Modeling and Exploring Semantic View Metadata in Knowledge Graphs with VoSV

Tulio V. Rolim[1,4,†], Renato S. Freitas[1], Narciso Arruda[1], Asley Lioba[1], José W. F. da Silva[1], Chiara Renso[2], Marco A. Casanova[3] and Vania M. P. Vidal[1,*,†]

[1] *Federal University of Ceará (UFC)*
[2] *ISTI Institute of National Research Council – Pisa – Italy*
[3] *Department of Informatics – Pontifical Catholic University of Rio de Janeiro*
[4] *Federal Institute of Education, Science and Techonology of Piauí*

Abstract

A Knowledge Graph (KG) provides a semantic framework for integrating and managing diverse data organizations. This paper introduces *VoSV* (Vocabulary of Semantic View), a domain-independent vocabulary designed to annotate metadata in semantic views based on a structured Data Design Pattern (DDP). The DDP organizes data into a four-level hierarchical model, supporting scalable maintenance and context-aware exploration. At the core is the Metadata Graph, built using *VoSV*, which captures detailed, machine-readable metadata about structure, provenance, and quality. These semantic annotations enhance key governance functions such as data lineage, quality evaluation, and usability. The paper also presents an interactive tool for exploring metadata, allowing users to visually inspect metadata elements across multiple levels of the semantic view metadata graph. Together, the DDP, *VoSV*, and exploration tool promote transparency, accountability, and trust by offering a structured, semantic approach to documenting and managing data within KGs.

Keywords

Vocabulary, Metadata, Knowledge Graph, Ontology, Semantic Integration

1. Introduction

A Knowledge Graph (KG) integrates different types of data sources to into well-grounded knowledge management, integration and intelligent analysis source. [1] cite that well formulated KG is really necessary when influencing large language models (LLMs). Though LLMs exhibit prowess in understanding and generating natural language, they mainly rely on probabilistic correlations in order to generate their outputs, which may lead to factual inaccuracies in the information in their output and also to semantic ambiguity and lack of specificity when the models are employed in specific domains. KGs, on the other hand, define the content explicitly in a structured and meaningful way, allowing it to be read by machines and such content is expressed in terms of the entities.

A KG ensures consistency, factual reliability, and contextual adequacy by provisioning access to verified information for concept disambiguation and maintaining logical coherence across the different semantic tasks that would, in turn, result in delivering applications that are more consistent and semantically stronger when applied to knowledge-intensive domains.

In this perspective, a semantic view of a KG provide a unified ontological framework emerging from the semantic integration of the data sources in a data lake [2]. This integration establishes a

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.
† These authors contributed equally.
tulio.xcrtf@gmail.com (T. V. Rolim); jrenatosfreitas@gmail.com (R. S. Freitas); narcisoarruda@gmail.com (N. Arruda); asleylc@gmail.com (A. Lioba); jwellingtonfranco@gmail.com (J. W. F. d. Silva); chiara.renso@isti.cnr.it (C. Renso); casanova@inf.puc-rio.br (M. A. Casanova); vaniap.vidal@gmail.com (V. M. P. Vidal)
0000-0002-6662-4938 (T. V. Rolim); 0009-0009-4359-9440 (R. S. Freitas); 0000-0003-3873-8468 (N. Arruda); 0009-0004-1162-9397 (A. Lioba); 0000-0001-9093-0428 (J. W. F. d. Silva); 0000-0002-1763-2966 (C. Renso); 0000-0003-0765-9636 (M. A. Casanova); 0000-0002-9370-8598 (V. M. P. Vidal)

comprehensive and coherent organizational data environment, enabling seamless access and fostering streamlined decision-making processes.

The construction and maintenance of a semantic view in KGs present three major challenges: (i) extraction and transformation – integrating data from heterogeneous sources within a data lake into a unified representation, using a shared vocabulary defined by a semantic view ontology; (ii) semantic linking is the process of making connections between semantically equivalent entities from various sources so that they can align semantically; (iv) data fusion and quality improvement is the merging of many representations of the same real-world object into one logical representation to enhance data quality.

To address these challenges, [3] defines a DDP that supports all three (3) of the above mentioned elements of Data Design Patterns by organizing the data in KGs in a logical manner. The DDP will help to structure the data and metatada based on four distinct hierarchical data layers to support semantic data integration and provide easier ongoing maintenance of the semantic view in different contexts.

This paper introduces VoSV (Vocabulary of Semantic View), a domain-independent vocabulary, as a mechanism for creating comprehensive annotations for the metadata of semantic views created using the DDP. The metadata graph, which is the foundation of this architecture, is generated on the basis of VoSV. It also provides explicit and machine readable representations of the structure, provenance, and quality of the data in the semantic views. VoSV provides the basis for a number of important data governance functions, including the ability to track lineage, assess quality and improve the usability of data, through rich semantic annotations.

In addition, the paper also describes an interactive tool that will allow users to browse and retrieve metadata. By providing a structured and semantically grounded framework for documenting and governing digital resources, the DDP methodology, VoSV ontology, and metadata exploration tool will help to advance transparency and trust in digital ecosystems The DDP methodology, VoSV ontology, and Metadata Exploration Tool support visual inspection of various metadata types (e.g., provenance, quality) across multiple levels of granularity, creating a common language for documenting and governing digital resources.

