An Approach to Construct a Metadata Graph for Specifying Enterprise Knowledge Graphs

Tulio Vidal Rolim[1], **Caio Viktor Silva Avila**[1], **Renato Freitas**[1], **Roberval Gomes Mariano**[1], **Vania Maria Ponte Vidal** [1]

[1]Department of Computing – Federal University of Ceará (UFC)
Fortaleza – CE – Brazil

{tulio.xcrtf,arlaass,jrenatosfreitas,mariano.rgm1,vania.pvidal}@gmail.com

Abstract. *Enterprise Knowledge Graph (EKG) is a concept that is increasingly being used to process large volumes of data at the organizational level, however most efforts are concentrated on building the Data EKG, disregarding the metadata related to the entire data integration process. At this juncture, this work presents an approach for building a metadata graph for specifying EKGs. As a form of validation, a case study with part of the metadata graph is adopted.*

1. Introduction

In recent years, large-scale data management has become a critical challenge for organizations. The emergence of concepts such as Big Data and the need for complex analyzes demand solutions that not only store large volumes of data, but also allow for its efficient interpretation and analysis [Halevy et al. 2009]. In this context, Enterprise Knowledge Graphs (EKGs) emerge as a promising solution, allowing the representation of data in a semantic and interconnected format [Ehrlinger and Wöß 2016].

The construction of EKGs is often restricted only to the integration and generation of the Data Graph (KG-Data), with no attention paid to the representation and structuring of metadata, especially regarding the elements and artifacts that are generated during this process, making it difficult the maintenance, evolution and discovery of information under the graph in general and not just the data.

By also creating a Metadata Knowledge Graph (KG-Metadata), it is possible to organize elements such as EKG metadata in a structured and semantically relevant way, facilitating the management, understanding and use of the EKG in the organization and enabling new significant information to be linked to a format machine readable, which is an important step in creating FAIR data[da Silva Santos et al. 2023]. However, building an EKG metadata graph to represent, specify, and document the metadata of the EKG construction process is not a trivial task. Some problems are: a) little documentation; b) lack of sharing of artifacts from the semantic integration process; c) lack of consensual information or inaccuracy among members of the knowledge team responsible for building the EKG; d) standardization of metadata.

In this sense, considering these problems and that the construction of a metadata graph has not yet been treated in a consolidated manner in the literature, the focus of this work is to present an approach to support the specification and documentation of the EKG through a Metadata Graph. In summary, with the following contributions:

- A vocabulary to structure and specify the EKG metadata construction process;
- An approach to building a Metadata Graph (META-EKG);

2. VEKG - A Vocabulary to describe and specify the Metadata Graph

The architecture to construct an EKG has inspiration from the framework proposed by [Vidal et al. 2015] for building the Semantic View, serving as the base to construct the Metadata Graph (Meta-EKG). The architecture is assembled from the data graphs of the views of the three layers of the architecture: (1) **Exported Semantic Views:** RDF view of data following ontology; (2) **Linkset Views:** View containing *owl:sameAs* links between resources; (3) **Data Fusion Views:** View of fused and sanitized data. A conceptual view of the architecture used as a basis is presented in the Figure 1.

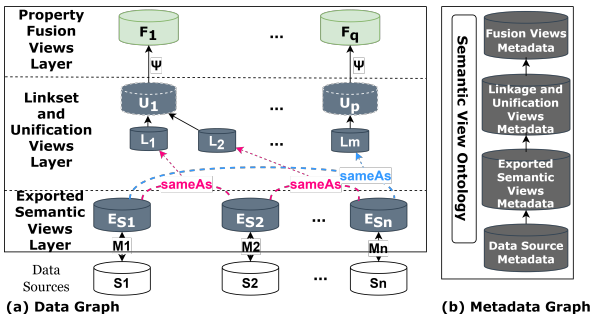

Figure 1. Architecture of Semantic Views.

