Metadata-Driven Construction of Semantic Views in Enterprise Knowledge Graphs with LLM Agents

Tulio Vidal Rolim[1,2], José Renato S. Freitas [1,3], Vânia Maria Ponte Vidal[1]

[1]Universidade Federal do Ceará (UFC)
Mestrado e Doutorado em Ciência da Computação (MDCC),

[2]Instituto Federal de Educação, Ciência e Tecnologia Piauí (IFPI)

[3]Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

{tulio.xcrtf,jrenatosfreitas}@gmail.com, vvidal@lia.ufc.br

Abstract. *An Enterprise Knowledge Graph (EKG) provides a robust foundation for knowledge management, data integration, and advanced analytics within organizations by offering a semantic view that unifies and semantically integrates heterogeneous data sources from the data lake. The data integration process remains complex and time-consuming due to schema mismatches, divergent terminologies, and inconsistencies in data collection practices. Recent advances in large language models (LLMs) have shown promise in addressing these challenges by assisting with various data integration tasks. This paper introduces a modular, agent-oriented architecture that supports the incremental and interactive construction of semantic views for EKGs. The architecture leverages LLM-powered agents in conjunction with a metadata graph that captures rich contextual information about each semantic view. This metadata graph plays a central role in enabling automation, enhancing explainability, and ensuring reusability throughout the construction process. By forming agent decisions in structured and trustworthy metadata, the proposed framework facilitates the development of semantic views of enterprise knowledge graphs.*

1. Introduction

An Enterprise Knowledge Graph (EKG) provides a robust foundation for knowledge management, data integration, and advanced analytics within organizations by offering a semantic view that integrates diverse data sources from the organization's data lake.

The primary goal of the semantic view in an EKG is to establish a unified ontological framework resulting from the semantic integration of heterogeneous data sources [Galkin et al. 2017]. This integration creates a comprehensive and coherent organizational data environment, enabling seamless access and supporting informed decision-making.

Constructing and maintaining a semantic view in an EKG system is a complex and time-consuming task due to schema mismatches, divergent terminologies, and inconsistencies in data collection practices. Recent research has demonstrated the potential of LLMs in supporting data integration tasks such as column type annotation, schema matching, and entity linkage [Feuer et al. 2023, Tu et al. 2023, Kayali et al. 2023, Liu et al. 2024]. LLMs can also answer questions about data terminology, generate transformation scripts, and assist in metadata enrichment—often without requiring task-specific training data. These capabilities open new avenues for enhancing the automation and flexibility of data integration processes.

This paper introduces a modular architecture based on LLM-driven agents that supports the incremental and interactive construction of semantic views for EKG, following the data design pattern *DDP_SV* proposed in [Vidal et al. 2024]. The *DDP_SV* pattern is tailored for the logical organization of data within the semantic view of an EKG, structuring data and metadata into four hierarchical layers. This structured approach addresses key challenges in semantic data integration, while also simplifying maintenance and enhancing the flexibility and depth of semantic view exploration across various contexts.

The *DDP_SV* framework relies on *VoSV* (Vocabulary of Semantic View). to annotate the metadata associated with semantic views. The resulting *VoSV*-based metadata graph forms a core component of the EKG's semantic layer, offering detailed, machine-readable descriptions of the structure, provenance, and quality of semantic view data. Through rich semantic annotations, *VoSV* enables critical data governance capabilities such as lineage tracking, quality assessment, and enhanced data usability.

Beyond documentation, the metadata graph modeled with *VoSV* provides a robust foundation for constructing agentic systems capable of automating the synthesis of data integration pipelines. By capturing comprehensive metadata—including input sources, transformation logic, provenance chains, and quality indicators—the *VoSV*-based graph serves as a structured and reliable context provider for LLM agents. This enables more accurate reasoning, improves explainability, and ensures principled governance throughout the integration process. We argue that metadata takes on an even more central role in the era of generative AI. As LLMs increasingly guide autonomous behaviors and decision-making, grounding their actions in well-defined, traceable, and explainable metadata is essential for maintaining control, fostering trust, and ensuring accountability within digital ecosystems.