The remainder of this paper is structured as follows. Section 2 introduces a four-level architecture for logically organizing data and metadata within a semantic view of a KG. Section 3 presents the core classes and relationships defined in the *VoSV* vocabulary. Section 4 describes the construction and exploration of the metadata graph for a semantic view modeled using *VoSV*. Section 5 discusses related works. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Data Design Pattern for Constructing a Semantic View

This section introduces a data design pattern, referred to as *DDP_SV*, specifically developed to logically organize both data and metadata within the semantic view [3]. At the core of the proposed framework, the semantic view is composed of two interconnected knowledge graphs: the **data graph** and the **metadata graph**. The data graph in Figure 1(a) represents the actual integrated data within the semantic view and works as the informational backbone of the knowledge graph. It is structured into a four-level hierarchical architecture, each level encapsulating a distinct stage of semantic integration:

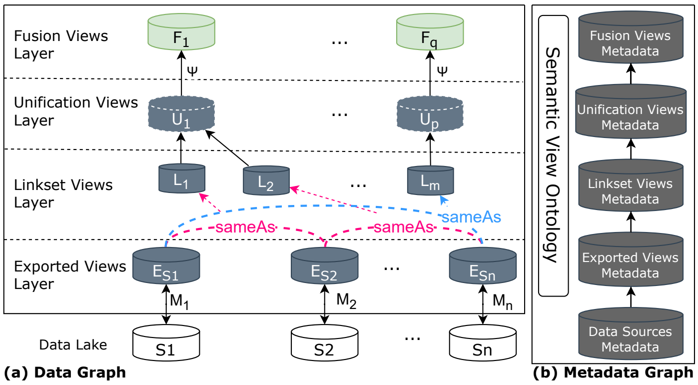

Figure 1: Data Design Pattern *DDP_SV* for KG's Semantic Views.

- **Exported Views Layer** – This foundational layer consists of RDF views generated by exporting data from raw sources in the data lake. Each exported view (EV) is obtained through an explicit mapping process that aligns the schema of a given data source (DS) with a shared vocabulary defined in the semantic view ontology (SVO). These mappings are specified by associating attributes from the DS with corresponding classes and properties in the SVO, typically through mapping languages such as RML/R2RML.

- **Linkset Views Layer** – This layer establishes semantic connections between equivalent entities across different exported views using identity relations such as *owl:sameAs*. The linkset views support semantic alignment and cross-referencing, forming the basis for further integration.

- **Unification Views Layer** – Built on top of the linkset views, this layer performs the integration of semantically equivalent entities into a single, canonical representation. The goal of unification is to ensure that all resources referring to the same real-world object under a common identity.

- **Fusion Views Layer** – The topmost layer addresses and resolves conflicts that may arise when the canonical representations produced by the unification views contain inconsistent or contradictory information. Building upon the aggregated representations produced by the unification views, this layer applies conflict resolution strategies to produce a consolidated and consistent view.

The metadata graph, depicted in Figure 1(b), serves as the repository for all metadata related to the semantic view. It plays a critical role in describing both the SVO and the views in all levels of the *DDP_SV*. Crucially, the metadata graph is semantically linked to the data graph, enabling integrated operations and contextual awareness. This tight integration supports a holistic understanding of the semantic view, empowering metadata-driven processes such as data discovery, quality assessment, lineage tracking, semantic governance, and view reuse.

3. Modeling Semantic View Metadata

This section presents the core classes of the *VoSV* (Vocabulary of Semantic View) and their interrelationships to represent the metadata of semantic views constructed using the *DDP_SV* design pattern.

Figure 2 provides an overview of the *VoSV*, highlighting its primary components and the semantic links between them. At the center of the model is the core class *vosv:SemanticView*, which conceptually encapsulates the main components of a semantic view. These components—and their respective roles within the metadata graph are described in the following subsections.

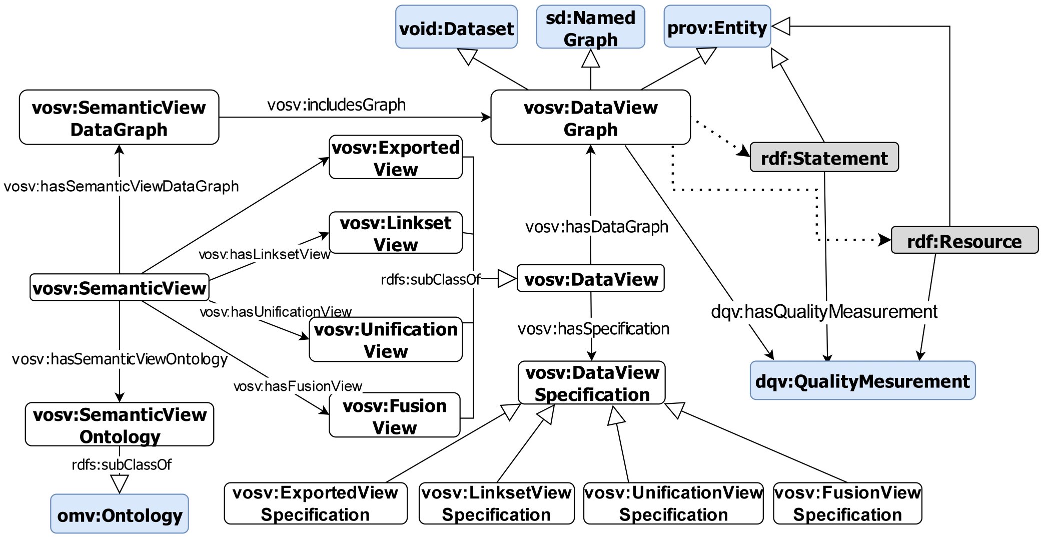

Figure 2: Overview of *VoSV* Ontology.

3.1. Reused Vocabularies

For modeling and construction of *VoSV*, standard W3C recommendation vocabularies were reused to allow the representation of metadata for elements of the knowledge graph construction process based on semantic integration. The Table 1 presents a list of reused vocabularies and their properties and concepts used in VoSV.

