An LLM-based Multi-agent Framework for Publishing RDF Views into Enterprise Knowledge Graphs

Asley Lioba[1], José Renato S. Freitas[1], Tulio Vidal Rolim[2], Vânia M. P. Vidal[1]

[1]Universidade Federal do Ceará (UFC)

[2]Instituto Federal do Piauí (IFPI)

{asleylc,jrenatosfreitas,tulio.xcrtf,vaniap.vidal}@gmail.com

Abstract. *The publication of heterogeneous data into Enterprise Knowledge Graphs (EKGs) remains challenging due to schema matching, semantic inconsistencies, and the effort required to create mappings and target ontology. This paper presents a metadata-driven multi-agent framework based on Large Language Models (LLMs) for generating RDF views from data sources. The framework orchestrates agents for schema extraction, ontology bootstrapping-refining, semantic mapping generation, and RDF materialization tasks. The approach leverages metadata, represented through the Vocabulary of Semantic Views (VoSV) into a data design pattern (DDP_SV), as in-context learning to support explainability, governance, and incremental publication. As a way to perform a preliminary evaluation of this framework, a case study using Brazilian public government purchases data sources demonstrates the feasibility of automating RDF view publication.*

1. Introduction

Enterprise Knowledge Graphs (EKGs) based in Resource Description File (RDF)[1] format provide a semantic layer capable of unifying heterogeneous organizational data sources into a shared conceptual representation (target ontology) in Ontology Web Language - OWL[2]). One of the main challenges in EKG construction is the publication of raw data as a knowledge graph, where data sources are transformed into the RDF view graph aligned with target ontology.

Traditional RDF views publication pipelines require significant manual effort for schema interpretation, ontology engineering, mapping construction and validation. Recent advances in Large Language Models (LLMs) introduced new possibilities for automating RDF View publishing tasks, especially in ontology generation, schema matching, and mapping creation [Mavridis et al. 2025].

This paper proposes a framework for publishing RDF view into EKGs. Unlike other approaches to knowledge graph construction, our proposal orchestrates specialized agents, based on LLM, that extract schemas, generate ontologies, build mappings and materialize semantically aligned RDF views from heterogeneous data sources. The proposed framework is built on two previous conceptual foundations: (i) Data Design Pattern for Semantic Views (*DDP_SV*); and (ii) Vocabulary of Semantic Views (VoSV).

[1]https://www.w3.org/RDF/
[2]https://www.w3.org/OWL/

The main contribution of this work is a framework for RDF view generation using specialized agents grounded by semantic metadata. The differential of this work is the proposal that specialized agents are guided in a Metadata-driven reasoning with human validation to publish RDF Views.

2. LLM-based Multi-agent Framework

The LLM-based multi-agent framework including its architecture and workflow, designed to proposed framework allows users to construct and publish RDF Views incrementally and interactively with the support of LLM-driven agents, while building and leveraging the RDF view's metadata graph, instantiated with VoSV, to enhance automation, explainability, and reusability. Figure 1 presents the proposed framework structure.

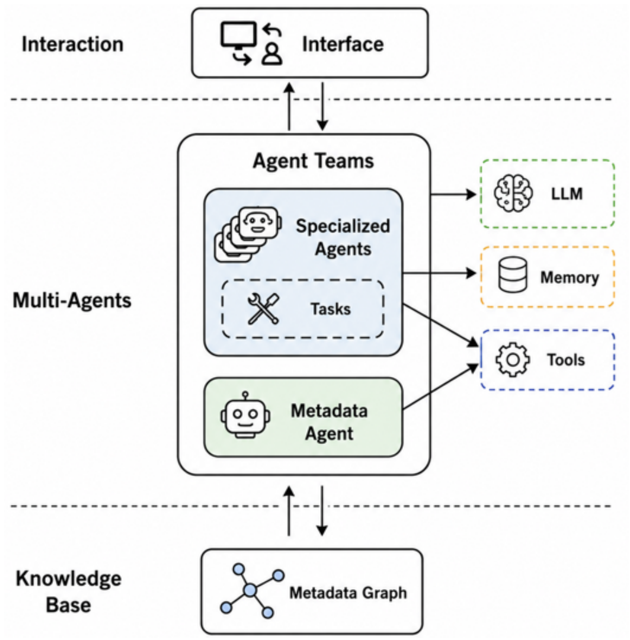

Figure 1. Multi-agent framework architecture for developing the RDF views.