The rest of the paper is organized as follows: Section 2 presents the *DDP_SV* framework. Section 3 introduces a modular, agent-based architecture designed to support the construction of semantic views in alignment with the *DDP_SV* framework. Section 4 discusses a RAG-based execution pipeline integrated into the agent architecture, enabling agents to perform tasks by combining metadata-driven reasoning with user feedback. Section 5 discusses related work. Finally, Section 6 outlines the main conclusions and directions for future work.

2. Constructing Semantic Views with *DDP_SV* Framework

This section begins presenting a data design pattern—referred to as *DDP_SV*—specifically developed to provide a logical organization of both data and metadata within the semantic view of an EKG. In addition, Section 2.2 introduces *VoSV*, a domain-independent vocabulary designed to capture and represent the metadata of semantic views constructed using the *DDP_SV*.

2.1. Four Layer Architecture for Constructing EKG's Semantic Views

The work in [Vidal et al. 2024] introduced a data design pattern, referred to as *DDP_SV*, specifically developed to provide a logical organization of both data and metadata within the semantic view of an EKG. As illustrated in Figure 1, the semantic view in this architecture consists of two interconnected knowledge graphs: a data graph that represents the integrated content, and a metadata graph that captures the structure, provenance, and quality associated with the view.

The data graph in Figure 1(a) represents the actual integrated data within the semantic view and works as the informational backbone of the EKG. It is structured into a four-level hierarchical architecture, each level encapsulating a distinct stage of semantic integration:

- **Exported Views Layer** – Contains RDF views generated by mapping raw data sources to a shared vocabulary defined in the Semantic View Ontology (SVO), ensuring semantic consistency;

- **Linkset Views Layer** – Establishes identity links (e.g., owl:sameAs) between equivalent entities across exported views, enabling semantic alignment and cross-referencing;
- **Unification Views Layer** – Integrates semantically equivalent entities into canonical representations, consolidating references to the same real-world objects;
- **Fusion Views Layer** – Resolves conflicts among unified entities, applying resolution strategies to produce a consistent, enriched, and trustworthy view.

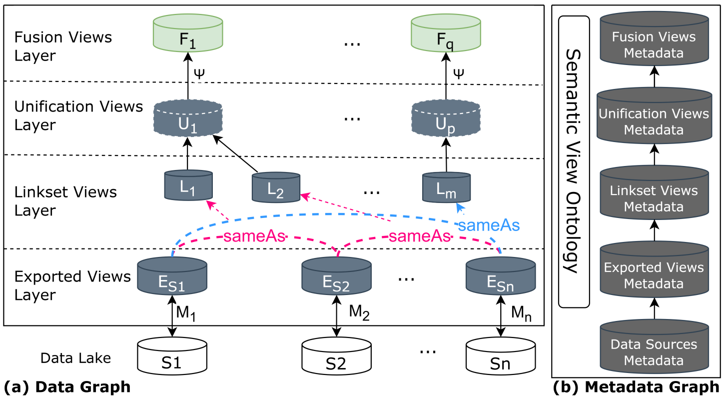

Figure 1. Data Design Pattern for EKG's Semantic Views.

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.

2.2. Modeling Semantic View Metadata with *VoSV*

To structure and model the metadata graph, is proposed *VoSV* (Vocabulary of Semantic View)—a domain-independent vocabulary that captures the metadata of semantic views built with the *DDP_SV* data design pattern.

Figure 2 offers an overview of *VoSV*, highlighting its key components and the semantic relationships among them. At the heart of the model lies the class *vosv:SemanticView*, which conceptually aggregates all core elements of a semantic view.

