-specific class type that meets FAIR principles by promoting documentation [9] and metadata management [10]. This paper focuses on catalogues storing metadata of research datasets, allowing for their discovery, access, and understanding.

Metadata records are not typically considered as primary resources, but rather “surrogate, excerpt, abstract, or description giving some attributes of another resource.” [11]. They are created, checked, and updated to guarantee the accuracy and understanding of their metadata by agents (both humans and machines) who aim to discover, cite, and reuse research data [3]. To achieve this goal, metadata records consist of elements with assigned values describing the resource of interest. In order to add meaning and contribute to understanding, these metadata elements are organized in metadata models created from one or several metadata schemas. A schema “(also called metadata model) refers to a high-level, annotation model used for capturing descriptive information about varied facets of an information resource, facilitating the broader objective of achieving a unified understanding of the semantics of the data.” [12]. It can be generic or domain-specific, modeling data structures of specialized domains. In addition, a schema presents a conceptualization that is usually formalized in a specification and follows patterns to guide this process [13].

Despite the concern with its creation, Connolly [11] raises questions like “Whence comes the list of attributes?” and “What's the expressive capability, structure, and meaning of attribute values?”. These and other issues highlight the importance of knowing the nature of entities, attributes, and value spaces in the data catalogue domain schemas. If we consider that good metadata records should be treated as digital objects [14], we can adopt ontology-based conceptual models to represent them. In addition, it is possible to treat elements relevant to catalogue management using multi-level conceptual modeling [8]. These models, functioning as domain ontologies, are essential for ISs.

3. Ontology-driven Conceptual Modeling

Conceptual modeling represents physical and social world aspects, aiming at their understanding and communication among humans [15]. According to Sales [16], the ontology-driven conceptual modeling discipline is similar to conceptual modeling. However, it formally captures domain knowledge driven by an ontological foundation. This foundation describes the nature of things that exist, their properties and relations, with the goal of achieving greater expressivity.

The adoption of the consistent basic categories defined by foundational ontology enables [1]: (i) increased expressiveness and formality, providing semantics that better represents the real world, employing well-founded types and constraints; (ii) simplicity for the understanding of those involved, reducing ambiguities; (iii) use of formal theories to assist the identification of the relationship between the concepts involved and how they behave; (iv) support for semantic interoperability at the conceptual level, establishing "contracts" that capture the conceptualizations and representations in models or other instruments utilized to harmonize the knowledge. These aspects collaborate with the appropriate association between conceptualization elements of different systems, promoting interoperability.

In this context, models are employed to negotiate meaning and semantic interoperability between communities, organizations, and authorities [17]. Once a robust conceptual model is defined, different information models can be generated using distinct logical languages and meeting different non-functional implementation requirements [17].

It is worth mentioning that ISs quality directly depends on how accurate the models they adopt are in treating the reality they intend to represent [1]. Thus, the models should unambiguously represent all relevant aspects of the associated conceptualization and restrict the possible states of the specific IS to those representing the intended state of affairs.

In this work, we adopt the Unified Foundational Ontology (UFO) and the Multi-Level Theory (MLT) as our ontological foundation. UFO is a top ontology built upon several theories and its concepts can be associated with the MLT elements. When combined, UFO-MLT establishes an approach for developing conceptual models that represent types and types of types, adhering to foundational ontology rules [18]. These ontologies are presented next. For further studies on the matter, we recommend [19, 20] for UFO and [18, 21, 22] for MLT.

3.1. *Unified Foundational Ontology (UFO)*

According to Guizzardi et al. [20], the UFO is a top ontology for conceptual modeling, developed considering theories from formal ontology in philosophy, cognitive science, linguistics, and philosophical logic. A set of micro-theories addresses the fundamental conceptual modeling notions such as the theory of types and taxonomic structures; part-whole relations, particularized intrinsic properties, attributes, and attribute value spaces; and others. UFO aims to provide foundations for domain analysis in conceptual modeling, as well as for designing concrete models and modeling grammars.

As aforementioned, good metadata records should be treated as digital objects [14]. In this context, they can be categorized as endurants in UFO, i.e., entities that can suffer changes over time without losing their identity. Figure 1 presents the taxonomy established for categorizing Endurant Types in UFO.

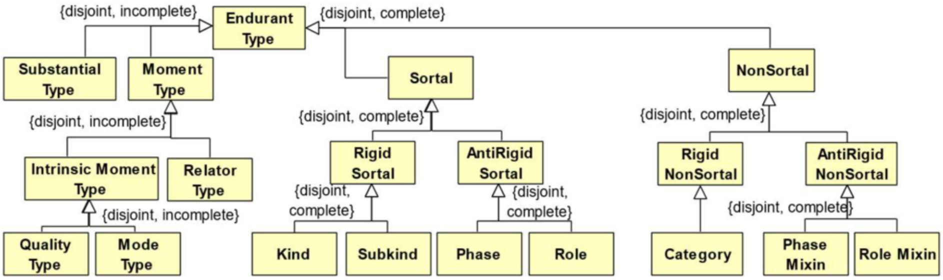


Figure 1. Taxonomy for Endurant Type in UFO, adapted from [20].