Interaction: The interface component is responsible for communication between the user and the agents, with interactions generally expressed in natural language. In some cases, several interactions may be performed in a loop until the user is satisfied with the result generated by a specialized AI agent. This interaction is called *Human-in-the-loop* (HITL).

Multi-agents: This layer houses specialized AI agents that are organized into *teams* according to each level of the data design pattern *DDP_SV*[Vidal et al. 2024]. They are assigned one or more predefined tasks and can use tools, shared contextual memory, beyond LLMs for reasoning and task completion. An agent communicates with others, enabling the delegation of tasks and/or the dependency of tasks from different agents, generally orchestrated sequentially.

Knowledge Base: In our multi-agent framework, the knowledge base, composed of the data graph and the RDF view metadata graph, serves as a reliable source of information to be used in contextual learning by the specialized agents LLMs.

2.1. The Multi-agent Framework Workflow

In the proposed framework, the specialized agents are equipped with its own Retrieval-Augmented Generation (RAG) pipeline. This RAG pipeline is responsible for contextualizing tasks and adding the metadata retrieved by the *MetadataAgent* to the task prompts. Specialized agents interact with *MetadataAgent* to retrieve and update relevant metadata. Additionally, they can enter a user interaction loop to validate intermediate results.

The specialized agents of the *PublishingTeam* are designed to handle distinct aspects of publishing a data source in an EKG. Table 1 presents the multiple publishing agents and how they operate.

Table 1. Agents, inputs, outputs, and tasks.

Agent	Description	Input	Output	Tasks
[1]	Extracts the schema or data model from data sources. This includes relational sources and semi-structured sources such as CSV and JSON.	1. File path 2. Data source name	1. RDF graph containing the extracted data source schema	1. retrieves_csv_path 2. extracts_csv_schema 3. extracts_sql_schema
[2]	Transforms the extracted schema directly into an RDF/OWL ontology without customization. It identifies classes, properties, relationships, domains, ranges, and class hierarchies. In next, this agent identifies established and compatible vocabularies, facilitating their use in the ontology's classes, properties, and relations.	1. Extracted schema 2. Data source name	1. RDF/OWL bootstrap ontology 2. Schema matching	1. bootstrap_ontology
[3]	Creates descriptive artifacts that establish semantic correspondences between a data source schema and the target ontology. It identifies equivalences and subsumptions, and validates whether the mapping satisfies the required criteria.	1. Schema matching 2. Refined data source ontology 3. Data source schema	1. RML mapping 2. Metadata mapping	1. mapping_generator
[4]	Materializes the RDF view data graph from a data source using user-approved mappings generated by the Semantic Mapping Constructor and transformation engines that interpret these mappings.	1. RML mapping 2. Data source ontology	1. RDF view dataset 2. Provenance	1. triplification
[5]	Acts as a centralized component responsible for supporting metadata management and the tasks of the other specialized AI agents, providing the following services: Respond to metadata queries in alignment with the *DDP_SV* framework and the *VoSV* vocabulary from specialized agents; Record the outcomes of tasks performed by specialized agents into the metadata graph	1. Task Metadata Request	1. Query Result containing the metadata requested	1. query_metadata 2. insert_metadata

Legend: [1] SchemaExtractor; [2] BootstrapOntologyExtractor; [3] SemanticMappingConstructor; [4] RDFViewMaterializer; [5] MetadataAgent.