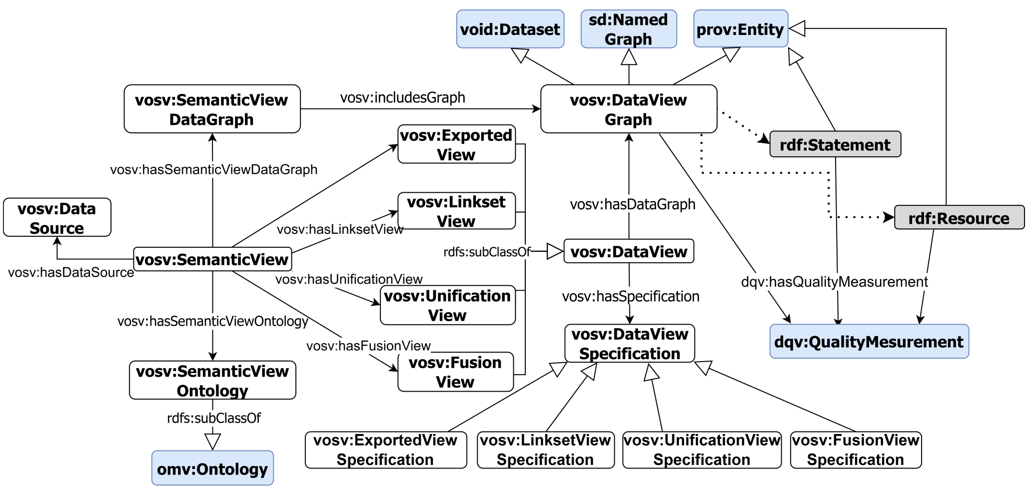

Figure 2. Vocabulary of Semantic Views (*VoSV*).

Building on this vocabulary, is proposed a process to construct both the data graph and its metadata graph within the *DDP_SV* framework. This process contains five ordered steps:

1. Semantic View Ontology Modeling – Defining a shared vocabulary to guide integration;
2. Exported View Construction – Mapping raw data sources to the ontology;
3. Linkset View Construction – Creating identity links across exported views;
4. Unification View Construction – Merges equivalent entities into canonical representations;
5. Fusion View Construction – Resolving conflicts to produce a consistent, enriched view.

Together, *VoSV* and this step-wise methodology provide a structured, semantically rich foundation for building and maintaining high-quality EKGs.

3. A Multi-Agent based Architecture for Interactive LLM-based Semantic View Construction with *DDP_SV*

This section presents a modular, agent-based architecture that supports the construction of EKG semantic views following the *DDP_SV* design pattern. The architecture enables users to incrementally and interactively build semantic views with the assistance of LLM-powered agents, leveraging a *VoSV*-based metadata graph to promote automation, explainability, and reusability. Figure 3 illustrates the main components of the architecture, which are described below.