Vocabulary	Namespace	Properties / Concepts Used
Dublin Core	<http://purl.org/dc/terms/>	description, creator, created, format, issued, modified, language, publisher, contributor, subject, license
VOID	<http://rdfs.org/ns/void#>	Dataset, Linkset, vocabulary, dumpDataset, linkPredicate
PROV-O	<http://www.w3.org/ns/prov>	used, atTime, generatedAtTime, wasGeneratedBy, wasAttributedTo, Entity, Activity
PAV	<http://purl.org/pav/>	createdBy, providedBy, createdOn, lastUpdateOn, sourceLastAccessedOn, generatedAtTime
R2RML / RML	<http://www.w3.org/ns/r2rml#> <http://semweb.mmlab.be/ns/rml#>	TriplesMap, LogicalTable, LogicalSource, SubjectMap, PredicateObjectMap, source, template, sqlQuery, predicate
FOAF	<http://xmlns.com/foaf/spec/>	homepage
VANN	<http://purl.org/vocab/vann/>	preferredNamespacePrefix
OMV	<http://omv.ontoware.org/2005/05/ontology#>	Ontology, hasDomain

Table 1: Reused Vocabularies in VoSV.

3.2. Modeling the Semantic View Ontology

The semantic view ontology (SVO) serves as the foundational schema for constructing the semantic view of a KG. In the proposed framework, the SVO spans all layers of the *DDP_SV* and is built by integrating semantically equivalent elements across exported views.

In our incremental, pay-as-you-go approach, the SVO evolves progressively as new data sources are integrated. The initial version of the SVO mirrors the classes and properties of the first data source's ontology. When integrating a new DS, the SVO is extended to incorporate new classes and properties that accurately represent the semantics of DS. This integration involves two key processes:

- **Unification of Semantically Equivalent Properties** – Properties with similar semantic roles across different exported views are identified and unified under a common vocabulary. This standardization ensures a consistent vocabulary for describing attributes across all layers.
- **Establishment of Generalization Classes** – Semantically equivalent classes from various sources are grouped under shared superclasses, forming generalization classes. These provide an

abstract semantic foundation for the unification and fusion views, supporting alignment at the conceptual level. This systematic approach ensures that the SVO remains scalable, maintainable, and semantically coherent, even as new data sources are integrated.

3.3. Modeling DataViews in the Semantic View

In the *VoSV*, all those individual views that compose a semantic view-namely exported views, linkset views, unification views, and fusion views-are modeled as instances of the class *vosv:DataView*. Each *vosv:DataView* is made up of two major components (see Figure 2)-a data view specification and a data view graph, which will be described in the subsections that follow.

3.3.1. Specification of Data Views with vosv:DataViewSpecification

Conceptually, the specification of a semantic view is defined as the union of the specifications of its data views. The class *vosv:DataViewSpecification* provides a declarative description of how a particular data view is constructed. It captures: (i) the input data sources utilized; (ii) the transformations applied to the data; and (iii) the structure of the resulting RDF triples.

To support specialization based on the four types of data views defined in the *DDP_SV* architecture, *vosv:DataViewSpecification* is further refined into the four subclasses described in the following (refer to Figure 2).

(i) Specification of Exported Views. The *vosv:ExportedViewSpecification* class defines the specification for exported views through three core properties:

- ***vosv:hasDataSource*** – specifies the source dataset from which the view extracts data;
- ***vosv:hasOntology*** – describes the classes and properties utilized in the exported view. This ontology fragment represents a subset of the SVO;
- ***vosv:hasMappings*** – defines the schema mappings and transformation logic connecting the source data to the RDF representation.

(ii) Specification of Linkset Views. The *vosv:LinksetViewSpecification* class captures the essential components of a linkset and is described by properties:

- ***vosv:hasSourceClass*** – identifies the source class (subject) from which links originate;
- ***vosv:hasTargetClass***– identifies the target class (object) to which links point;
- ***vosv:hasMatchFunction*** – describes the logic or rules used to establish semantic links (e.g., *owl:sameAs*) between source and target instances.

(iii) Specification of Unification Views. The subclass *vosv:UnificationViewSpecification* defines the specifications for unification views using two main properties:

- ***vosv:hasGeneralizationClass*** – specifies the generalization class (from the SVO) whose instances are to be unified;
- ***vosv:hasNormalizationFunction*** – defines the function used to canonicalize multiple IRIs representing the same entity into a single target IRI.

(iv) Specification of Fusion Views. The subclass to specification of fusion views is modeled by *vosv:FusionViewSpecification*, including:

- ***vosv:hasUnificationView*** – references the input Unification View that supplies the data;
- ***vosv:hasPropertyFusionAssertion*** – defines one or more property-level fusion assertions that resolve conflicting values for specific unified properties.

3.3.2. Representing the RDF Graph of a Data View with vosv:DataViewGraph

The class *vosv:DataViewGraph* represents the concrete RDF implementation of a data view within a semantic view. It encapsulates the actual RDF triples generated through either the materialization or virtual execution of the transformation logic defined in the corresponding view specification. Each *vosv:DataViewGraph* supports rich metadata annotations that describe its structure, provenance, and quality. To enable this, the class *vosv:DataViewGraph* is defined as subclass of the following foundational classes:

- ***void:Dataset*** – to support statistical and structural descriptions of the dataset, such as triple count, class usage, and property distribution;
- ***prov:Entity*** – to capture provenance metadata, including generation activities, responsible agents, and derivation history;
- ***sd:NamedGraph*** – to support SPARQL service descriptions, allowing the graph to be queried or accessed as part of a SPARQL endpoint.

In RDF, the triples contained within a named graph are effectively represented as quadruples, where the fourth element specifies the graph URI. This structure allows each triple in the semantic view to be explicitly associated with its corresponding *vosv:DataViewGraph*, enabling precise tracking of which view produced which data.