The workflow of activities carried out by the *PublishingTeam* agents is performed sequentially, where at each stage outputs are generated, as shown in Figure 2

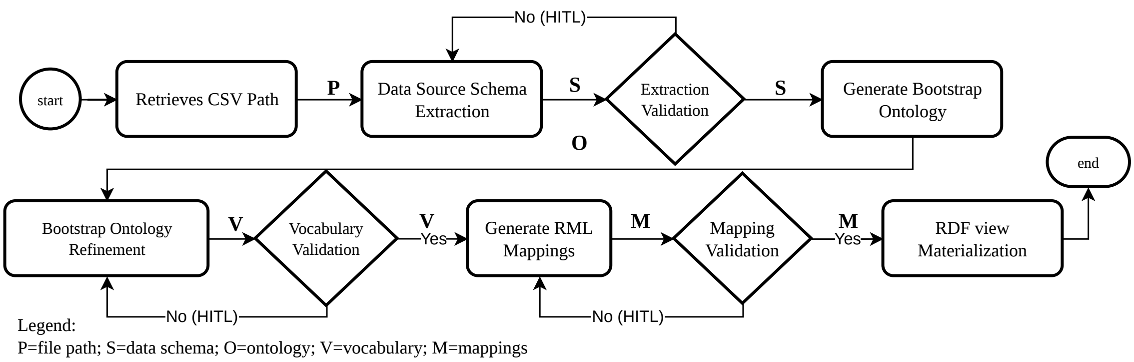

Figure 2. The Workflow of tasks for agents on the PublishingTeam.

3. Publishing Team in Action

This section presents our proposed LLM-based multi-agent framework assisting in the publishing of real and public Brazilian data sources: *government purchases* audited by the CGU, focusing on contracts between companies and public organizations. As a way to demonstrate the practical application of our framework, the agents, tasks, and workflow orchestration (available at [3]), were implemented using the CrewAI framework. The main LLM used was the llama-3.3-70b-versatile.

3.1. Step 1 - Data Source Schema Extraction

The case study of the publishing stage begins with a natural language request via the system interface to publish the CGU's government purchases data source. The *SchemaExtractor* agent, with the help of the *MetadataAgent*, initiates the retrieval of the path or connection string where the CGU's public purchases data are stored, previously registered in the metadata graph.

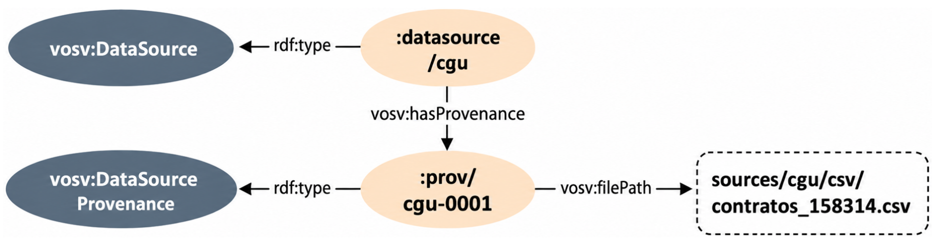

Figure 3. CGU data source metadata in the metadata graph.

Next, the *SchemaExtractor* agent identifies the data source type and runs the appropriate assigned task. Currently, there are two tasks for this agent: (i) one to extract schema from relational data; and (ii) another to extract schema from a CSV file. These tasks extract, primarily, the field names and their data types. In addition, they extract primary, unique and foreign keys, if any.

Once the CSV schema extraction task is completed, the outcome is the RDF graph representing the schema of CGU's public procurement data, with selected columns marked by the user through the vosv:isActive property. Columns assigned the value "true" are considered during the semantic mapping process, whereas those assigned "false" are ignored. This configuration is particularly useful when a data source contains dozens of tables and hundreds of attributes, many of which may be irrelevant for ontology alignment and mapping generation tasks.