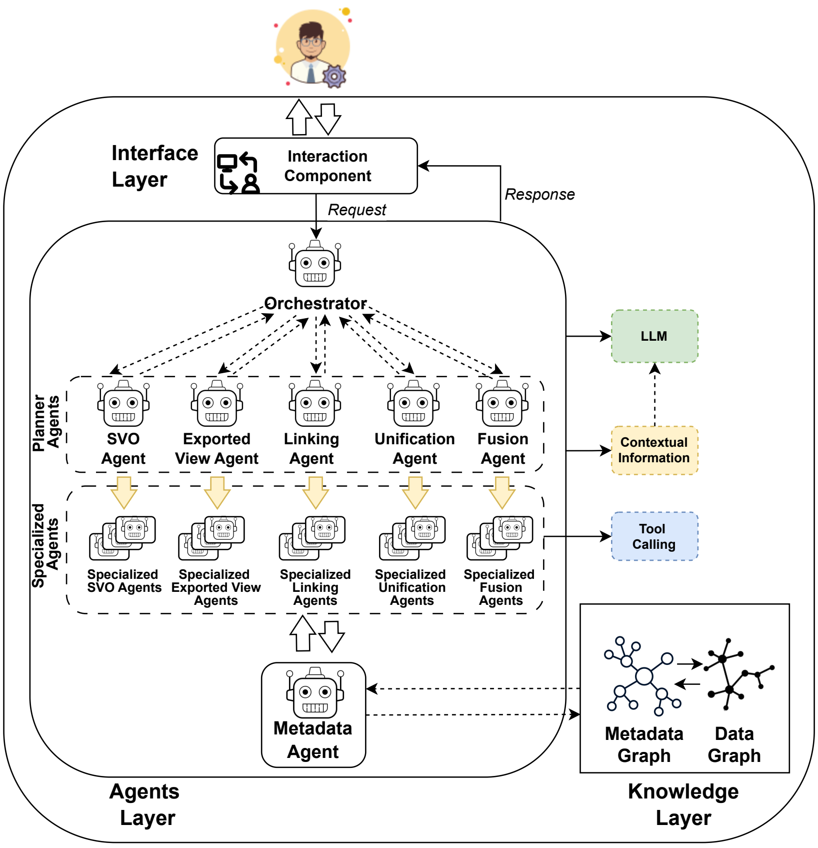

Figure 3. Multi-Agent Architecture for Interactive LLM-based Semantic View Construction.

(i) Interface Layer: Focusing on capturing user goals and mediating communication with agents, the *Interaction Component* – receives instructions, typically in natural language, expressed by human-users (data architects or knowledge analysts). These are parsed by an Orchestrator Agent, which uses an LLM to identify the user's objectives and decompose them into high-level semantic tasks to the planner agents. This component realizes the validation and routing. Firstly, the semantic plan is generated and presented to the user for review. Upon validation, it is dispatched to the corresponding specialized agents.

(ii) Knowledge Layer: At the core of this architecture, the metadata graph, built upon *VoSV*, forms a key component of the semantic layer of the EKG. It provides detailed, machine-readable descriptions of the structure, provenance, and quality of the semantic view data. Through rich semantic annotations, the *VoSV*-based metadata graph provides specialized agents with a structured, reliable foundation that supports the automated synthesis of data integration pipelines. By formalizing semantic view specifications and exposing them through metadata, *VoSV* enables the development of agentic systems capable of dynamically constructing, validating, and managing semantic views in complex enterprise environments.

The metadata graph is connected to the data graph, which can be used to enhance and extend metadata specifications. Metadata graph can be queried and updated at every stage, serving as a source of traceability, knowledge reuse, future validation, and integration evolution.

(iii) Contextual Information: In the proposed architecture, the contextual workspace functions as the internal memory layer that maintains dynamic, short-term information relevant to the ongoing construction of semantic views. It includes user-defined goals, task planning states, agent decisions, and interaction history, providing a coherent context for reasoning and coordination among agents.

(iv) LLM: Within the multi-agent architecture, the LLM acts as a textual inference engine, accessed on demand by agents. The LLM is triggered in a controlled manner by the agents, who construct the context and prompts necessary for its execution. In this way, the main responsibilities of the LLM are:

- Receives prompts dynamically generated by the agents based on the context, rules, and specific metadata;
- Generates structured textual responses, suggestions, or candidate decisions (e.g., mappings, suggested classes, validations).

(v) Tool-Calling: Tool Calling supports the construction of the semantic views by allowing agents to delegate the execution of technical tasks to external tools (APIs, services, libraries) or programmatic functions (algorithms, heuristics) — with greater precision and reliability.

Each stage can be assisted by reusable tools called on demand. For example, to build the SVO, Ontop [Xiao et al. 2020] (bootstrapping) can be used; for mappings – pyRML [Nuzzolese 2025], in the same way for subsequent stages.

(vi) Agent Layer: The proposed approach adopts a hierarchical multi-agent architecture designed to support the incremental, modular, and explainable construction of semantic views within an EKG. The architecture enforces a step-by-step workflow, where each phase is executed in a controlled, traceable, and transparent manner, with explicit human validation at every decision point.

At the top of the hierarchy is the Orchestrator Agent, responsible for orchestrating the overall process. It interprets the user's high-level goals and delegates them to the appropriate

Planner Agent. There are five Planner Agents, each corresponding to one step of the *DDP_SV* framework: (i) Semantic View Ontology Modeling Agent, (ii) Exported View Construction Agent, (iii) Linkage View Construction Agent, (iv) Unification View Construction Agent, and (iv) Fusion View Construction Agent.