According to [20, 21], Endurants Types are invariant structures represented by object-like entities. They are partitioned according to the ontological nature of their instances into Substantials and Moments. Substantials are existentially independent individuals, such as a person or an organization. Moments are specific aspects of individuals, existentially dependent on them or other individuals. Moments are categorized into Intrinsic Moment or Relator. The former is partitioned into Quality and Mode.

Quality refers to a particular aspect of an individual. It can be compared with another individual, considering the assumed value in a certain quality space. Mode is an aspect that can have its own qualities. A Relator is an aggregation of qua individuals. It is existentially dependent on multiple individuals, namely, the bearers of its constituting qua individuals.

Endurant Types are also classified according to orthogonal characteristics of how they apply to their instances [20, 21]. These characteristics refer to sortality and rigidity. Sortals are Endurant Types that provide a uniform principle of identity for their instances, i.e., a principle that captures which properties two instances of a type must have in common in order for them to be the same. In particular, the principle of identity tells which changes an Endurant can undergo while maintaining its identity. Sortal may provide the identity principle directly to its instances or inherit this principle from another Sortal Type. However, all Sortal Types share the same identity principle and inherit it from a unique Sortal. Non-sortal Types classify endurants with distinct identity principles and are known as dispersive types. They aggregate properties that are common to different sortals.

Sortal and Non-sortals are distinguished according to their rigidity. Rigid Types classify their instances while they exist. Anti-rigid Types classify their instances contingently, with their instances moving in and out of their extension without ceasing to exist, i.e., maintaining their identity. Phase Mixin and Role Mixin have similar foundations, as they apply to types whose instances are associated with different identity principles.

3.2. Multi-Level Theory (MLT)

In certain domains, the traditional two-level classification system (types/classes and instances/objects) employed in conceptual models is not sufficient. These domains require the representation of types of types (or categories of categories) to model their conceptualizations accurately. To address this issue, multi-level conceptual modeling is adopted [18].

The MLT was adapted to UFO and is used for conceptual modeling of multi-level types [22]. In a recent study, Fonseca et al. [21] presented an ontological analysis of the concept of Type, categorizing types as endurants. From this analysis, independent axiomatizations were developed from UFO and MLT, resulting in a rich set of rules that prevent common errors in multi-level models and incorrect combinations of metaproperties. This study aims to incorporate MLT as a micro-theory of UFO to address higher-order types.

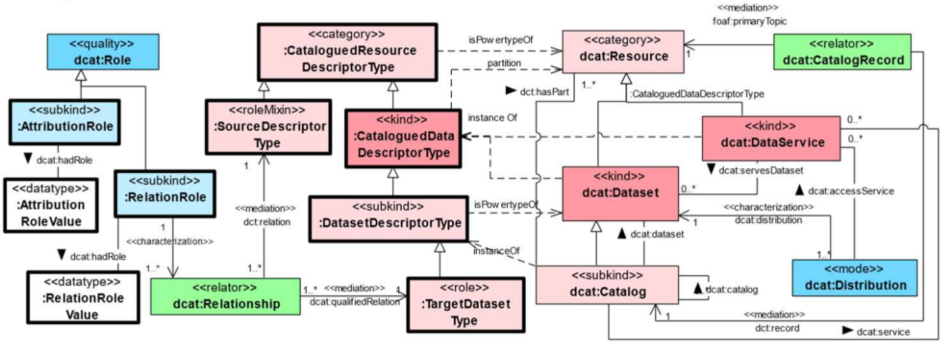
This paper employs the approach for relations handling between levels defined by MLT [21, 22]. In multi-level models, structural relations characterize how types are related in terms of their intensions, i.e., the properties they possess that apply to their instances. These relations can be intra-level or cross-level. The specialization relation is an intra-level relation between a more specialized type and a type of the same order. It can be further divided into specialization and proper specialization. A type t1 specializes a type t2 if every possible instance of the former is necessarily an instance of the latter. The proper-specialization relation characterizes the specialization between two distinct types, i.e., not all instances of t2 are specialized on t1.

MLT also establishes subordination as an intra-level relation. This relation between higher-order types of equal order is reflected in specializations between types of lower order, i.e., a subordination between types that are instances of related higher-order types. This relation is essential for complex domains, where subordination between defined types must be represented to promote an understanding of the classification criteria involved [22]. Thus, if a type t1 isSubordinateTo a type t2, then the intension of every instance of t1 adds some classification criterion to the intention of some instance of t2, i.e., every instance of t1 proper specializes some instance of t2 [23].

The cross-level structural relations occur between types of adjacent orders. These relations support the analysis of the different powertype notions in the literature [22, 24]. The relation isPowertypeOf follows Cardelli's notion of Powertype [25]. Thus, if a type t1 isPowertypeOf a type t2, then every specialization of t2 is an instance of t1, including t2 itself. The categorizes relation, in turn, follows the notion of Odell [26]. Thus, a type t1 categorizes a base type t2 if every instance of the former is a proper specialization of the latter. Therefore, instances of t1 are those types whose intensions include not only the base type intension, but also additional constraints defined by the categorizing type. In this case, different specializations of t2 may exist based on criteria distinct from the one established by the type t1. A variation of categorization is the partition relation. A type t1 partitions a type t2 if t1 categorizes t2 and each instance of t2 is an instance of exactly one instance of t1.