3.2. Step 2 - Data Source Ontology Extraction

The *BootstrapOntologyExtractor* agent receives the CGU's public purchases data schema as input, in RDF format. This data schema is obtained by the *MetadataAgent* through a SPARQL query, generated from a dynamic prompt. The *MetadataAgent* interprets the user's initial request for semantic publication of the CGU data source and constructs the SPARQL query, whose dynamic prompt defines which data schema should be queried and retrieve the metadata of the CGU data schema.

[3]https://anonymous.4open.science/r/ekg-public-contracts-D18D

At the end of the bootstrap ontology generation task, the bootstrap ontology of CGU's public purchases shown in Listing 1 is generated. A positive example of the use of metadata used to augment the LLM reasoning is that, in this task, it inferred the relationship — which was implicit — between a contract and a public organization (which makes a purchase), as well as a supplier (which sells a product or service).

```
@prefix cgu: <http://semantic.br/contracts#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
### Classes
cgu:Contract rdf:type owl:Class .
cgu:PublicOrganization rdf:type owl:Class .
cgu:ManagementUnit rdf:type owl:Class .
cgu:Supplier rdf:type owl:Class .
### Object Properties
cgu:hasPublicOrganization a owl:ObjectProperty ;
   rdfs:domain cgu:Contract ;
   rdfs:range cgu:PublicOrganization .
cgu:hasSupplier a owl:ObjectProperty ;
   rdfs:domain cgu:Contract ;
   rdfs:range cgu:Supplier .
### Datatype Properties
cgu:codigoOrgao a owl:DatatypeProperty ;
   rdfs:domain cgu:PublicOrganization ;
   rdfs:range xsd:integer .
cgu:nomeOrgao a owl:DatatypeProperty ;
   rdfs:domain cgu:PublicOrganization ;
   rdfs:range xsd:string .
cgu:niFornecedor a owl:DatatypeProperty ;
   rdfs:domain cgu:Supplier ;
   rdfs:range xsd:integer .
cgu:nomeRazaoSocialFornecedor a owl:DatatypeProperty ;
   rdfs:domain cgu:Supplier ;
   rdfs:range xsd:string .
```

Listing 1. Fragment of Bootstrap ontology of CGU's public purchases generated for the OntologyExtractor agent.

3.3. Step 3 - RML Mappings Construction

The next phase of *PublishingTeam* is the *SemanticMappingConstructor* agent, response for construction relationship from datasource and ontology through the task *Mapping_Generator*, which receive the CGU ontology from last interaction, schema of CGU and metadata. The task can use human interaction to validate the generated RML.

```
# Mapping nomeOrgao to cgu:nomeOrgao
rr:predicateObjectMap [
rr:predicate <cgu:nomeOrgao>;
rr:objectMap [
 rml:reference "nomeOrgao";
 rr:datatype xsd:string
]].
```

Listing 2. Fragment of RML mapping generated from the CGU bootstrap ontology.

The *mapping_generator* task establishes a semantic connection between the source schema and the generated ontology by defining how source fields are transformed into RDF triples through RML mappings. For example, in the Listing 2, the statement *"rr:predicate cgu:nomeOrgao"* specifies the semantic property used to represent the extracted data in the target ontology, while the statement *"rr:objectMap [rml:reference "nomeOrgao"]"* indicates that the value associated with this property is obtained from the *nomeOrgao* column in the CSV source. Additionally, the declaration *"rr:datatype xsd:string"* defines the datatype assigned to the extracted value in the next step.

3.4. Step 4 - RDF View Materialization

The last agent of the *PublishingTeam* is *RDFViewMaterializer*, following the mapping definition approved by the human interact, and is tasked with the execution of the Triplification task. This agent uses previously generated RML rules as parameters as an initial configuration based in the *Morph-KGC* tool [Arenas-Guerrero et al. 2024]. This engine is used to performs the materialization by executing the RML against the data source, converting rows into RDF triples, populating the instances and properties, (see Figure 4):

Figure 4. Triples generated from RDFViewMaterializer.