Each Planner Agent supervises a specific phase of the *DDP_SV* process and is responsible for decomposing its assigned goal into a sequence of fine-grained subtasks. These subtasks are then assigned to one or more Specialized Agents, each customized to perform a particular operation, such as schema analysis, mapping generation, alignment validation, link discovery, or fusion logic application.

To support metadata handling across all phases, the architecture includes a dedicated Metadata Agent that acts as a centralized support component. This agent provides the following key services:

- Respond to metadata queries from specialized agents, offering filtered, contextual metadata tailored to the current task;
- Ensure consistency and traceability by maintaining the metadata graph in alignment with the *DDP_SV* framework and the *VoSV* vocabulary;
- Record the outcomes of tasks performed by specialized agents;
- Facilitate knowledge reuse, enabling agents to leverage decisions made previously.

By abstracting metadata access and update operations from task execution logic, the Metadata Agent enables a clean separation of concerns, ensuring that Specialized Agents remain focused on their semantic reasoning tasks while maintaining robust, consistent, and traceable metadata management throughout the semantic view construction process. Each specialized agent is equipped with a structured and conversational RAG (*Retrieval-Augmented Generation*) pipeline. In this setup:

- Agents retrieve contextual metadata exclusively through the Metadata Agent;
- Construct LLM prompts based on the retrieved metadata, and;
- Use the LLM to generate task-specific proposals or decisions (e.g., class suggestions, mappings, fusion strategies).

As part of the interactive design, every specialized agent enters a dialogue loop with the user upon completing a subtask. The user is invited to review and validate the intermediate results, with the ability to approve, modify, or reject the output. Only after explicit user confirmation does the system commit the result to the metadata graph and proceed to the next step in the construction workflow.

In the proposed architecture, inference activities are not centralized in a monolithic reasoning component, but instead are delegated to specialized agents and the metadata agent. Each specialized agent is responsible for applying domain-specific heuristics, rules, or logic within the scope of its task—for example, suggesting schema alignments, resolving data conflicts, or assessing quality constraints.

The metadata agent plays a complementary role by reasoning over the structured metadata graph, obtaining contextual insights such as semantic consistency, reuse of existing mappings, or applicability of transformation patterns. This distributed inference strategy enhances modularity, enables agent-specific explanations, and supports incremental, context-aware decision-making throughout the semantic view construction process.

4. RAG-based Pipeline Combining Metadata-driven Reasoning with User Feedback

In the proposed framework, each Specialized Agent is equipped with its own Retrieval-Augmented Generation (RAG) pipeline. This pipeline is responsible for contextualizing and executing subtasks by integrating metadata and, when necessary, human feedback. It interacts with the Metadata Agent to retrieve and update relevant metadata and may engage in user interaction loops to resolve uncertainties or validate intermediate results. Figure 4 illustrates the RAG pipeline workflow, detailing its interaction with the Metadata Agent and the user, as outlined in the following steps.

