metis: AI Agent Platform for Human-AI Interaction with Knowledge Graphs*

Linn Aung[1], Aaron Eberhart[1], Peter Haase[1,*], Nicolas Heist[1,*], Liubov Kovriguina[1,*], David Lamprecht[1] and Nazanin Mashhaditafreshi[1]

[1]*metaphacts GmbH, Wieslocher Str. 38, 69190 Walldorf, Germany*

Abstract

We present metis, an AI agent platform that assists users with semantic modeling, search, and discovery across knowledge graphs. It combines the conversational capabilities of Large Language Models with the precision of semantic knowledge graphs, integrating seamlessly with the functionalities of the metaphactory platform, such as semantic search and visualization. metis delivers AI agents that provide generative power, semantic precision & contextual, explainable insights.

Keywords

metis and metaphactory, AI agents for semantic modeling, agentic interfaces to knowledge graphs

1. Introduction

The integration of Large Language Models (LLMs) and Knowledge Graphs (KGs) has shown significant results across a variety of individual tasks, in particular, SPARQL query generation [1, 2], named entity recognition and linking [3], knowledge graph construction and validation [4], retrieval-augmented generation [5], and ontology modeling and evaluation [6]. However, moving from the successful execution of isolated tasks to creating a holistic agent that can manage complex processes and deliver tangible business value remains a substantial challenge [7]. This evolution requires a combination of generative AI capabilities with symbolic and procedural constraints, carefully balancing conversational interaction, data retrieval, and visualization to truly address user needs.

In this paper, we present metis, an AI agent platform built on a foundation of knowledge-driven, explainable AI. metis offers businesses a trusted, context-aware platform that transforms enterprise data into actionable insights. At its core, a semantic model acts as a trust and context layer, guiding every AI interaction to ensure that responses are not only relevant but also explainable and aligned with enterprise semantics. By integrating the conversational power of LLMs with the precision of semantic KGs and the rich features of the metaphactory platform [8], metis empowers enterprises to build and deploy AI agents that provide intelligent, reliable support for a variety of business operations.

2. metis Foundations and Architecture

At the heart of metis lies a semantic model, the trust and context layer, that informs and scopes every AI interaction. This model guides the agents, ensuring not only relevancy of the results but also explainability and alignment with enterprise semantics.

The generic architecture of metis [1], is structured into three main layers: the *User Interface*, the *metis agent* layer, and the underlying metaphactory knowledge graph platform. The interaction process

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

la@metaphacts.com (L. Aung); ae@metaphacts.com (A. Eberhart); ph@metaphacts.com (P. Haase); nh@metaphacts.com (N. Heist); lk@metaphacts.com (L. Kovriguina); dl@metaphacts.com (D. Lamprecht); nm@metaphacts.com (N. Mashhaditafreshi)

0009-0009-4338-0983 (L. Aung); 0000-0003-3007-5460 (A. Eberhart); 0000-0002-7561-7000 (P. Haase); 0000-0002-4354-9138 (N. Heist); 0000-0001-9962-1138 (L. Kovriguina); 0000-0002-9098-5389 (D. Lamprecht); 0009-0002-5867-8853 (N. Mashhaditafreshi)

[1]See the architecture in the platform documentation: https://help.metaphacts.com/resource/Help:AIServicesFoundations.

begins when a user sends a *Request* through the *User Interface*, which can include a message and files in natural language. This request is ingested by the *Conversational Agent* in the metis layer. The agent's *Context Manager* enriches the request by combining it with predefined instructions, dialogue history, and the relevant semantic model, which is retrieved from metaphactory. This enriched prompt is then passed to an LLM-powered *Orchestrator* that plans and executes a workflow to fulfill the request. If necessary, the *Orchestrator* performs tool calls, sending specific input parameters to one or more available tools. These tools are built on top of *metaphactory APIs* and interact with the data layer, which is managed by a Repository Manager. The tool returns a *Tool Result*, consisting of either the requested output or an error message. The *Orchestrator* may iteratively call tools until the necessary information is retrieved and uses the results for *Response Formulation*. Finally, the resulting *Response*, which can be a combination of natural language and various visualizations, such as tables or charts, is sent back to the *User Interface*.