4. Related Work

The evolution of knowledge graph engineering has shifted from expert-driven methodologies toward automated frameworks supported by LLMs. Early studies focused on reducing manual effort through machine-assisted environments, such as the human-dependent validation approach proposed by [Kommineni et al. 2024]. In specialized domains, research emphasized schema alignment and semantic precision, as demonstrated in industrial mappings [da Silva et al. 2025], clinical data transformation [Conrado et al. 2025].Despite these advances, existing approaches still face challenges related to semantic persistence, domain dependency, and governance. To address these limitations, this study proposes a metadata-driven, LLM-based multi-agent framework for publishing RDF Views into EKGs.

Studies such as [Hofer et al. 2024] demonstrated that, in zero-shot settings, commercial LLMs often produce invalid RML rules, confuse semantically close properties, and fail to identify correct data iterators. Unlike approaches based mainly on automated repair, this study addresses these issues through the incorporation of HITL validation before triplification. [Alzahrani and O'Sullivan 2025] showed that contextual metadata can improve the precision and validity of LLM-generated RML artifacts. However, their metadata is mainly used as static prompt context. In contrast, this study treats metadata as an active architectural component through a dedicated agent connected to a semantic metadata graph based on VoSV and the DDP_SV design pattern supporting a RAG process and providing continuous governance and semantic context.

5. Final Considerations

This article presented a multi-agent framework based on LLMs to support RDF view publishing in EKGs. The proposed publishing agents demonstrated effectiveness in automating key tasks involved in the construction of RDF views, including schema extraction, ontology generation, semantic mapping generation, and RDF materialization. By anchoring agent workflows to a formal semantic metadata graph, the framework ensures structured execution, transparency, and effective *human-in-the-loop* interaction, allowing automate and optimize the publication of diverse data sources into an EKG. The current study is limited by its evaluation on a single data source and the absence of comprehensive metrics to assess the quality of the generated artifacts. Future work will address these limitations by defining evaluation metrics, validating the framework across multiple datasets, and comparing the performance of different LLM models.

References

Alzahrani, S. and O'Sullivan, D. (2025). Guiding llm generated mappings with lifecycle-based metadata: An early evaluation.

Arenas-Guerrero, J., Chaves-Fraga, D., Toledo, J., Pérez, M. S., and Corcho, O. (2024). Morph-KGC: Scalable knowledge graph materialization with mapping partitions. *Semantic Web*, 15(1):1–20.

Conrado, R. C., Campos, E. M., Traina Jr, C., Traina, A. J., and Cazzolato, M. T. (2025). From tables to graphs with clinicoatlas: Leveraging llms to support modeling and mining knowledge graphs. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 64–69. SBC.

da Silva, G. M., de Armas, E. M., Saieg, P., de Castro, R. A., Coqueiro, T. D., Lemos, M., and Castro, L. C. (2025). Llms na geração de mapeamentos semânticos: Experimento no domínio de plantas industriais. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 449–458. SBC.

Hofer, M., Frey, J., and Rahm, E. (2024). Towards self-configuring knowledge graph construction pipelines using llms-a case study with rml. In *Fifth International Workshop on Knowledge Graph Construction@ ESWC2024*.

Kommineni, V. K., König-Ries, B., and Samuel, S. (2024). From human experts to machines: An llm supported approach to ontology and knowledge graph construction. *arXiv preprint arXiv:2403.08345*.

Mavridis, A., Tegos, S., Anastasiou, C., Papoutsoglou, M., and Meditskos, G. (2025). Large language models for intelligent rdf knowledge graph construction: results from medical ontology mapping. *Frontiers in Artificial Intelligence*, 8:1546179.

Vidal, V., Freitas, R., Arruda, N., Casanova, M. A., and Renso, C. (2024). A data design pattern for building and exploring semantic views of enterprise knowledge graphs. In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 1–13, Porto Alegre, RS, Brasil. SBC.