4. Ontology-driven Multi-level Conceptual Model Using DCAT

In [8], we performed an ontological analysis of the elements that compose DCAT. This vocabulary is a metadata schema, i.e., a logical model created to promote interoperability between web catalogues, emphasizing the description of datasets and data services. From this analysis, we obtained an ontology-driven multi-level conceptual model, i.e., an initial domain ontology for catalogues. In this process, entities classified as higher-order types were identified, complementing the ontological analysis with MLT concepts. This not only accommodates higher-order types but also characterizes the various powertypes that define the base types of DCAT. These powertypes play a crucial role in catalogue management.



<<quality>> dcat:Role
<<subkind>> :AttributionRole
dcat:hadRole
<<datatype>> :Attribution RoleValue
<<subkind>> :RelationRole
<<characterization>>
dcat:hadRole
<<datatype>> :RelationRole Value
<<relator>> dcat:Relationship
<<category>> :CataloguedResource DescriptorType
<<roleMixin>> :SourceDescriptor Type
<<mediation>> dct:relation
<<kind>> :CataloguedData DescriptorType
<<subkind>> :DatasetDescriptorType
<<mediation>> dcat:qualifiedRelation
<<role>> :TargetDataset Type
isPowertypeOf
partition
dct:hasPart
instance Of
isPowertypeOf
instanceOf
<<mediation>> foaf:primaryTopic
<<category>> dcat:Resource
<<relator>> dcat:CatalogRecord
:CataloguedDataDescriptorType
<<kind>> dcat:DataService
dcat:servesDataset
dcat:accessService
<<kind>> dcat:Dataset
<<characterization>> dcat:distribution
dcat:dataset
<<subkind>> dcat:Catalog
dcat:catalog
<<mode>> dcat:Distribution
<<mediation>> dct:record
dcat:service
1
1..*
1..*
1
1..*
1..*
1
1
1..*
0..*
0..*
0..*
1
1..*
1

Figure 2. DCAT-UFO-MLT Model, adapted from [8].

After the analysis, we obtained the model presented in Figure 2. This model and the following ones presented in this article were developed using the Visual Paradigm[2] tool, version 17.0, with the OntoUML [3] plugin, version 0.5.3, and available in the OntoUML/UFO Catalog[4]. According to [18], OntoUML is a conceptual modeling language whose primitives reflect the ontological micro-theories that compose UFO. Therefore, the plugin supports modeling by installing the stereotypes and adding features such as intelligent diagram coloring. The color coding for classes represents the nature of their possible instances where object types are represented by classes in red, relator types by classes in green, and mode or qualities types by classes in blue [18]. Furthermore, in the models, DCAT entities use the prefix "dcat". The prefix "dct" indicates Dublin Core terms. Finally, dashed arrows are used to define dependency relationships between types and the relations labels refer to the applied predicates names.

In Figure 2, entities are associated with UFO stereotypes defining metaproperties. To the right, with simple borders, are the DCAT first-order types, i.e., types whose instances cannot have instances. The dcat:Resource is a *category*, i.e., a non-sortal that gathers common properties from different catalogued resources descriptors (Sortals); the entities dcat:Dataset and dcat:Dataservice are *kinds*, establishing the principle of identity for their instances. The entity dcat:Distribution describes the different serializations of a dataset for access or transfer. Its instances are dependent on exactly one dcat:Dataset instance, making it possible to access and even understand the dataset. Furthermore, if a dcat:Dataset instance is removed, all its dcat:Distributions instances are also removed. Thus, because it is part of the dataset descriptor characterization, it is categorized as a *mode*. The dcat:Catalog a *subkind* of dcat:Dataset, inheriting the identity principle. Finally, dcat:CatalogRecord is a *relator*, i.e., an existentially dependent entity, emerging from the relationship of dcat:Resource to dcat:Catalog, registering relevant information from a catalogued resource descriptor in a catalogue.

On the left side of the model, we have the second-order types, i.e., types whose instances are first-order types. The types with simple borders are those from DCAT, whose analysis classified them as higher-order types. Those with large borders refer to the new types established to handle DCAT ambiguities and powertypes. These types were also associated with UFO metaproperties. Furthermore, dashed arrows indicate the relations between higher-order types and their instances.

The created second-order types are shown next. :CataloguedResourceDescriptorType is categorized as *category* and *powertype* of dcat:Resource. Thus, dcat:Resource and all its specializations are instances of this new type. :CataloguedDataDescriptorType is a specialization of the first type and categorized as a *kind* that *partitions* dcat:Resource. Thus, dcat:Dataset and dcat:DataService are descriptor types handling catalogued data, instances of :CataloguedDataDescriptorType. :DatasetDescriptorType is a *subkind* of :CataloguedDataDescriptorType and *powertype* of dcat:Dataset. Through it, we identify all possible types of dataset descriptors in the model, as they are its instances. :TargetDatasetType is a *role* of :DatasetDescriptorType establishing different types of dataset descriptors related to dcat:Relationship. This class, together with :SourceDescriptorType, a *rolemixim*, makes explicit the DCAT constraints to new relationships in specific models. Thus, for specific catalogue models, instantiations of

[2] https://www.visual-paradigm.com/

[3] https://github.com/OntoUML/ontouml-vp-plugin

[4] https://github.com/OntoUML/ontouml-models

dcat:Relationship can occur between any resource descriptor type and dataset descriptor types.