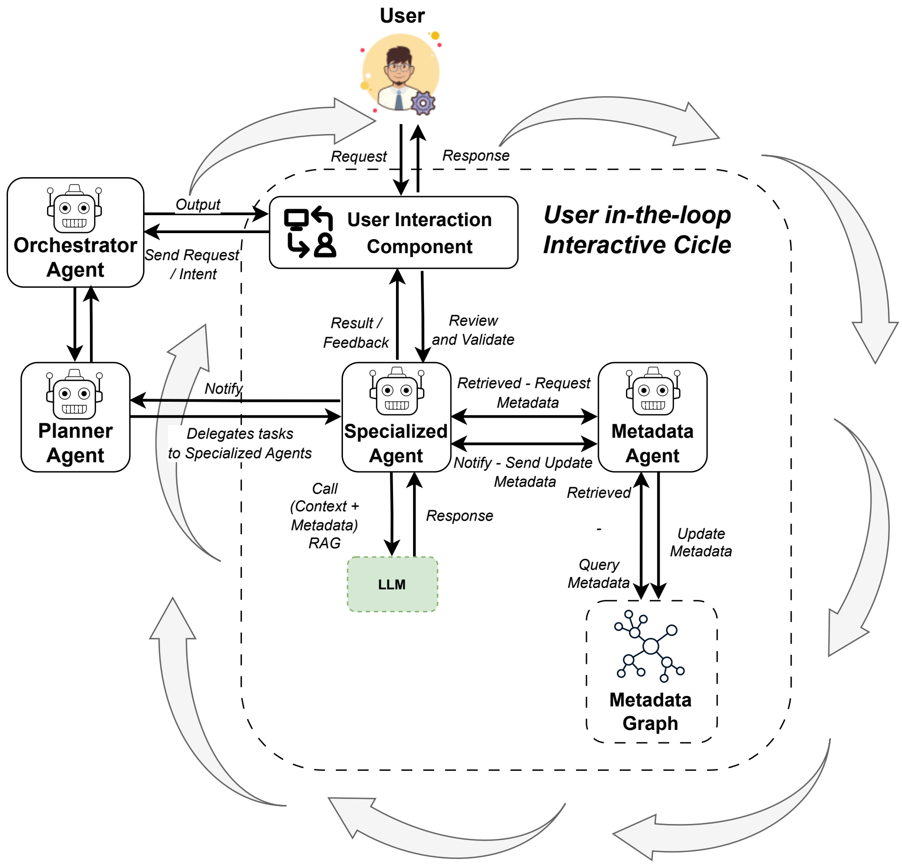

Figure 4. RAG-Driven Workflow for Task Execution.

1. The Planner Agent delegates a subtask to a Specialized Agent;
2. The Specialized Agent activates a RAG pipeline, which performs the following actions:
    - Queries the Metadata Agent to retrieve task-relevant metadata (e.g., previous mappings, ontology fragments, data view specifications);
    - Constructs a prompt using the retrieved metadata and invokes a LLM to generate outputs as (e.g., match suggestion, mapping recommendation, fusion strategy).
  - If the generated output is uncertain, incomplete, or ambiguous:
    - The agent initiates a turn-based interaction with the user to request clarification or validation;
    - The user provides feedback, which is used to update the RAG context;
    - The generation step is re-executed, now incorporating the user's input.
3. Once a final result is obtained;
    - Is submitted to the Metadata Agent to be persisted as part of the metadata graph;
    - The Specialized Agent notifies the Planner Agent that the task has been successfully completed.

4.1. Case Study

To demonstrate this approach is proposed a case study to construct a semantic view - especifically to linkset views (LV_MusicArtists) that connects equivalent musical artists across MusicBrainz and DBpedia data surces. To achieve this, a specialized multi-agent pipeline is adopted. As a premise, the successful construction of the exported views to Music Brainz (EV_MusicBrainz) and to DBpedia (EV_DBpedia) will be considered.

The Linkset View construction process is driven by a coordinated set of agents that operate incrementally and consult the Metadata Graph (modeled with VoSV) to to provide context, structure and semantic knowledge, benefiting from an organized memory so that the agent can reason better, act more accurately, generate more useful outputs and guide future decisions. RAG is employed to access and contextualize relevant metadata previously registered during the Exported View construction phase to the LLM.

Phase 1: User Goal Initialization

Human Prompt:

```
"I want create a linkset of musical artists from MusicBrainz and DBpedia."
```

The Orchestrator Agent interprets the user's intent and routes the request to Linkset View Planner Agent, responsible for coordinating the construction of the Linkset View.

Phase 2: Task Planning

The Linkset View Planner Agent defines task planning and delegates to specialized agents.

```
[
  { "step": 1, "agent": "Match Agent" }, { "step": 2, "agent": "LinksetViewGeneration Agent" }
]
```

Phase 3: Match Identification

Initially, the Match Agent call the Metadata Agent to query the Metadata Graph to retrieve the Classes involved from SVO and the Exported Views.

Match Agent Prompt:

```
"Retrieve from Metadata Graph using the
vocabulary in :SemanticViewOntology
classes of music artist and your superclass
related in common by rdfs:subClassOf."
```

Metadata Agent Retrieve:

```
Response:

{
"source_ev": "EV_MusicBrainz",
"target_ev": "EV_DBPedia",
"source_class": "svm:MusicArtist_MusicBrainz",
"target_class": "svm:MusicArtist_DBpedia",
"generalization_class: svm:MusicArtist".
}
```

From the result of the Metadata Agent, the Match Agent can now identify the svm:MusicArtist match class that should be used to specify the linkset view. This structured information constitutes the semantic basis for the next step, where the agent will use the LLM to determine the appropriate Match Property class to be used in the link between instances.