The class *vosv:SemanticViewDataGraph* represents the complete semantic view data graph of the semantic view, which is formed as the union of the *vosv:DataViewGraph* instances associated with all its constituent data views (e.g., exported, linkset, unification, and fusion views). The class *vosv:SemanticViewDataGraph* is defined as a subclass of *void:Dataset*, and each of its *vosv:includesGraph* properties directly references the IRI of a named graph, which corresponds to an instance of *vosv:DataViewGraph*.

3.4. Modelling Quality Metadata

Quality metadata plays a critical role in ensuring that a semantic view within a KG adheres to high standards of reliability, consistency, and trustworthiness [4]. These annotations capture key data quality dimensions—such as completeness, consistency, and coherence—and may also include constraints and validation rules designed to enforce data integrity.

The overall quality of a semantic view is assessed based on the data quality of its constituent data views, including exported, linkset, unification, and fusion views. To support a nuanced and scalable assessment process, data quality annotations are applied at four distinct levels of granularity:

- **Triple Level** – At the most granular level, individual RDF triples are evaluated to assess precision, correctness, or conflict indicators;
- **Instance Level** – The quality of a resource or instance is derived from the aggregated quality of its constituent triples;
- **Data View Level** – A data view's overall quality is evaluated by integrating the quality metrics of all instances within the data view;

- **Semantic View Level** – The quality of the complete semantic view is computed by aggregating the results of its component data views.

To ensure semantic interoperability and alignment with established standards, the *VoSV* adopts the class *dqv:QualityMeasurement* from the Data Quality Vocabulary (DQV)[5]. This class enables the representation of both quantitative and qualitative assessments of data quality. Each measurement is associated with a metric, which defines a standard procedure for evaluating a specific data quality dimension by observing concrete features of the data.

As illustrated in Figure 2, the classes *vosv:DataViewGraph*, *rdf:Resource* (representing individual entities), and *rdf:Statement* (leveraging RDF-Star to reference specific triples) are linked to instances of *dqv:QualityMeasurement* through the property *dqv:hasQualityMeasurement*. This design enables a structured and fine-grained approach to data quality evaluation across all levels of the semantic view, supporting detailed assessment at the graph, resource, and triple levels. By following this multi-level evaluation process, the *VoSV* provides a structured, extensible, and standards-compliant approach to data quality annotation.

3.5. Modelling Provenance Metadata

Provenance metadata captures information about the origin, derivation, and transformation of the data represented in the semantic view. It plays a critical role in ensuring transparency, accountability, and trust in integrated data by documenting where the data came from, how it was processed, and by whom.

Within the *VoSV*, provenance is consistently modeled across the different levels of the semantic view architecture defined by the DDP_SV. Specifically, provenance metadata is expressed at three levels: data view, resource and triple.

To ensure interoperability and adherence to established standards, the *VoSV* reuses the PROV-O[6] ontology to describe the provenance of data views graph, resources, and triples.

The class *vosv:DataViewSpecification* captures provenance information about the construction of each data view, such as the input data sources, the transformation or mapping strategies applied, and the declarative specifications used. This high-level provenance enables users to trace the data flow and decision points in the creation of each view.

The class *vosv:DataViewGraph*, which represents the implementation of the RDF graph resulting from the view, also includes provenance metadata—such as the agents responsible for generating the data (*prov:wasGeneratedBy*), and the activities that led to its creation (*prov:Activity*). This information supports auditability and enables verification of the trustworthiness of the produced graph.

At a finer granularity, provenance metadata can be computed and attached to individual resources and triples. Example, a resource of a fusion view can be annotated with details about which exported view resources it originated from and which fusion rule was applied. This level of detail is essential to enable fine-grained lineage tracking, auditable data integration, and knowledge construction.

4. Constructing and Exploring the Semantic View Metadata Graph

This section discusses the construction and exploration of the metadata graph of a semantic view modeled using the *VoSV* vocabulary. As a case study, we use the semantic view *SV_Music*, which integrates music-related data from two heterogeneous sources: *DBpedia* and *MusicBrainz*. These sources offer rich and complementary descriptions of musical artists.

4.1. Case Study: The Semantic View 'SV_Music'

Figure 3 illustrates a representative fragment of the semantic view *SV_Music*, which serves as the running example throughout the following sections. Figure 3(a) shows the metadata graph, annotated using *VoSV*, while Figure 3(b) presents the corresponding data graph.