Based on the generic architecture of the metis platform, different kind of agents can be configured. We here focus on two types of agents equipped with a set of tools to support specific tasks: a *Semantic Modeling Assistant* to collaboratively develop and refine ontologies [2], and a *Search & Discovery Agent* with access to the knowledge graph to resolve information needs of its users (see the architecture in Appendix A).

Semantic Modeling Assistant. The Semantic Modeling Assistant is designed to support users in the collaborative development and refinement of ontologies. This agent guides users through a structured modeling process based on the established Linked Open Terms (LOT) methodology[3] [9] as well as metaphacts' Semantic Modeling Guidelines[4]. Instead of a rigid implementation of LOT within the agent, metis leverages the entire metaphactory ecosystem to cover the ontology lifecycle.

The initial steps of the LOT methodology, such as *Ontology Requirements Specification*, are driven by metis. The assistant helps users formulate competency questions and define use cases, which are stored as explicit modeling artifacts in the ontology. For the *Ontology Implementation* phase, metis assists with the crucial sub-activities of reuse, conceptualization, and evaluation. It facilitates ontology reuse by treating the *metaphacts Ontology Repository*[5] as a primary resource hub. The agent distinguishes between *conceptualization* and *encoding*, presenting modeling concepts like classes and properties in a human-interpretable format while handling the formal encoding in the background. *Ontology evaluation* is supported through both formal SPARQL-based checks and by applying the LLM's reasoning capabilities to critique modeling artifacts. The subsequent steps of *ontology publication and maintenance* are handled by the robust, built-in functionalities of the metaphactory platform, such as automated documentation generation and editorial workflows. The assistant's architecture (Figure **??**) features pre-configured tools tightly integrated with metaphactory to create and persist these modeling artifacts.

Search & Discovery Agent. The Search & Discovery Agent enables users to perform intuitive exploration and targeted search across knowledge graphs using natural language. This agent is highly customizable; users can adapt its instructions, enable or disable specific tools, and tune LLM parameters to fit their domain and use case. Its architecture (cf. Figure A) gives it access to three categories of tools:

- *Structured Querying* tools perform tasks like entity lookup and linking, as well as translation of natural language to SPARQL for precise data aggregation and analysis.
- *Unstructured Retrieval* tools leverage metaphactory's retrieval functionality to find information in textual content that cannot be easily mapped to a graph pattern.
- *Visualization* tools render the retrieved data in various formats, such as tables, charts, and graphs, to present it to the user in the most appropriate way.

[2] See the architecture in the platform documentation: https://help.metaphacts.com/resource/Help:SemanticModelingAssistant.
[3] https://lot.linkeddata.es/
[4] https://metaphacts.com/images/PDFs/metaphacts_Semantic_Modeling_Guidelines_2.0.pdf
[5] https://ontologies.metaphacts.com/

Through this combination of tools, the agent supports multi-turn dialogues, allowing users to iteratively refine their information needs and build on previous results to gain deep, context-aware insights from their data.

3. Demo Scenarios for metis Agents

Demo Scenario for the Semantic Modeling Assistant. The Semantic Modeling Assistant supports creating ontologies from scratch by guiding the user through the LOT methodology or refining and evaluation existing ontologies. The following scenarios are demonstrated in our supplemental material and partially shown in Figure 1:

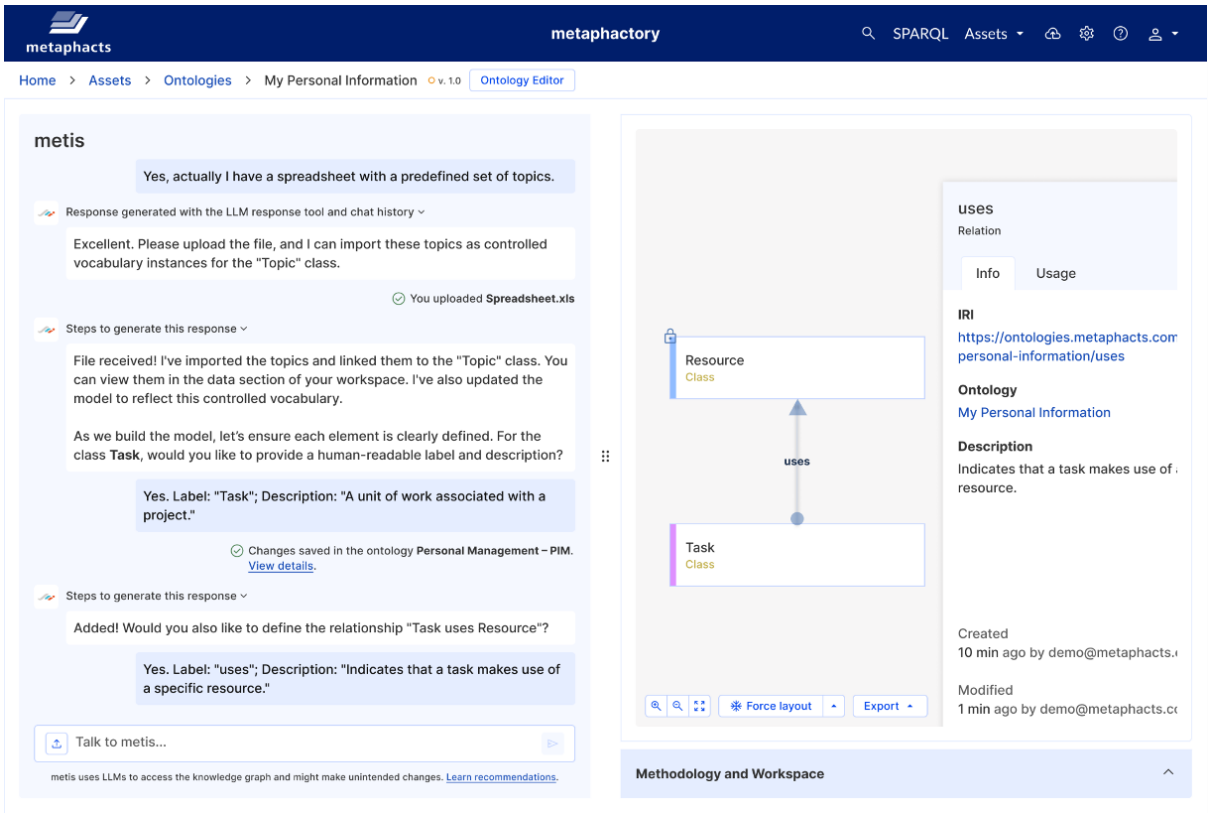

Figure 1: Semantic Modeling Assistant demo scenario creating the Personal Information ontology.

1. **From specification to implementation**. In this scenario the ontology is developed from scratch. metis guides the user through the ontology requirements specification, ontology formalization, ontology implementation, ontology evaluation, and ontology publication activities. The user can provided additional input by uploading files to support the ontology development process.
2. **Reusing ontological resources**. Ontology developers both reuse and re-engineer ontological resources. metis identifies existing ontology resources that can be reused facilitating the integration, adaptation, and extension of these resources to meet specific domain requirements.
3. **Evaluate and restructure ontological resources**. Ontology engineers systematically evaluate and restructure (e.g., modularize, prune, extend, specialize) ontological resources. metis incorporates requirements specification and built-in evaluation mechanisms to assist in assessing and refining ontological resources.

Figure 1 shows an example how metis facilitates the construction of a *Personal Information* ontology by building a controlled vocabulary for the *Topic* class, which is derived from a spreadsheet of predefined topics. During the modeling process metis ensures that each ontology element is explicitly defined

with appropriate metadata. In the example the *Task* class is enriched with a human-readable label and description. Furthermore, metis tracks semantic relations and links the *Task* class to the *Resource* class through the newly defined relation *uses*.