In addition, we have dcat:Relationship, categorized as a *relator* mediating a relationship between any catalogued resource descriptor type (:SourceDescriptorType) and a dataset descriptor type (:TargetDatasetType). To handle the relationship functionality, we have dcat:Role. It was categorized as *quality*. Handling ambiguity, dcat:Role was specialized in :AttributionRole and :RelationRole, both *subkinds*. The former assigns the agent role related to a dataset descriptor type, and the latter defines the relation role between resource descriptor types and dataset descriptor types, characterizing dcat:Relationship. Each *subkind* has its own datatype (value space) in a specific domain model, contributing to standardized values management and interoperability.

Second-order types contribute to the definition of domain-relevant concepts employed by IS for managing types in the catalogue. In addition, they allow the understanding of the catalogue structure, facilitating access. The resulting ontology-based multi-level conceptual model is capable of [8]: (i) promoting context for attributes used by different descriptors; (ii) improving the understanding of the resources available in the catalogue, contributing with interoperability; (iii) establishing new specific relations according to communities' needs; and (iv) providing means for catalogue management itself, establishing rules to be attended by users when publishing their metadata, ensuring conformity.

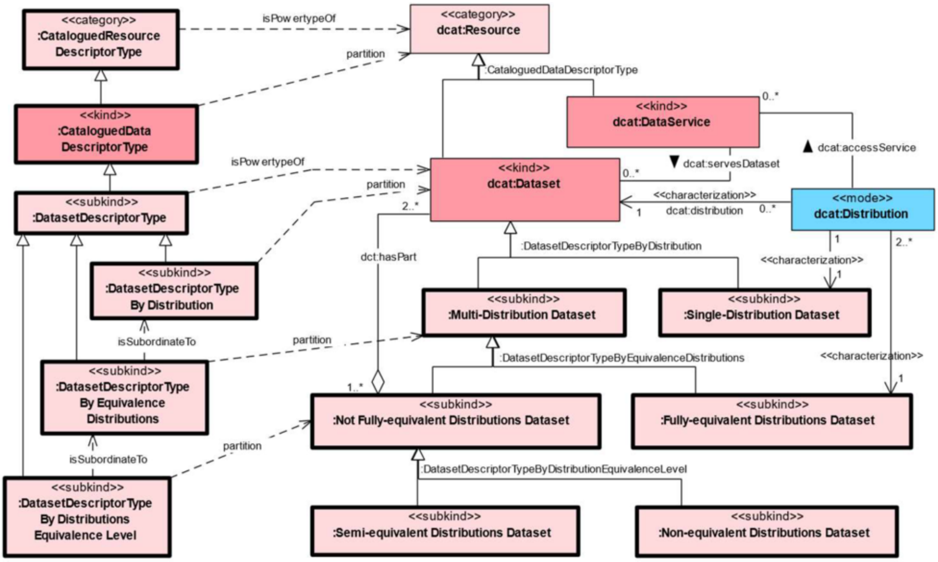
Despite providing a semantically improved model, aspects related to dataset descriptors and data equivalence across datasets multiple distributions were not considered. However, this information is deemed relevant for the standardization among ISs, favoring interoperability. This approach is presented in the next section.

5. Extending the Model to Describe Datasets Data Equivalence

According to DCAT [7], datasets with multiple distributions may present different situations regarding the data they store. First, we have dataset distributions that are fully equivalent in terms of data. In this case, there is no loss of information among the data files. An example is a dataset whose distributions are made up of the same data available in different formats, i.e., we would have different serializations of an RDF graph using RDF/XML, RDF/Turtle, and RDF/JSON-LD. A second case would be distributions with some data equivalence but different levels of fidelity, for example, a dataset whose data files present different aggregation levels over the same data set. Thus, there are differences between the distributions, but still, they refer to the same data. Finally, a third case would be distributions with no data equivalence, i.e., each one contains relevant parts referring to the dataset. An example would be a dataset of COVID-19 cases. In this dataset, we have a data file with patients' data, another one with data related to patients' exams, and yet another with the performed attendance outcomes. They are all part of the same dataset but store different data.

For DCAT [7], handling these different situations is up to the applications. Thus, the catalogue provider establishes the way of describing based on user expectations and practices within the relevant community. However, this potential diversity of descriptions hampers interoperability and requires additional efforts, even human intervention, for search and analysis engines.

This section presents a proposed approach for treating dataset descriptors based on their distribution, building upon the model presented in [8]. Thus, we differentiate dataset descriptors describing single files from those presenting multiple distributions. Furthermore, the applied approach aims to identify the degree of equivalence between the data contained in multiple distributions. By identifying these differences, we establish contracts which must be respected by all catalogue users, regardless of their community.