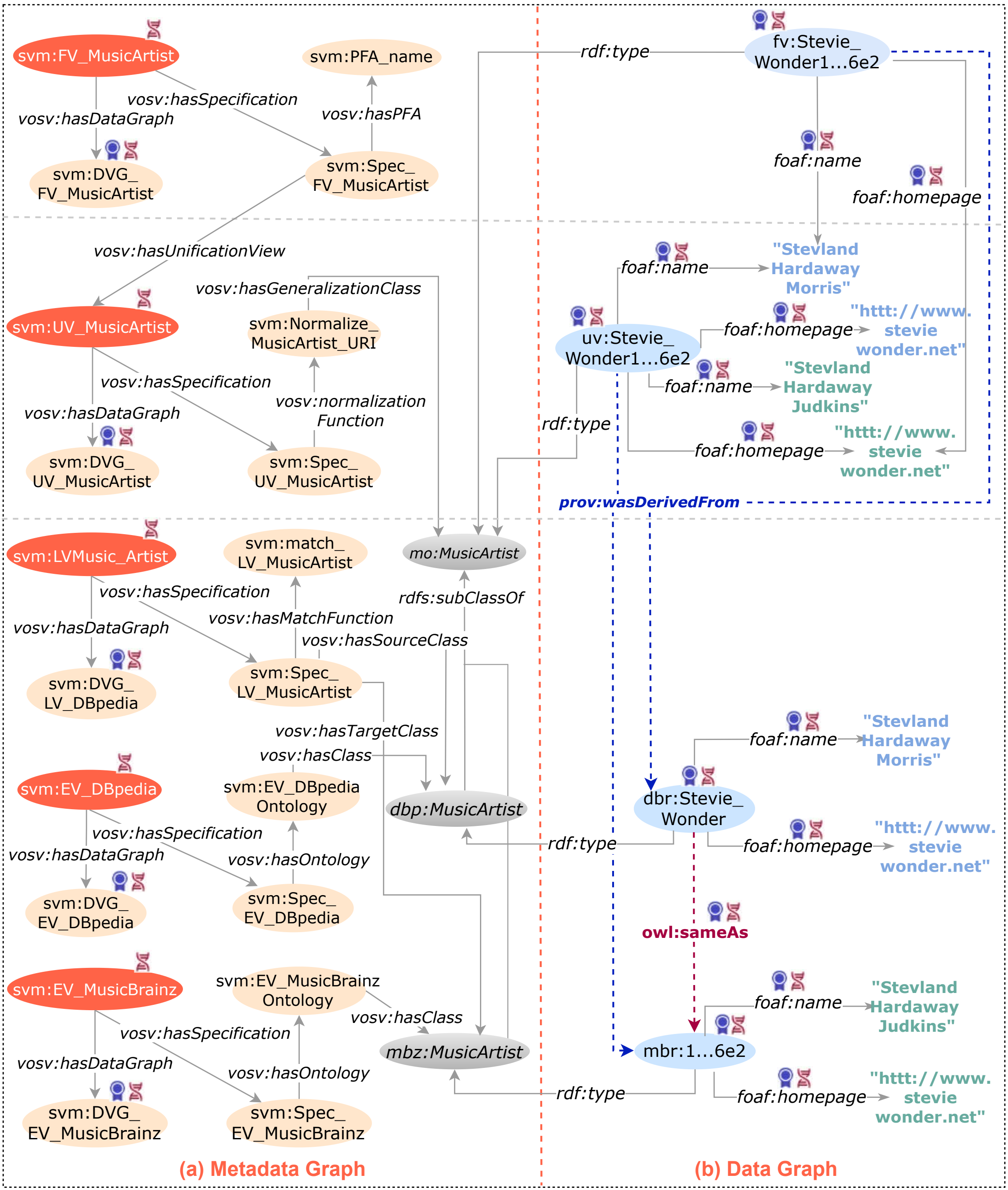

Figure 3: Fragment of *SV_Music* .

This figure highlights the framework's capability to explicitly link metadata elements to the data they describe, enabling integrated exploration and contextual understanding.

In Figure 3, quality and provenance metadata are visually indicated by the icons (quality) and (provenance), respectively. These icons appear alongside specific resources and RDF triples within the semantic view, indicating the presence of associated metadata and facilitating rapid identification and exploration of key trustworthiness indicators.

4.2. Constructing the Semantic View Metadata Graph

This section illustrates the use of the *VoSV* to construct the metadata graph of a semantic view following the *DDP_SV*. The *DDP_SV* supports a "pay-as-you-go" strategy [7][8], enabling the incremental construction and maintenance of the semantic view as new data sources are integrated or updated over time. The approach consists of five main steps, which are detailed below.

4.2.1. Step 1 – Semantic View Ontology Modeling

As discussed in Section 3, the *DDP_SV* supports an incremental, pay-as-you-go approach to modeling the SVO, which involves two key activities: (i) the unification of semantically equivalent properties across different data sources and (ii) the establishment of generalization classes to represent common concepts shared across datasets. The detailed process of designing and evolving the SVO is beyond the scope of this paper.

The ontology of the semantic view *SV_Music*, named *SV_Music_OWL*, is constructed by uniting the vocabularies of data sources *DBpedia* and *MusicBrainz*. Figure 4 depicts, in UML, the main classes of the *SV_Music_OWL*. It reuses terms from three well-known vocabularies: *Dublin Core* (DC), *Friend of a Friend* (FOAF), and *Music Ontology* (MO).

SV_Music_OWL has the generalization classes *mo:Track*, *mo:Record*, and *mo:MusicArtist* to represent broader concepts encompassing more specific subclasses. For example, the generalization class *mo:MusicArtist* has the subclasses *dbp:MusicArtist*, and *mbz:MusicArtist*, which are specifically defined to annotate instances originating from the *DBpedia* and *MusicBrainz* data sources, respectively. This allows for tracking the provenance of the resources and visualizing the resource in different contexts.

The class *dbp:MusicArtist*, for example, is exclusively for the ontology of the *svm:EV_DBpedia*, meaning that instances of *dbp:MusicArtist* are derived from the *DBpedia* data source.

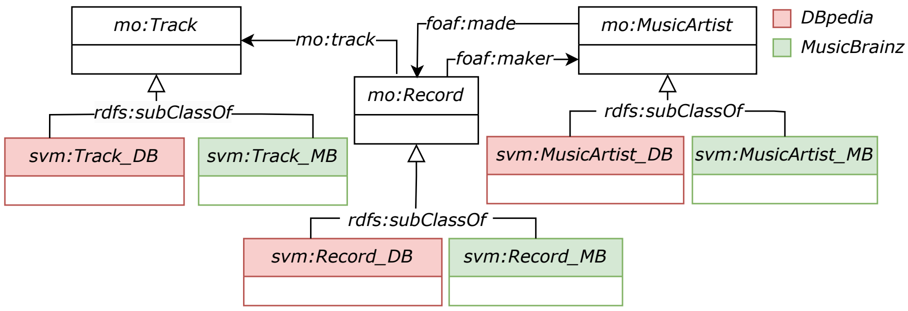

Figure 4: Main classes and relationships of SV_Music_OWL.