In Metadata Graph the representation and structuring is done through the vocabulary to express the metadata of the elements and artifacts generated during the process of semantic integration and construction of the EKG named as (VEKG) available in: `https://tinyurl.com/3jedctv4`. VEKG is structured in RDF and its structure is based on the organization through six areas consistent with the EKG construction steps, reflecting direct navigability for artifacts and resources. VEKG can be used for various purposes, such as specifying metadata in new EKGs, or even already built ones, therefore generating an associated metadata graph, as well as through the reuse of metadata as a basis for guidance the construction of new EKGs. Figure 2 presents an overview of VEKG.

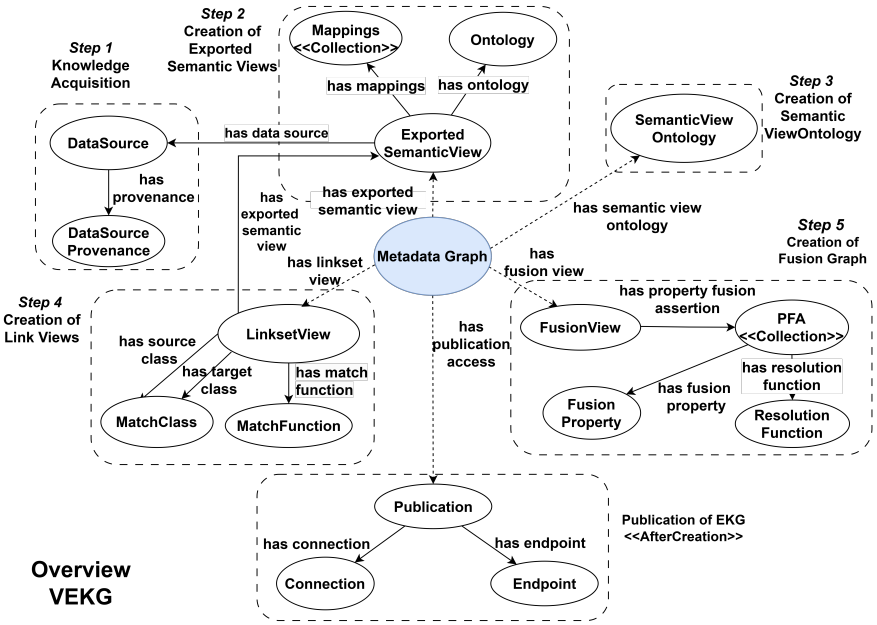

Figure 2. Main Classes of VEKG.

For modeling and construction of VEKG, standard W3C recommendation vocabularies (*Dublin Core, PROV-O, PAV, DQV, R2RML, FOAF, ODP, RDFS, OWL, VOID*) were reused to enable the representation of metadata for elements of the enterprise knowledge graph construction process based on semantic integration.

3. Approach to Construction of the Metadata Graph

In the proposed approach, a META-EKG is built in parallel with the development of the EKG itself, following the "pay-as-you-go" strategy. By adopting this strategy, organizations can gradually build their EKG, starting with a focused scope and gradually expanding it based on the value it brings [Sequeda et al. 2019]. This approach enables effective cost management, prioritization of investments and ensures that EKG evolves with the organization's changing needs and objectives.

The proposal for the incremental construction of the META-EKG comprises five primary steps and a one additional to express EKG access and publication metadata, where for each step carried out in the construction of the EKG, one specification and metadata generation activity occurs. Figure 3 shows the main EKG metadata and artifacts generated at each step.