<<category>>
:CataloguedResource DescriptorType
isPowertypeOf
<<category>>
dcat:Resource
partition
:CataloguedDataDescriptorType
<<kind>>
:CataloguedData DescriptorType
<<kind>>
dcat:DataService
0..*
▲ dcat:accessService
isPowertypeOf
▼ dcat:servesDataset
<<kind>>
dcat:Dataset
0..*
partition
<<characterization>>
dcat:distribution
1
0..*
<<mode>>
dcat:Distribution
<<subkind>>
:DatasetDescriptorType
2..*
:DatasetDescriptorTypeByDistribution
1
2..*
dct:hasPart
<<characterization>>
1
<<subkind>>
:DatasetDescriptorType By Distribution
<<subkind>>
:Multi-Distribution Dataset
<<subkind>>
:Single-Distribution Dataset
isSubordinateTo
partition
:DatasetDescriptorTypeByEquivalenceDistributions
<<characterization>>
1
<<subkind>>
:DatasetDescriptorType By Equivalence Distributions
1..*
<<subkind>>
:Not Fully-equivalent Distributions Dataset
<<subkind>>
:Fully-equivalent Distributions Dataset
isSubordinateTo
partition
:DatasetDescriptorTypeByDistributionEquivalenceLevel
<<subkind>>
:DatasetDescriptorType By Distributions Equivalence Level
<<subkind>>
:Semi-equivalent Distributions Dataset
<<subkind>>
:Non-equivalent Distributions Dataset

Figure 3. Descriptor types for datasets according to data equivalence.

Figure 3 highlights new entities established from the ontological analysis of the dataset descriptor according to the data in its different distributions, as mentioned in the DCAT. In this model extract, new second-order types partition first-order types, including explicit rules concerning the dataset descriptor in terms of distribution. We see the new second-order types on the model left, with larger borders, and, on the right, specializing dcat:Dataset, the instances of these types. It should be noted that the data equivalence characterization only applies when describing datasets with multiple distributions.

In the model shown in figure 3, :DatasetDescriptorTypeByDistribution is a *subkind* of :DatasetDescriptorType. Furthermore, it *partitions* the dataset descriptor by establishing specializations categorized as *subkinds*. These new specializations classify the dcat:Dataset according to the composition of its distributions. Thus, it can be a :Multi-Distribution Dataset when it presents multiple distribution descriptors or :Single-Distribution Dataset when it presents only one. The latter allows establishing the 1:1 cardinality for the *characterization* relation with dcat:Distribution.

In order to describe a dataset with multiple distributions, it is necessary to show whether data equivalence exists between them. To do this, we first define the second-order type :DatasetDescriptorTypeByEquivalenceDistributions. It is also a *subkind* of the entity :DatasetDescriptorType and *isSubordinateTo* :DatasetDescriptorTypeBy Distribution. The subordination relation implies that this entity intention adds a classification criterion to the intention of the :DatasetDescriptorTypeByDistribution entity. Its instantiations proper specialize some instance of the entity it subordinates. In our model, it *partitions* :Multi-Distribution Dataset, defining two specializations, :NotFully-equivalentDistributionsDataset and :Fully-equivalent DistributionsDataset, both classified as *subkind*. The :Fully-equivalentDistributions Dataset classifies datasets descriptors whose distributions reflect the same data, i.e., the distributions being described are different serializations of the same data set. Thus, we establish a *characterization* relation with dcat:Distribution with cardinality 2..*, i.e., a dataset descriptor will have at least two distribution descriptors.

The second-order type :DatasetDescriptorTypeByDistributionsEquivalenceLevel is also a *subkind* of the :DatasetDescriptorType and *isSubordinateTo* :Dataset-DescriptorTypeByEquivalenceDistributions. This type *partitions* :NotFully-equivalent-DistributionsDataset, specializing it into two *subkinds,* according to the level of data equivalence between them. The first is :Semi-equivalentDistributionsDataset which describes datasets whose distributions have some equivalence among their data. The second type is :Non-equivalentDistributionsDataset which describes datasets whose distributions are related to the dataset theme but have different data.

In the model, the subordination relations between second-order types are responsible for the classification criteria which establish the specializations between the first-order types. Another relevant aspect is using "subkind" for the new classes. The UFO *subkind* rigidity provides a relevant understanding of the model. If a single-distribution dataset descriptor is instantiated, it cannot be transformed over time into a multi-distribution one. Each descriptor unambiguously describes a published dataset. Thus, if a repository establishes a new version for a dataset, it will have a different descriptor. This unambiguous identification is relevant for research reproducibility. Therefore, the descriptor will describe a particular dataset/version while it exists. It is worth mentioning that although the descriptor arises from the resource, it is not existentially dependent on it. Hence, as expected for FAIR data, the descriptor can be kept in the catalogue even if the dataset is no longer available.

According to the model, :NotFullyEquivalentDistributionsDataset has a compositional relationship consisting of two or more dataset descriptors. This organization is essential for the end users to know the specific content of each distribution (data file) and, with this knowledge, discover which one best suit their needs.

Distribution descriptors characterize dataset descriptors, providing information on technical specifications, usage, and data access. This proposal explicitly outlines constraints for these descriptors when describing single or with full data equivalence datasets.

Considering the composition of the types of dataset descriptors with non-equivalent distributions (:Semi-equivalentDistributionsDataset or :Non-EquivalentDistributions-Dataset), their description can be obtained in two ways: (i) simply, with only their base descriptor, and (ii) more completely, adding the information from each dataset descriptor that composes it. The component dataset descriptors aggregate relevant information. Similarly, the technical information of these descriptors types can be obtained: (i) from their (high-level) distribution descriptor, for example, by making explicit the information of a zipped file, and (ii) by combining that provided in the distribution descriptors of the dataset descriptors that compose it. Consequently, inference engines can extract a more extensive information set, thus favoring their localization, access, and understanding capabilities.