4.2.2. Step 2 – Exported View Construction

In this step, the objective is to construct an exported view for each data source by leveraging the existing semantic view ontology (SVO). The development of an exported view for a data source (DS) comprises two main tasks: (i) generation and validation of mappings from DS to SVO and (ii) implementation of the data graph, which can be virtual or materialized. The detailed process of constructing the export view is beyond the scope of this paper.

In the case study, the semantic view *SV_Music* includes two exported views: *svm:EV_DBpedia* and *svm:EV_MusicBrainz*. As illustrated in the metadata graph in Figure 3(a), each exported view is represented as an instance of *vosv:ExportedView*, encapsulating metadata that describes both its view specification and corresponding data graph. These data view graphs serve as access points to the RDF triples of each exported view and are implemented as distinct named graphs within the semantic view architecture.

4.2.3. Step 3 – Linkset View Construction

This phase focuses on creating linkset views to connect instances from an exported view to semantically equivalent instances in other exported views within a semantic view. The development of linkset views is carried out in three structured steps: (i) selection of source and target classes; (ii) definition of matching criteria; and (iii) creation of named graphs for linkset views.

Based on the class hierarchy defined in *SV_Music_OWL* in Figure 4, three linkset views are established to align semantically equivalent entities across sources: (i) *svm:LV_MusicArtist* – linking *dbp:MusicArtist* to *mbz:MusicArtist*; (ii) *svm:LV_Record* – linking *dbp:Record* to *mbz:Record*; and (iii) *svm:LV_Track* – linking *dbp:Track* to *mbz:Track*.

As shown in the metadata graph in Figure 3(a), the linkset view *svm:LV_MusicArtist* is modeled as an instance of *vosv:LinksetView*. It encapsulates metadata that defines both its view specification and the associated data graph, which together describe how identity links are established between musical artist entities from *DBpedia* and *MusicBrainz*. Figure 3(b) shows one example of sameAs triple (*dbr:Stevie_Wonder*, *owl:sameAs*, *mbr:1…6e2*) of *svm:LV_MusicArtist*.

4.2.4. Step 4 – Unification View Construction

In this study, a unification view must be defined for each generalization class present in the *SV_Music_OWL*. The construction of a unification view for a generalization class requires the definition of a "normalization function". This function is responsible for remapping all IRIs that refer to semantically equivalent entities—declared as such via *owl:sameAs* links—into a single canonical IRI. The purpose is to unify multiple representations of the same real-world entity across different exported views.

In the case study, within the semantic view *SV_Music*, three unification views are constructed: *svm:UV_MusicArtist*, *svm:UV_Record*, and *svm:UV_Track*. Each unification view consolidates semantically equivalent instances from different data sources into a single, unified representation. For example, the unification view *svm:UV_MusicArtist* merges instances of *dbp:MusicArtist* from *svm:EV_DBpedia* and *mbz:MusicArtist* from *svm:EV_MusicBrainz*.

Figure 3(b) illustrates an instance of *svm:UV_MusicArtist*, where resources *dbr:Stevie_Wonder* (from *DBpedia*) and *mbr:1…6e2* (from *MusicBrainz*) are identified as semantically equivalent. These resources are normalized to a canonical IRI, *uv:Stevie_Wonder1…6e2*, which aggregates all classes, attributes, and relationships from both *dbr:Stevie_Wonder* and *mbr:1…6e2*.

The implementation of the data graph for a unification view is virtual. That is, the unification view is computed dynamically at query time rather than materialized and stored in advance. During query execution, all properties and relationships associated with semantically equivalent IRIs are consolidated and presented under the canonical IRI defined by the normalization function.

4.2.5. Step 5 – Fusion View Construction

This step focuses on implementing fusion views to resolve data conflicts arising from overlapping properties in the unification views. This involves defining fusion strategies that reconcile discrepancies and consolidate information accurately. Fusion views ensure that the most reliable and relevant data is presented in the semantic view, enhancing data quality and trustworthiness.

To construct a fusion view involves three main tasks: (i) identification of properties with conflicts information; (ii) creation of property fusion assertions; and (iii) implementation of the fusion view data graph.

As shown in the metadata graph in Figure 3(a), the fusion view *svm:FV_MusicArtist* is represented as an instance of *vosv:FusionView*. It encapsulates metadata specifying both its view specification and the associated data graph. To address conflicts for the property *foaf:name*, a property fusion assertion (PFA) named *svm:PFA_name* was defined as part of the fusion logic. Figure 3(b) illustrates the result of this fusion process for the canonical resource *fv:Stevie_Wonder1…6e2*, which is generated by consolidating information from the resources *dbr:Stevie_Wonder*(from *DBpedia*) and *mbr:1…6e2* (from *MusicBrainz*).

4.3. Exploring the Metadata of Semantic Views

This section introduces a tool designed to support the interactive exploration and visualization of semantic view metadata. This tool is an enhanced version of the KG_Explorer platform [3], extended with new functionalities specifically developed for navigating and inspecting the metadata graph. These enhancements enable users to explore the structural, provenance, and quality metadata associated with each component of the semantic view.

In a semantic view constructed using the *DDP_SV*, as illustrated in Figure 3, resources and triples within the data graph are explicitly linked to their corresponding metadata in the metadata graph. This design enables integrated, contextual exploration of both data and metadata.

The tool allows users to inspect selected resources from the data graph in detail, including both their structural attributes and associated metadata. For example, Figure 5 displays the screen for visualizing the resource *dbr:Stevie_Wonder* within the data graph shown in Figure 3(a). The metadata displayed includes the resource's types—*dbp:MusicArtist* and the generalization class *mo:MusicArtist*—as well as the named graph to which the resource belongs: <http://blind-review/graph/ev_dbpedia>.