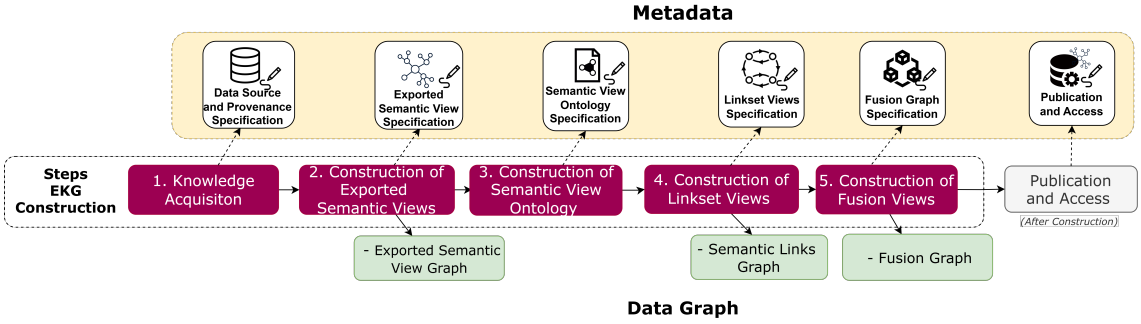

Figure 3. Approach to Construct Metadata Graph.

3.1. Step 1 – Acquisition of Knowledge

The metadata related to the knowledge acquisition step consists of specifying the metadata of data sources, provenance and associated artifacts. For data sources, the Data Catalog Vocabulary (DCAT)[Consortium et al. 2014] vocabulary is reused with additional types and properties. **vekg:DataSource** is a subclass of **dcat:DataCatalog**. It includes metadata reused by dublin core vocabulary such as title (**dc:title**), description (**dc:description**), publisher (**dc:publisher**), issuance date (**dc:issue**), and license (**dc:license**) to specify relevant general information from data sources. To define access to artifacts related to the data source, the properties **vekg:urlDataDictionary** for the data dictionary or manual, **vekg:urlDataSchemeDiagram** for the data schema diagram or model, and **vekg:urlScript** for the source script/file used, e.g *.json, .csv* or dump *.sql*.

Metadata and source provenance information are represented in the **vekg:DataSourceProvenance** class, used to represent provenance and capture essential information for each data source, such as: data provenance (authorship / entity publishing the data source), location, date of creation, date import, last update, last time accessed (for data sources external), update frequency, access and other metadata.

3.2. Step 2 – Construction of Exported Semantic Views

The step of constructing the semantic views of the sources involves the step of greater structuring in relation to the representation of each source in a semantic view consisting of mappings. Therefore, to specify relational sources, the R2RML[W3C 2012] structure (SubjectMap, LogicalTable, PredicateObject) is reused in VEKG together with complementary information about relational schemas, such as primary key fields, whether the attribute can be null, index, etc. Considering non-relational data sources, a structure similar to RML is adopted, maintaining standardization and ease of vocabulary reuse.

Mapping metadata is represented through a resource of the **vekg:Mappings** class type, a collection (**cp:Collection**) that stores a set of instances of the **vekg:RelationalMapping** or **vekg:LogicalMapping** type. The **vekg:RelationalMapping** Class has a relationship with a collection of type *TriplesMap*, a set of mappings based on the R2RML schema. Each mapping is specified and represented containing metadata on significant properties for clarity of transformation rules, as well as serving to identify relevant attributes and characteristics of the schemas used in the mappings to generate the EKG.

3.3. Step 3 – Construction of Semantic View Ontology of EKG

To perform the metadata specification in the Semantic View Ontology, some metadata is provided as: **vekg:subject_ontology** which covers subjects and concepts related to the vocabulary, this makes it easier to identify which terms and things are represented in the semantic view ontology. For access to artifacts related to the semantic view ontology, **vekg:urlOntology** and **vekg:urlClassDiagram, vekg:urlOntologySpecification** respectively provide a direct URL to the class diagram model, the ".OWL" file and the address of the ontology specification document. The semantic view ontology also provides direct access to the vocabularies through the **vekg:hasSemanticViewOntology** property.

Furthermore, popular metadata properties such as **dc:creator, dc:format, dc:issued, dc:language** are reused. This metadata provides detailed insight into the structure, authorship, format, version, issue date, languages, publishing entity, and access to the semantic view ontology. They are crucial for the understanding, reuse and integration of the ontology in different semantic and application contexts, promoting interoperability and standardization in the domain.