The new taxonomy establishes rules to describe datasets according to the equivalence of their distributions, expanding the knowledge about them. Therefore, it can be employed by search and analysis engines to improve their activities. By addressing these issues, we aim to increase the management capacity of the descriptors in the catalogues and, through standardization, the interoperability aspects. Understanding the organization of the data described in the different distributions associated with the dataset descriptor is an essential requirement for search and analysis mechanisms as well as for understanding the dataset itself.

6. Extended DCAT-UFO-MLT usage associated with a real scenario

To demonstrate the expressiveness obtained from the extended model, we describe a dataset of patients cases that tested positive for COVID-19. This dataset is published in the FAPESP repository [27], an existing repository developed using the DSpace digital platform [28]. DSpace offers mechanisms for storing, curating, and preserving resources associated with metadata that allow their discovery and access.

According to the repository organization, the datasets belong to the "COVID-19 DataSharing/BR" collection, which is part of the "FAPESP COVID-19 DataSharing/BR" community. The data made available by partner hospitals and institutions are stored as items (datasets) with associated data files (bundles/bitstream). We use as an example the data from Sírio-Libanes Hospital (SLH) "COVID Data-SLH". Data collected by this hospital is organized in specific data files, as presented earlier: a data file for patients, one for exams, and another for outcomes. They are in CSV format and have been published as a zipped file in the repository. The information about each data file is presented in plain text in the dataset description field. To support search mechanisms, the keywords "covid-19", "test results", "serology", "PCR", "coronavirus", and "pandemic" have been registered for the dataset. Besides the mentioned data files, the zipped file has an Excel spreadsheet with a data dictionary for accessing the data. This spreadsheet is not treated in this paper. Instead, it is part of future work regarding the data structure described in the distribution descriptors.

Through a quick analysis, we could classify the SLH dataset as a single-distribution dataset. However, examining its definition and content, we observe that, although it presents itself with a single distribution, this distribution is a compendium of data files, each one comprising relevant research data. Because they contain different data, all of them must have a specific description concerning their conceptual part and distribution. In this case, to provide FAIR metadata for this dataset, it is necessary, besides describing the distribution referring to the zipped file, to describe the data files it contains. This way, we increase the dataset visibility and understanding.

With this perspective, by adopting the extended model, we initially represent the data files of the zipped file. Each one is described by a :Single-DistributionDataset with its technical specification, such as size and format, described by a distribution descriptor. These descriptors are part of the SLH dataset descriptor, which is classified as a :Non-equivalentDistributionDataset, since it contains dataset descriptors with different data. It also has a distribution descriptor describing the zipped file technical information. The dataset descriptor classification is inferred from the compositional analysis and offers software agents a differentiated view of the dataset, facilitating its access and providing a better understanding of its constitution.

If we had described the SLH dataset as a single-distribution dataset just because of the zipped file, we would fail to present relevant metadata about the data files that compose it, which provide valuable knowledge about their contents. On the other hand, if we had described only the data files that comprise it, we would fail to provide relevant zipped file physical aspects to mechanisms that will access the dataset. Thus, aiming at FAIR metadata that contributes to location, access, reuse, and interoperability, it is necessary to enrich the metadata, thoroughly describing the dataset.

It should be noted that if the repository only stores CSV data files after the zipped file ingestion, the base dataset could be described without a distribution descriptor. However, it would comprise three single-dataset descriptors, each with its respective distribution descriptor.

Due to space limitations, figure 4 presents a simplified view, containing the zipped file and the data files referring to patient demographics and exams. At the top of the figure, the catalogue schema-level highlights the entities to describe datasets. At the bottom, the dataset information is stereotyped with the DSpace data model entity names, representing the repository instances stored in a relational database. Thus "COVID Data SLH" is treated as an <<Item>>, the SLH_Jun2021, a <<Bundle>> composed of the <<Bitstream>> SLH_Jun2021.zip, SLH_Exam_3.csv and SLH_Demographic_3.csv. In the middle, we present the catalogue instances, with the metadata records that describe the dataset hosted in the repository. The "isDescribedBy" arrows indicate the association between the repository resources and their respective descriptors in the catalogue. The dashed arrows indicate the entities' instantiations in the catalogue domain. Note that the access URLs for all distributions descriptors point to the page for the zipped file.