In the screen shown in Figure 5, icons displayed next to the resource's label and its properties provide direct access to the associated provenance and quality metadata, allowing users to quickly assess the reliability and origin of the data.

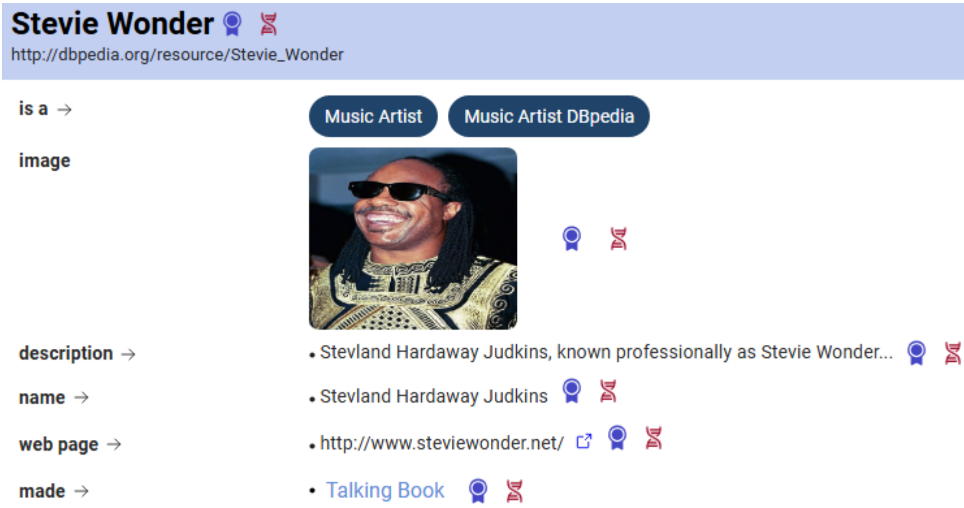

Figure 5: Exploration Screen for *dbr:Stevie_Wonder* resource.

The tool retrieves quality and provenance metadata by querying the metadata graph and displays the information within the resource's exploration screen. Users can interactively select and explore the specific metadata elements they wish to examine. Quality metadata is retrieved by querying the precomputed quality measurements (*dqv:QualityMeasurement*) of data quality dimensions: consistency, completeness and timeliness. The computation of provenance metadata depends on the context of the resource–whether it belongs to an exported view, unification view, or fusion view as discussed below.

In the context of exported views, the provenance of a resource is retrieved by querying the metadata of the corresponding exported view, specifically its data view specification and the associated data view graph. The keys metadata retrieved include: (i) the data source that resource was derived from and (ii) the mapping rules that generated the resource. The provenance of individual triples is also derived from the metadata of the exported view specification, particularly by referencing the mapping rules defined for the triple's predicate.

In the context of unification views, the provenance of a unified resource comprises several key components: (i) the generalization class that defines the semantic type of the unified resource; (ii) the original resources from the exported views that were unified, referenced using *prov:wasDerivedFrom*; (iii) the normalization function applied to canonicalize the IRIs of the resources into a single, unified IRI within the unification view. This normalization process is modeled as a *prov:Activity*, and the unified resource is linked to it using *prov:wasGeneratedBy*.

In the context of fusion views, the provenance of a consolidated resource includes: (i) the generalization class; (ii) the original resources from the exported views that were merged (referenced using *prov:wasDerivedFrom*); and (iii) the property fusion assertions (PFAs) that resolve conflicting values at the property level. Each PFA is modeled as a *prov:Activity*.

The provenance of a triple in the fusion view context includes the original source triples from the exported views, which are referenced using *prov:wasDerivedFrom*, and the specific PFA responsible for resolving conflicting values for that property, which is referenced using *prov:wasGeneratedBy*.

5. Related Works

To the best of our knowledge, there is no existing vocabulary directly comparable to *VoSV* in its comprehensive treatment of semantic view metadata within KGs. Nevertheless, other vocabularies and models address specific aspects relevant to the representation and governance of semantic views, which was briefly review below.

The Vocabulary of Interlinked Datasets (VoID) [9], recommended by the W3C, is a widely adopted standard for describing RDF datasets and their interconnections. It supports metadata for dataset cataloging, discovery, and linkage. However, VoID primarily provides a static description of datasets and their links, and lacks the expressive capabilities required to model the structure, provenance, and transformation logic of semantic views within a KG. The Vocabulary for Attaching Essential Metadata (VAEM), proposed by [10], provides mechanisms to annotate metadata for artifacts such as ontologies, vocabularies, and services. It includes support for describing authorship, publication dates, licensing, and dependencies—thereby facilitating reuse and discovery. Despite its utility, VAEM does not address the modeling of semantic data views or their internal transformations, and therefore can not capture the semantics of view definitions or associated data graphs.

The Data Knowledge Vocabulary (DKV) proposes an ontology for documenting knowledge about data, including aspects such as provenance, generation processes, and data quality [11]. While DKV aligns with the goals of transparency and explainability in data pipelines, it treats datasets as monolithic entities. It does not model the internal decomposition into multiple views, nor the specific transformations, alignments, or fusion operations that underpin semantic integration in KGs. The work presented in [12] represents a significant effort in using knowledge graphs to document data integration processes. The author proposes a semantic data model capable of integrating ontologies, data sources, and mappings rules. While the conceptual approach aligns with the goals of *VoSV*, it is primarily limited to representing metadata related to exported views. As a result, it does not support the annotation of other types of semantic views—such as linkset views and fusion views—which are essential for capturing the full range of integration strategies in a knowledge graph.