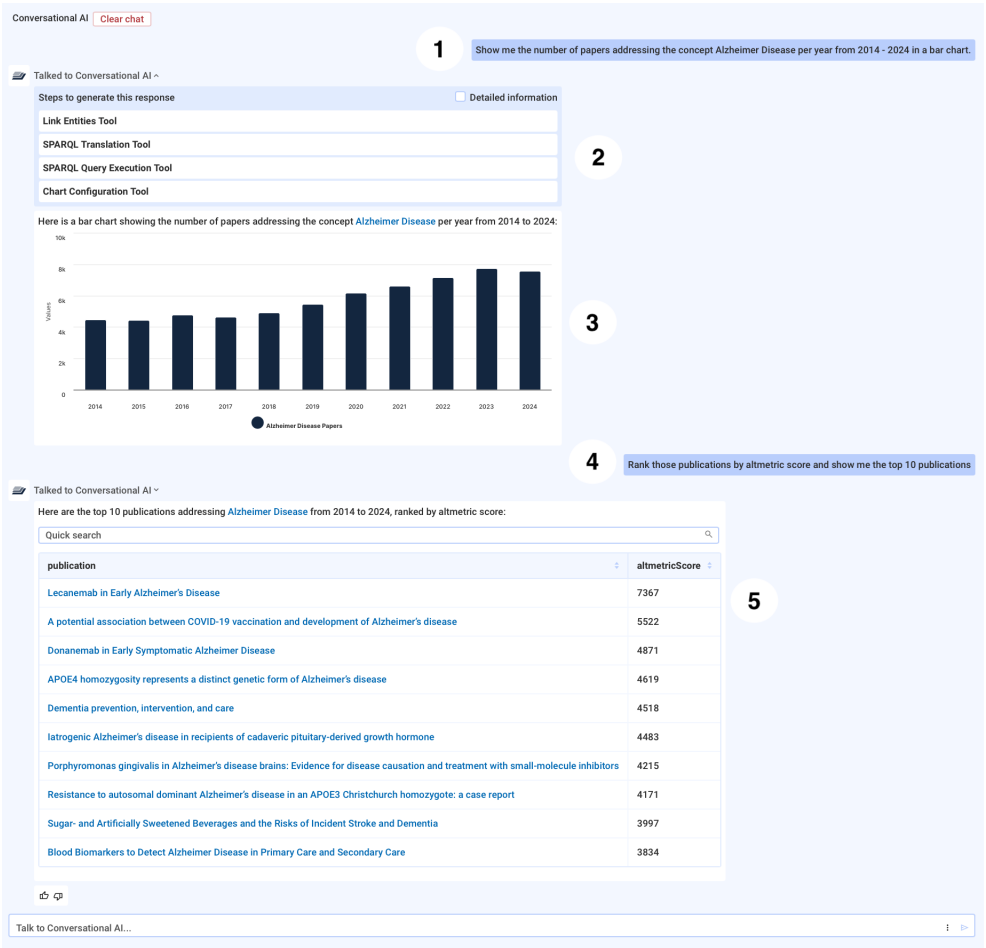

Figure 2: Search & Discovery Agent demo scenario over the Dimensions Knowledge Graph.

Demo Scenario for the Search & Discovery Agent. We have deployed the Search & Discovery Agent across several knowledge graphs, covering different domains. Notably, SemOpenAlex[6] [10] and the Dimensions Knowledge Graph [7] that are massive knowledge graphs with billion of RDF triples about scientific publications and their associated entities, such as authors, institutions, publishers, funders and sources, from all scientific domains. Additionally, the agent has been integrated with Wikidata [11], a comprehensive cross-domain knowledge graph, as well as the metaphacts Enterprise Information Architecture Knowledge Graph, further demonstrating its versatility and applicability across diverse knowledge bases. Figure 2 demonstrates how users interact with the Dimensions Knowledge Graph via the agent's natural language interface. Retrieved results are grounded in scientific metadata from the underlying database and appropriately visualized, enabling users to effectively interpret and trust the presented information.