Then, for the Match Agent to determine the equivalent property used to identify in both instances (match property), it must analyze the mappings of each exported view (EV_MusicBrainz, EV_DBPedia) stored in the metadata graph to find the corresponding property used to define the URI templates. To do this, the Match Agent performs a request prompt such as:

Match Agent Prompt (to LLM):

```
"Extract URI templates and subject
mappings from each EV. Identify
which field was used to compose
the URI."
```

Metadata Agent Retrieve:

```
Response:
{
"source_field": "mbid",
"target_field": "id"
}
```

After identifying the (source and target) fields used in the uri templates of the exported views mappings, the Match Agent is executed with enriched context to suggest the appropriate match property.

Match Agent Prompt (to LLM):

```
You are a knowledge graph expert agent specialized in analyzing URI construction metadata
to suggest equivalent linking properties

Context:
- Source Exported View: EV_MusicBrainz
- Target Exported View: EV_DBPedia
- Match Class : svm:MusicArtist
- Source URI uses field: mbid
- Target URI uses field: id

Rule:
If two classes have different properties mapped field/columns as key in a subject
identifiers (rr:template), but both represent semantically the same concept, suggest a
common property as **match property**.

Example:
- (Student Mapping):
rr:template "http://.../student/{s_id}", rr:predicate :student_id → :s_id
- (Person Mapping):
rr:template "http://.../person/{p_id}", rr:predicate :person_id → p_id
→ Suggestion: use `id` as generalized match property.


Instruction:
Consult the metadata graph to determine which properties were used to build the subject
URI templates for each class, using rr:template and rr:predicate.
```

The Match Agent presents the recommended match property and the user is then asked to validate. Below is the output containing the validation prompt, alongside the user's validation choice.

Match Agent Prompt:

```
"Use id as the match property
(based in equivalence of mbid and id)
as linking keys?"
```

Human Prompt:

```
"Yes, use."
```

Phase 5: Linkset View Generation

The Linkset View Generation Agent presents a sample of the generated linkset view and queries the user for validation. Therefore, the Metadata Agent writes the linkset view LV_MusicArtists to the metadata graph.

Match Agent Prompt:

```
You agree with the generated linkset view?
:LV_MusicArtists a vosv:LinksetView ;
  vosv:linksBetween :EV_MusicBrainz, :EV_DBPedia ;
  vosv:hasMatchClass svm:MusicArtist ;
  vosv:hasMatchProperty svo:id ;
  vosv:usesPredicate owl:sameAs .
```

Human Prompt:

```
"Yes, I agree."
```

5. Related Work

The construction of EKGs has gained significant attention as a means of semantically integrating heterogeneous organizational data to support knowledge management, data analytics, and decision-making. Despite advancements in knowledge graph engineering, traditional approaches continue to rely heavily on manual work and expert knowledge, making the process labor-intensive and difficult to scale. Recent studies have begun to explore how LLMs can assist in automating EKG semantic view construction, yet several limitations persist.

[Laurenzi et al. 2024] propose a six-step design-science-driven methodology for EKG development supported by LLMs. Their approach uses LLMs in discrete tasks such as competency question formulation, ontology construction, and knowledge integration. However, their process remains predominantly linear and human-supervised, lacking mechanisms for agent collaboration or dynamic contextual awareness during graph development. In parallel, LLM-based Multi-Agent Systems (LMAs) have emerged as a powerful paradigm in domains such as software engineering. [He et al. 2024] demonstrate that multi-agent collaboration can enhance autonomy, robustness, and scalability in complex workflows by distributing tasks among specialized agents orchestrated through shared goals and communication protocols. While their work is primarily situated in software engineering, it lays the foundation for extending such collaborative agent architectures to the enterprise knowledge graphs and semantic integration.