In contrast, *VoSV* fills this gap by offering a vocabulary specifically tailored to annotate metadata across all levels of a semantic view, including structural and semantic specifications in semantic views within dynamic semantic integration environments.

6. Conclusions and Future Work

This paper addressed the problem of constructing and exploring the metadata graph of a semantic view within a knowledge graph. The semantic view is organized into a four-level hierarchical structure that supports modularity, scalability, and semantic traceability.

The paper presented *VoSV*, a domain-independent vocabulary specifically designed to annotate metadata across all levels of semantic views. The metadata graph modeled using *VoSV* offers a powerful foundation for efficient data management and utilization of the semantic view in KGs. *VoSV* enables structured and standards-compliant representation of key metadata—such as data quality and provenance—across multiple levels of granularity. It helps add detailed notes down to each resource and triple. This setup makes it easier to track where data comes from, explain how information is combined, and build knowledge that can be checked. With this aware design, the semantic views in the KG become more open, trusted, and manageable. They also stay verified, reusable, and well-kept even in large data systems.

To support practical usage, we have also shared a handy tool that helps visually check metadata letting users look into parts like provenance chains, quality notes, and structural metadata.

As future work, we intend to carry out empirical studies to evaluate the work, including also validating *VoSV* and KG_Explorer with experts in semantic knowledge graph. In addition, we envision *VoSV* being applicable across a wide range of scenarios. Among these, two key use cases reflect our current interests:

- Empowering semantic view construction with LLM agents – The *VoSV*-based metadata graph provides large language model (LLM) agents with a structured, reliable foundation that supports the automated synthesis of data integration pipelines [13]. By formalizing semantic view specifications and exposing them through machine-interpretable metadata, *VoSV* enables the development of agentic systems capable of dynamically constructing, validating, and managing semantic views in complex data integration environments.
- Establishing a source of trust in LLM-powered question answering – By integrating curated data sources, formal validation mechanisms, rich metadata documentation, and structured governance models, the metadata knowledge graph serves as a robust pillar of trust. In this context, *VoSV* transforms LLMs from generic text generators into guided and accountable agents—grounded in contextual semantics, capable of delivering context-aware, verifiable answers [14].

7. Declaration on Generative AI

No generative AI tools were used in the creation of the scientific content, data analysis, and presentation of results in this article. Generative AI was only employed, for minor language polishing and proofreading. All ideas, experiments, and conclusions are the sole responsibility of the authors.

References

[1] J. Sequeda, O. Lassila, Building enterprise knowledge graphs, in: Designing and Building Enterprise Knowledge Graphs, Springer, 2021, pp. 97–128.

[2] M. Galkin, S. Auer, M. E. Vidal, S. Scerri, Enterprise knowledge graphs: A semantic approach for knowledge management in the next generation of enterprise information systems, in: International Conference on Enterprise Information Systems, volume 2, 2017, pp. 88–98.

[3] V. Vidal, R. Freitas, N. Arruda, M. A. Casanova, C. Renso, A data design pattern for building and exploring semantic views of enterprise knowledge graphs, in: Anais do XXXIX Simpósio Brasileiro de Bancos de Dados, SBC, Porto Alegre, RS, Brasil, 2024, pp. 1–13. URL: https://sol.sbc.org.br/index.php/sbbd/article/view/30678.

[4] T. Hartmann, B. Zapilko, J. Wackerow, K. Eckert, Constraints to validate rdf data quality on common vocabularies in the social, behavioral, and economic sciences, arXiv preprint arXiv:1504.04479 (2015).

[5] R. Albertoni, A. I. E. Debattista, M. Dekkers, C. Guéret, D. Lee, N. Mihindukulasooriya, A. Z. Contributors, et al., Data on the web best practices: Data quality vocabulary (2016).

[6] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, J. Zhao, Prov-o: The prov ontology, World Wide Web Consortium (2013).

[7] N. W. Paton, K. Christodoulou, A. A. Fernandes, B. Parsia, C. Hedeler, Pay-as-you-go data integration for linked data: opportunities, challenges and architectures, in: Proceedings of the 4th International Workshop on Semantic Web Information Management, 2012, pp. 1–8.

[8] J. F. Sequeda, W. J. Briggs, D. P. Miranker, W. P. Heideman, A pay-as-you-go methodology to design and build enterprise knowledge graphs from relational databases, in: The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Springer, Auckland, New Zealand, 2019, pp. 526–545. URL: https://link.springer.com/chapter/10.1007/978-3-030-30796-7_32.

[9] K. Alexander, R. Cyganiak, M. Hausenblas, J. Zhao, Describing linked datasets, in: Proceedings of the Linked Data on the Web Workshop (LDOW), 2009.

[10] R. Hodgson, I. Polikoff, Vocabulary for attaching essential metadata (vaem), https://lov.linkeddata.es/dataset/lov/vocabs/vaem, 2011.

[11] M. Böhmer, A. Dabrowski, E. Thordsen, Data knowledge vocabulary (dkv), https://lov.linkeddata.es/dataset/lov/vocabs/dk, 2017.

[12] S. Azizi, Documenting Data Integration Using Knowledge Graphs, Mestrado em ciência da computação, Gottfried Wilhelm Leibniz Universität, Hannover, 2023.

[13] D. Allemang, J. Sequeda, Increasing the llm accuracy for question answering: Ontologies to the rescue!, arXiv preprint arXiv:2405.11706 (2024). URL: https://arxiv.org/abs/2405.11706.

[14] J. Sequeda, D. Allemang, B. Jacob, Knowledge graphs as a source of trust for llm-powered enterprise question answering, Web Semantics 85 (2025) 100858. URL: https://www.sciencedirect.com/science/article/pii/S1570826824000441.