1. **Natural language request.** The user formulates an information need to retrieve the number of publications addressing *Alzheimer Disease* from the year 2014 to 2024.
2. **Transparent Multi-step Reasoning & Tool Orchestration.** The agent transparently uses a sequence of tools to fulfill the user request. First the entity *Alzheimer Disease* is identified in the

[6] https://semopenalex.org
[7] https://metaphacts.com/solutions/products/dimensions-knowledge-graph

request and linked to its corresponding IRI in the Knowledge Graph. Then a structured SPARQL query is generated and executed to retrieve the number of relevant publications.

3. **Result visualization.** After the data is retrieved by executing the SPARQL query the agent uses its visualization capabilities to present the number of publications per year as a bar chart.
4. **Explanations.** Along with the presented results, explanations about how the result was generated are provided to the user. Upon user request, detailed information about all tools used and the entire data lineage are presented.
5. **Multi turn dialogue.** The user sends a follow-up request to rank all publications from the previous result by their *altmetric score*[8].
6. **Context-aware answer.** The agent retrieves the 10 *Alzheimer Disease* publication with the highest *altmetric score* from 2014 to 2024 and presents the results in a table for clear visualization.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] L. Kovriguina, R. Teucher, D. Radyush, D. Mouromtsev, Sparqlgen: One-shot prompt-based approach for sparql query generation., in: SEMANTiCS (Posters & Demos), 2023.

[2] J. Sequeda, D. Allemang, B. Jacob, A benchmark to understand the role of knowledge graphs on large language model's accuracy for question answering on enterprise sql databases, in: Proceedings of the 7th Joint Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), 2024, pp. 1–12.

[3] M. Freund, R. Dorsch, S. Schmid, T. Wehr, A. Harth, Enriching rdf data with llm based named entity recognition and linking on embedded natural language annotations, in: International Knowledge Graph and Semantic Web Conference, Springer, 2024, pp. 109–122.

[4] S. Tsaneva, D. Dessì, F. Osborne, M. Sabou, Knowledge graph validation by integrating llms and human-in-the-loop, Information Processing & Management 62 (2025) 104145.

[5] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, J. Larson, From local to global: A graph rag approach to query-focused summarization, arXiv preprint arXiv:2404.16130 (2024).

[6] N. Fathallah, A. Das, S. D. Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, Neon-gpt: a large language model-powered pipeline for ontology learning, in: European Semantic Web Conference, Springer, 2024, pp. 36–50.

[7] B. Zhang, V. A. Carriero, K. Schreiberhuber, S. Tsaneva, L. S. González, J. Kim, J. de Berardinis, Ontochat: a framework for conversational ontology engineering using language models, in: European Semantic Web Conference, Springer, 2024, pp. 102–121.

[8] P. Haase, D. M. Herzig, A. Kozlov, A. Nikolov, J. Trame, metaphactory: A platform for knowledge graph management, Semantic Web 10 (2019) 1109–1125. URL: https://doi.org/10.3233/SW-190360. doi:10.3233/SW-190360.

[9] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, R. García-Castro, Lot: An industrial oriented ontology engineering framework, Engineering Applications of Artificial Intelligence 111 (2022) 104755.

[10] M. Färber, D. Lamprecht, J. Krause, L. Aung, P. Haase, Semopenalex: the scientific landscape in 26 billion rdf triples, in: International Semantic Web Conference, Springer, 2023, pp. 94–112.

[11] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85.

[8] https://www.altmetric.com

A. Search & Discovery Agent Architecture

Figure A shows the architecture of the Search & Discovery Agent, which has access to three groups of tools: structured retrieval tools, unstructured retrieval tools, and visualization tools.