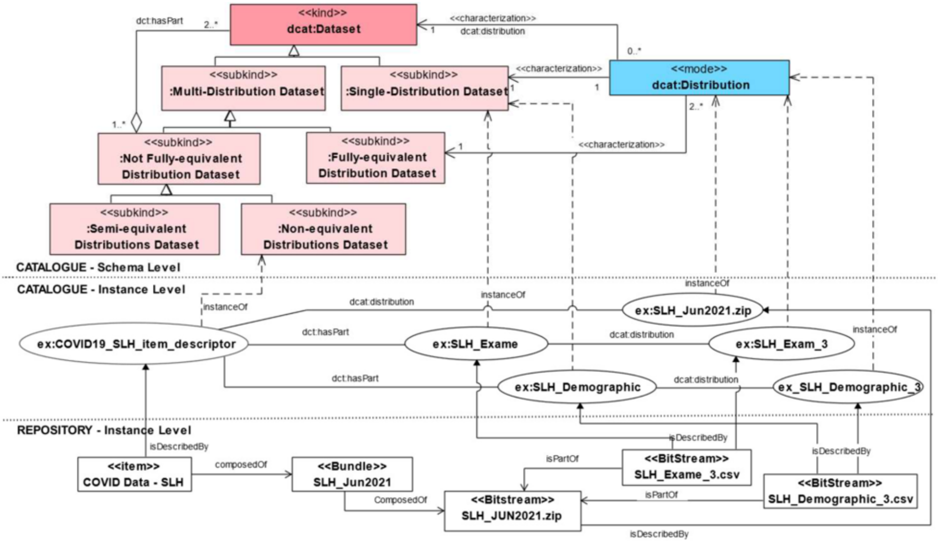


Figure 4. Data set representation without equivalence across data distributions

Using the model, the catalogue can describe the original dataset nature regarding the data across its distributions in a structured and standardized way, allowing agents to identify its different components. Furthermore, agents may gather the descriptors information, broadening the original dataset view. Moreover, the model supports the description of complex datasets like datasets similar to the example, where the exam and demographics data have more than one serialization. In this case, the original dataset would remain as :Non-equivalentDistributionsDataset, consisting of two :Fully-equivalentDistributionsDatasets. Each presents its different distributions. Hence, various combinations are possible, making explicit the information about the catalogued resources employed by IS.

7. Conclusion and Future Work

As aforementioned, using catalogues in conjunction with repositories increases visibility and, consequently, the discovery and reuse of data [3]. By addressing metadata, catalogues bring together information about catalogued resources from different and heterogeneous digital platforms. Good metadata records (instances in catalogue) should be treated as digital objects, having their own metadata [14]. Based on this view, we emphasize the need for well-grounded information structures to represent a diversity of objects organizations. The information structures such as ontologies and conceptual models are important to provide quality to IS [1]. They define a common terminology for the core concepts of a domain and establish a "contract" between the parties, promoting communication and semantic interoperability [1]. Those are relevant aspects in the catalogues and repositories domain.

In this paper, we extend the multi-level conceptual model based on DCAT started in [8] by examining the dataset descriptors categories concerning data equivalence across their different distributions. The new descriptors enriched the model by promoting: (i) qualification of existing concepts, expanding the model expressivity and semantics; (ii) increased knowledge about data distributions equivalence for information seekers; (iii) standardization of descriptors used by the ISs, improving their quality; and (iv) alignment to FAIR principles by providing contextualized, structured, and standardized metadata to support interoperability approaches among catalogues. The explicit distinction allows the catalogue to manage dataset descriptors according to their structural organization, avoiding semantic overload. In this way, the expressiveness and semantics of the model were increased, supporting a more comprehensive description of datasets published in repositories that serve different communities or present several forms of data access and processing.

In addition, we propose an approach for describing datasets with non-equivalent distributions, demonstrating the model with a dataset made available in a DSpace repository. The zipped file contains several data files and serves as an intermediate file. Each data file content is distinct, and to understand them, we need explicit dataset descriptors. These descriptors compose a main descriptor referring to the base dataset, identified as bearing non-equivalent distributions. Based on established rules, search and localization mechanisms may provide access to a broader range of information.

For future work, we are exploring: (i) the attributes and relationships representation for DCAT entities using MLT; and (ii) the data structure description in each distribution descriptor. In addition, from the generated model, we plan to implement an operational ontology with the gUFO ontology [29], a lightweight implementation of UFO, to support a FAIR Data Point [6] with native conformance.

In parallel, we aim to implement reference ontologies from the proposed conceptual model. Thus, this model (domain ontology) will function as a common terminology employed by human and machine agents. It will provide information about the basic concepts of the domain, and, through the multi-level approach, it will have elements for understanding the structuring/organization of the catalogue. These elements will contribute to develop solutions based on formal logic and Artificial Intelligence, optimizing search mechanisms and resource access.

Acknowledgments

This work has been partially supported with a student grant from CAPES (Process number 88887.613048/2021-00).

References

[1] Guizzardi G. Ontology, Ontologies and the "I" of FAIR. Data Intelligence. 2020;2(1-2):181–91. doi: 10.1162/dint_a_00040.

[2] Moreira JL, Bonino L, Ferreira Pires L, Van Sinderen M, Henning P. Towards findable, accessible, interoperable and reusable (fair) data repositories: Improving a data repository to behave as a fair data point. Liinc em Revista. 2019;15(2). doi: 10.18617/liinc.v15i2.4817.

[3] Sheridan H, Dellureficio AJ, Ratajeski MA, Mannheimer S, Wheeler TR. Data curation through catalogs: A repository-independent model for Data Discovery. Journal of eScience Librarianship. 2021;10(3). doi: 10.7191/jeslib.2021.1203.

[4] Wells D. Introduction to Data Catalogs. Alation, 2019. URL: https://www.alation.com/wp-content/uploads/Data-Catalogs-Intro-Dave-Wells-Alation.pdf

[5] Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, et al. The Fair Guiding Principles for Scientific Data Management and Stewardship. Scientific Data. 2016;3(1). doi: 10.1038/sdata.2016.18.

[6] Santos LOBS, Burger K, Kaliyaperumal R, Wilkinson MD; FAIR Data Point: A FAIR-Oriented Approach for Metadata Publication. Data Intelligence 2023; 5 (1): 163–183. doi: https://doi.org/10.1162/dint_a_00160.