[Zhu et al. 2024] propose AutoKG, a multi-agent system in which LLMs play distinct roles to collaboratively construct and reason over knowledge graphs. While promising, AutoKG does not explicitly model to representation of semantic views nor integrate a metadata governance mechanism. Similar in vision to Harmonia [Santos et al. 2025], our work also leverages LLM-driven agents in integration tasks. However, while that approach focuses on harmonizing tabular data via prompt-based workflows, our architecture targets the semantic structuring of EKGs using a data design pattern (*DDP_SV*), semantic metadata (*VoSV*), and multi-agent coordination to support long-term governance and semantic consistency.

6. Conclusions and Future Works

This paper presented a modular, LLM-driven multi-agent architecture designed to support the incremental and interactive construction of semantic views in EKGs, following the *DDP_SV* design pattern. At the core of the architecture is a *VoSV*-based metadata graph, which provides structured, machine-readable context to enhance automation, explainability, and governance throughout the integration process. The *DDP_SV* framework promotes a "pay-as-you-go" construction model, making it well-suited for dynamic and evolving data environments.

This proposal is grounded in the observation that existing systems offer limited support for the incremental and interactive construction of semantic views—particularly when it comes to leveraging LLMs and agent-based collaboration. This gap is precisely what this work aims to address. By equipping each specialized agent with a RAG pipeline, the architecture enables metadata-aware reasoning and incorporates user feedback when necessary, resulting in more accurate, explainable, and trustworthy task execution.

While the architecture proposed here is conceptual and yet to be fully implemented, we recognize the technical challenges ahead. Nonetheless, we believe that the articulation between the *DDP_SV* framework, specialized LLM agents, and structured metadata offers a promising direction for addressing the complexity of semantic view construction in EKGs. We view this workshop as an ideal venue to refine and advance this proposal through community feedback and discussion.

References

Feuer, B., Liu, Y., Hegde, C., and Freire, J. (2023). Archetype: A novel framework for open-source column type annotation using large language models. *arXiv preprint arXiv:2310.18208*.

Galkin, M., Auer, S., Vidal, M.-E., and Scerri, S. (2017). 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, pages 88–98. SCITEPRESS.

He, J., Treude, C., and Lo, D. (2024). Llm-based multi-agent systems for software engineering: Literature review, vision and the road ahead. *Proceedings of the ACM (forthcoming)*. Preprint available via arXiv.

Kayali, M., Lykov, A., Fountalis, I., Vasiloglou, N., Olteanu, D., and Suciu, D. (2023). Chorus: foundation models for unified data discovery and exploration. *arXiv preprint arXiv:2306.09610*.

Laurenzi, E., Mathys, A., and Martin, A. (2024). An llm-aided enterprise knowledge graph (ekg) engineering process. In *AAAI Spring Symposium Series (SSS-24)*. Association for the Advancement of Artificial Intelligence.

Liu, Y., Pena, E., Santos, A., Wu, E., and Freire, J. (2024). Magneto: Combining small and large language models for schema matching. *arXiv preprint arXiv:2412.08194*.

Nuzzolese, A. G. (2025). Streamlining knowledge graph creation with pyrml. *arXiv preprint arXiv:2505.20949*.

Santos, A., Pena, E. H., Lopez, R., and Freire, J. (2025). Interactive data harmonization with llm agents. *arXiv preprint arXiv:2502.07132*.

Tu, J., Fan, J., Tang, N., Wang, P., Li, G., Du, X., Jia, X., and Gao, S. (2023). Unicorn: A unified multi-tasking model for supporting matching tasks in data integration. *Proceedings of the ACM on Management of Data*, 1(1):1–26.

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.

Xiao, G., Lanti, D., Kontchakov, R., Komla-Ebri, S., Güzel-Kalaycı, E., Ding, L., Corman, J., Cogrel, B., Calvanese, D., and Botoeva, E. (2020). The virtual knowledge graph system ontop. In *International Semantic Web Conference*, pages 259–277. Springer.

Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., and Zhang, N. (2024). Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv preprint arXiv:2305.13168*.