3.4. Step 4 – Construction of Linkset Views

To handle the construction of linkage views, VEKG provides metadata to represent and specify semantic links between two resources from distinct data sources. Semantic Links are represented in the class **vekg:LinksetView**, which represents the "Link Views". Each LinksetView establishing relationship with 2 classes **MatchClass** through properties **vekg:hasSourceClass** and **vekg:hasTargetClass**, respectively. The semantic link specification rules are defined by the **vekg:MatchFunction** class, related by the **vekg:hasMatchFunction** property, containing the link predicate.

A **vekg:MatchFunction** also aims to specify the types and properties of the resources of a **MatchClass** type considered to establish the "linkage" using *1* or *N* **MatchProperty**. In addition, each **LinksetView** is connected with *1* or *N* ExportedSemanticViews through the **vekg:hasExportedSemanticView** relationship. Defining a link relationship may involve the need to use comparison or transformation functions such as

aggregration, concatenation, replace and others. To specify these types of functions, a Match Function can have a Link Function of one of the types (Compare, Aggregation, Transformation).

3.5. Step 5 – Fusion Graph Construction

The objective of a Fusion view is to resolve conflicts that may arise when different sources provide divergent information about the same entity or relationship, aiming to improve the quality, accuracy, and reliability of the information in a KG of Data.

In VEKG, the Fusion Views are defined by one Property Fusion Assertion to structure the rule adopted to perform the fusion between two resources, one **vekg:FusionProperty** in which it designates the property that will have its value evaluated and one **vekg:ResolutionFunction** that provides information about the type (**vekg:type**) of the function (Largest / Smallest / Average) and the comparison metric (**vekg:metric**) (e.g, last update, reputation, etc) to a **vekg:Class** that establishes the type-/class of the resource.

4. Case study

As a case study, this section will present the preliminary results of part of the metadata generation of documentation for the construction process of a fragment of an EKG (named as EKG-SEFAZ) implemented in the real world in the domain of public tax data, built in partnership with the Maranhão State Treasury Department (SEFAZ-MA).

This EKG aims to semantically integrate the various heterogeneous data sources present in the public body, in addition to importing external data sources, such as data from the Brazilian Federal Revenue Service (RFB). The documentation for this process was generated as metadata using the VEKG vocabulary, as described in Section 3. Below, we describe how each step of the EKG construction is documented using the VEKG vocabulary. The resulting metadata graph can be accessed at the following link. To conduct a partial evaluation of the metadata graph created for this case study, exploratory queries were executed on a GraphDB endpoint: *https://graphdb.arida.site/*.

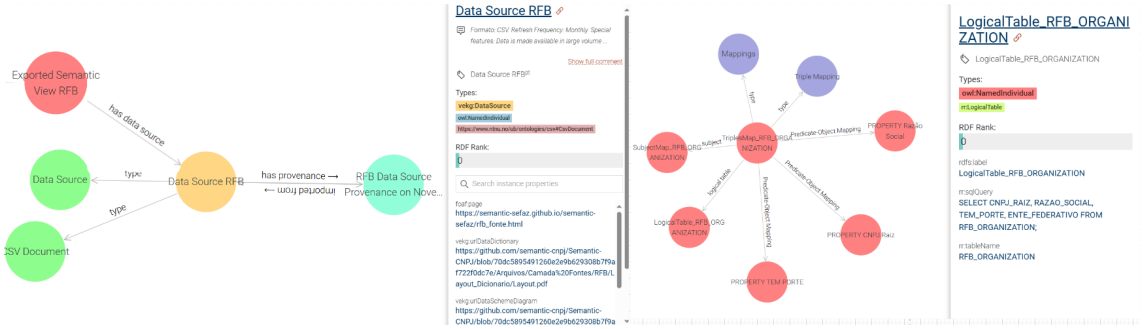

Figure 4. a) Metadata Graph Data Sources. b) Exported Semantic View Mappings.