[7] Albertoni R, Browning D, Cox S, Beltran AG, Perego A, Winstanley P. Data Catalog Vocabulary (DCAT) - Version 2. W3C Recommendation. 2020. URL: https://www.w3.org/TR/vocab-dcat-2/

[8] Borges V, de Oliveira NQ, & Campos MLM. A Multi-level Ontology-based Approach for Descriptors of Catalogued Resources. In Proceedings of the 15th Seminar on Ontology Research in Brazil and 6th Doctoral and Masters Consortium on Ontologies; 2022 Nov 22-25. V. 3346, pp. 46-59. CEUR-WS.

[9] Labadie C, Legner C, Eurich M, Fadler M. Fair enough? enhancing the usage of enterprise data with data catalogs. 2020 IEEE 22nd Conference on Business Informatics (CBI). 2020;201–10.; Antwerp, Belgium. doi: 10.1109/CBI49978.2020.00029.

[10] Dibowski H, Schimid S, Svetashova Y, Henson C, & Tran T. Using Semantic Technologies to Manage a Data Lake: Data Catalog, Provenance and Access Control. In SSWS@ ISWC, 2020. p. 65-80). URL: https://ceur-ws.org/Vol-2757/SSWS2020_paper5.pdf

[11] Connolly D. Catalogs: Resource Description and Discovery. W3C. 24 February 2014. URL: https://www.w3.org/Search/catalogs.html

[12] Satija MP, Bagchi M, Martínez-Ávila D. Metadata management and application. Library Herald, v. 58, n. 4, 2020. p. 84-107, doi: 10.5958/0976-2469.2020.00030.2.

[13] Greenberg J. Understanding metadata and metadata schemes. Cataloging & classification quarterly, 40(3-4), 2005. p.17-36. doi: 10.1300/J104v40n03_02.

[14] Cole TW. Creating a framework of guidelines for building good digital collections. 2002.

[15] Mylopoulos J. Conceptual modelling and Telos, Conceptual Modeling, Databases, and Case An integrated view of information systems development., 1992. p. 49–68.

[16] Sales TP. Ontology validation for managers. Universidade Federal do Espírito Santo, Vitória, Brazil. 2014.

[17] Guizzardi G, Zamborlini V. Using a trope-based foundational ontology for bridging different areas of concern in ontology-driven conceptual modeling. Science of Computer Programming, v. 96, 2014. p. 417-443, doi: 10.1016/j.scico.2014.02.022.

[18] Carvalho VA, Almeida JPA, Fonseca CM, Guizzardi G. Multi-level ontology-based conceptual modeling. Data & Knowledge Engineering, v. 109, p. 3-24, 2017.

[19] Guizzardi G, Fonseca CM, Almeida JPA, Sales TP, Benevides AB, & Porello D. Types and taxonomic structures in conceptual modeling: A novel ontological theory and engineering support. Data & Knowledge Engineering, 134, 101891.2021. doi: 10.1016/j.datak.2021.101891.

[20] Guizzardi G, Benevides AB, Fonseca CM, Porello D, Almeida JPA, & Sales TP. (2022). UFO: Unified foundational ontology. Applied ontology, (Preprint), 1-44.

[21] Fonseca CM, Guizzardi G, Almeida JPA, Sales TP, & Porello D. (2022, October). Incorporating Types of Types in Ontology-Driven Conceptual Modeling. In Conceptual Modeling: 41st International

Conference, ER 2022, Hyderabad, India, October 17–20, 2022, Proceedings (pp. 18-34). Cham: Springer International Publishing. doi: 10.1007/978-3-031-17995-2_2.

[22] Carvalho VA. Foundations for Ontology-based Multi-level Conceptual Modeling. 2016. 167 f. 2016. URL: http://nemo.inf.ufes.br/wp-content/papercite-data/pdf/foundations_for_ontology_based_multi_level_conceptual_modeling_2012.pdf.

[23] Carvalho VA; Alameida JPA. Toward a well-founded theory for multi-level conceptual modeling. Software & Systems Modeling, v. 17, n. 1, p. 205-231, 2018. DOI: 10.1007/s10270-016-0538-9.

[24] Almeida JPA, Carvalho VA, Brasileiro F, Fonseca CM, & Guizzardi G. 2018. Multi-level conceptual modeling: Theory and applications. In Proceedings of the XI Seminar on Ontology Research in Brazil and II Doctoral and Masters Consortium on Ontologies, São Paulo, Brazil, 2018 Oct 1-3. V. 2228, pp. 26-41. CEUR-WS.

[25] Cardelli L. Structural subtyping and the notion of power type. In: Proceedings of the 15th ACM SIGPLAN-SIGACT symposium on Principles of programming languages. 1988. p. 70-79.

[26] Odell J. Power types. In: Journal of Object-Oriented Programing, 7(2), 1994. pp. 8-12.

[27] FAPESP. FAPESP COVID-19 Data Sharing/BR. URL: https://repositoriodatasharingfapesp.uspdigital.usp.br/ [Accessed: 30- January- 2023]

[28] Donohue T. DSpace 7.x Documentation, fev 03, 2022. URL: https://wiki.lyrasis.org/display/DSDOC7x/DSpace+7.x+Documentation.

[29] Almeida JPA, Guizzardi G, Falbo RA, Sales TP. gUFO: a lightweight implementation of the Unified Foundational Ontology (UFO). URL: http//purl org/nemo/doc/gufo. 2020.