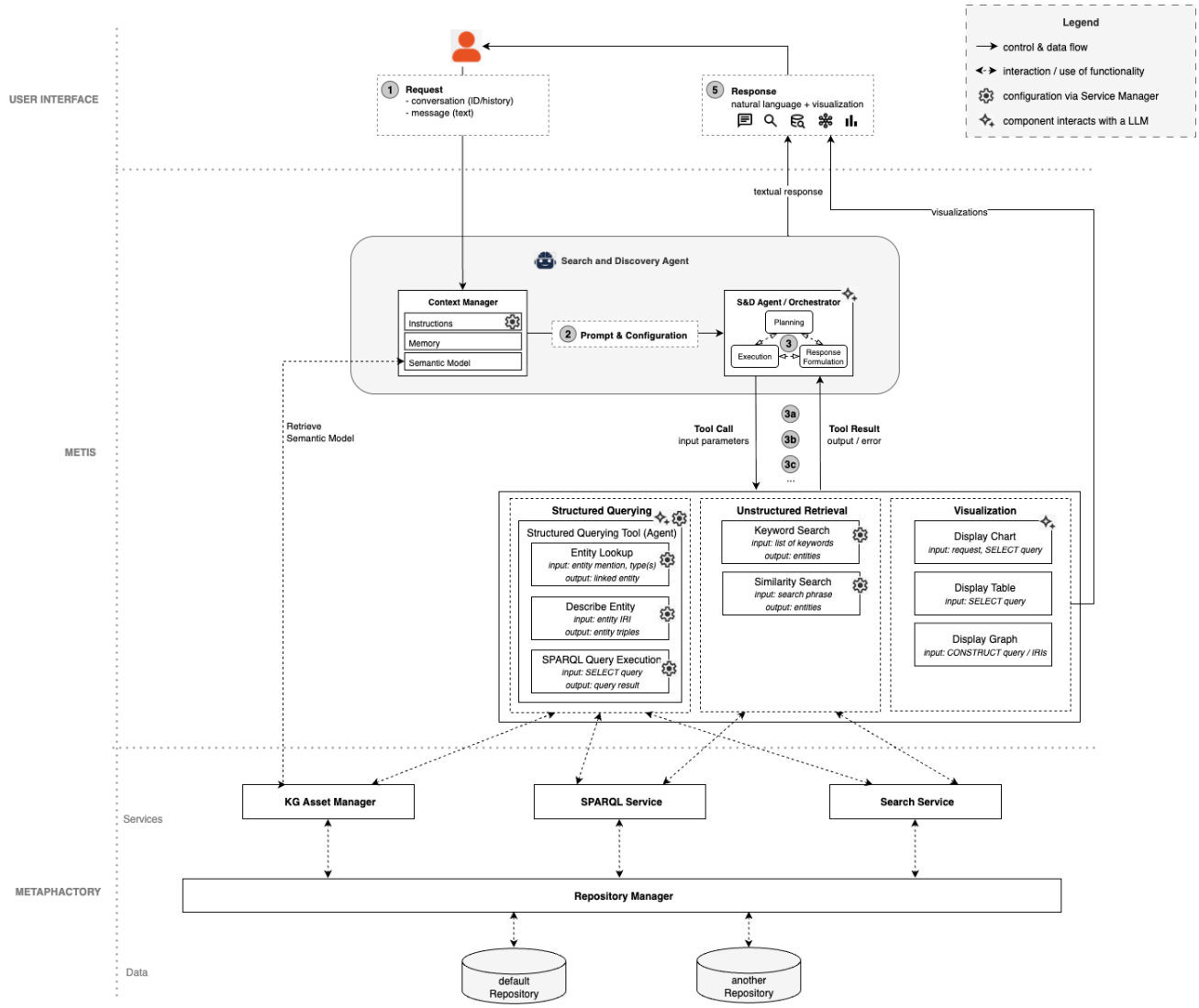

Figure A: Architecture of the Search & Discovery Agent.

B. Online Resources

Demo videos of metis are available via https://www.youtube.com/playlist?list=PLIseUuCw9-OAlmJxFDwNfmsRVPVGRHzNN. Further resources on metis are available at https://metaphacts.com/solutions/products/metis.