In this case study, only two data sources will be used: (1) SEFAZ-MA Registration Data; (2) Registration Data from the Brazilian Federal Revenue Service (RFB). Therefore, at this stage, two instances of the **vekg:DataSourceProvenance** class are defined, representing the bodies providing the data sources: (1) **sfz:DataSource_CAD** and (2)

sfz:DataSource_RFB, representing SEFAZ-MA and RFB, respectively. Each instance contains attributes that identify the sources, such as: **createdOn, importedOn, lastUpdateOn, sourceLastAccessedOn, createdBy, importedFrom, frequencyOfChange, previousVersion**. Each instance is linked to its respective **vekg:DataSourceProvenance** through the **vekg:hasProvenance** relationship. Furthermore, each instance has attributes that characterize the data source and its artifacts, such as in Figure 4. a).

In other example (Figure 4 b), the necessary artifacts for creating a collection of RDF views of data sources are documents. Each instance of **vekg:ExportedSemanticView** is related to a collection of mappings that tell how to transform data from data sources, to RDF data following the local ontology structure defined for the respective source. For example, considering data sources stored in a relational source, the exported semantic views **sfz:Exported_Semantic_View_CAD_SEFAZ** and **sfz:Exported_Semantic_View_RFB** each have a mapping collection **sfz:Mappings_CAD** and **sfz:Mappings_RFB**, respectively by **vekg:hasMappings**.

The specification of these mappings using the R2RML vocabulary base allows data from relational databases to be transformed into RDF triples in a systematic way. In this case study, each mapping in the set of triples mappings (*TriplesMapping*). These define Logical Table, Subject Maps, and Predicate Object Maps (POMs) by triple mapping. For example, considering the mapping from Organization to RFB Data Source, **sfz:TriplesMap_RFB_ORGANIZATION**) has 1 Logical Table (**LogicalTable_RFB_Organization**), 1 SubjectMap (SubjectMap_RFB_Empresa) and 3 Predicate Object Map (**POM_CNPJ_raiz, POM_razao_social, POM_tem_porte**).

When specifying the LogicalTable, it is possible to obtain the name of the table involved in the mapping through the rr:tableName "RFB_ORGANIZATION" property, as well as a SQL query used **rr:sqlQuery** ***"SELECT CNPJ_RAIZ, RAZAO_SOCIAL FROM RFB_ORGANIZATION;"***, allowing verifiability and understanding of the data mapped directly from the relational source, also contributing to the clarity, verifiability and evolution of EKGs.

5. Conclusion

This paper presents a preliminary approach and its steps for building a metadata graph focused on specifying EKGs. First, the paper presents the architecture and concepts that underlie the approach for representing EKGs. Then, a vocabulary for describing and specifying the metadata graph is presented.

Therefore, the approach for specifying and constructing the metadata graph is presented, detailing the steps in line with the EKG. Finally, a case study is proposed to demonstrate as preliminary results a metadata graph constructed to represent an EKG that integrates 2 data sources.

Future work intends to expand the vocabulary of the metadata graph with other concepts related to the implementation and use of semantic data integration tools, in addition to evolving the vocabulary and its representation in RDF-star format.

References

Consortium, W. W. W. et al. (2014). Data catalog vocabulary (dcat).

da Silva Santos, L. O. B., Burger, K., Kaliyaperumal, R., and Wilkinson, M. D. (2023). Fair data point: a fair-oriented approach for metadata publication. *Data Intelligence*, 5(1):163–183.

Ehrlinger, L. and Wöß, W. (2016). Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE intelligent systems*, 24(2):8–12.

Sequeda, J. F., Briggs, W. J., Miranker, D. P., and Heideman, W. P. (2019). 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, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*, pages 526–545. Springer.

Vidal, V. M. et al. (2015). Specification and incremental maintenance of linked data mashup views. In *CAiSE*, pages 214–229. Springer.

W3C (2012). R2rml: Rdb to rdf mapping language.