Ontology-Driven LLM Service Protocol for HPC Platforms: Design for Workflow Interoperability and Provenance-Aware Automation

Yejin Kwon
Korea Institute of Science and Technology Information(KISTI), Analysis Platform Team, Dept. of Supercomputing Acceleration Research
Daejeon, Republic of Korea
yejinkwon@kisti.re.kr

Jeongcheol Lee
Korea Institute of Science and Technology Information(KISTI), Analysis Platform Team, Dept. of Supercomputing Acceleration Research
Daejeon, Republic of Korea
jclee@kisti.re.kr

Young B. Park
Dankook University,
Dept. Software Engineering
Yongin-si, Republic of Korea
ybpark@dankok.ac.kr

***Abstract*— Recently, web-based High-Performance Computing (HPC) platforms have increasingly adopted Large Language Model (LLM) services to provide user-friendly and scalable access to simulation tools. In this study, we propose an ontology-based LLM service architecture that integrates with the Model Contact Protocol (MCP) to enable seamless linkage between user queries, retrieval-augmented generation (RAG), and HPC platform service tools. The ontology framework provide knowledge graph, structural consistency, semantic reasoning, and verification of generated results, thereby enhancing reliability and reusability of LLM outputs.**

Keywords—HPC, Large Language Model, Model Context Protocol, Ontology

I. INTRODUCTION

Recently, many web platform services are utilizing various LLM(Large Language Model) to provide user-friendly and scalable services. Based on the Langchain service, a basic LLM service[1][2], a MCP(Model Contact Protocol)[3][4] is defined to enable the execution of actual platform services, and services for linking services with LLM are being built. The RAG service's result data, generated based on user queries, is generated as a detailed descriptive response document. A protocol is designed to link and execute these responses with specific service tools, enabling the process of linking these responses through the MCP service platform. To ensure this service structure and integration with various service tools, the design and implementation of the MCP protocol have become core service design elements.

In this paper, we analyze ontology-based data relevance to define the linkage process with HPC service tools and proceed with the design of a service that enables simulation execution and result data analysis. This is an intermediate step in the design and implementation of an MCP protocol for simulation execution linked to an LLM service on an HPC-based simulation platform. We then analyze the data relevance based on the ontology. We then define the linkage process with HPC service tools and proceed with the design of a service that enables simulation execution and result data analysis. Existing HPC simulation platforms have services that can be mapped to service tool units, such as user authority management, simulation job execution and monitoring, and simulation software registration. To link specific user requests with service tools, we designed a protocol for analyzing user query requests and structuring the resulting data using an ontology-based GPT model. This structured data structure then matches the generated user query results with the MCP. The protocol's structure necessitates accurate entity and association data analysis. Since the actual HPC platform tools are executed based on the analyzed result data, it is crucial to execute appropriate user tools and return user results through analysis of these structured relationships and various user execution data. Therefore, rather than returning result data through existing RAG processing, we perform ontology-based data analysis to effectively adapt to execution and result analysis through structured analysis and linkage with HPC service tools based on the analysis results. While data structuring for the MCP protocol can be achieved through linkage with existing RAG services, we designed the MCP protocol to link and execute user service execution tools with enhanced accuracy by defining the precise platform structure and tool linkage process.

II. ONTOLOGY BASED LLM SERVICE

A. HPC based Web Paltform

The current HPC platform architecture is built on a variety of service tools. It supports various services for registering simulation applications in computational science and engineering and visualizing simulation results. It is built on a microservices-based architecture. Therefore, the HPC platform's service tools can operate independently, but they must also be integrated to manage simulations on HPC clusters and to execute and monitor simulations based on user queries. Figure 1 below provides a simplified diagram of the HPC web platform's services[5-7].

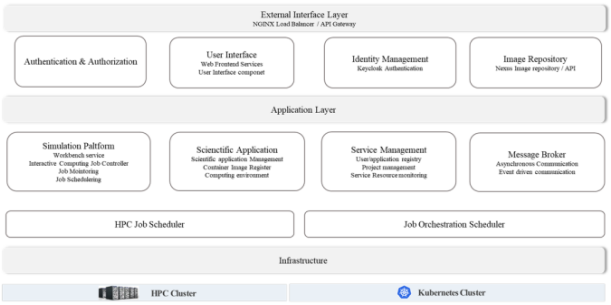

Fig. 1. Web based HPC Platform Architecture

The user interface provided is basically identical to the structure of other web platforms, and the key difference is that it provides a simulation execution environment based on Docker images, so it must be possible to share and execute images containing the user's computational science and engineering simulation code. Therefore, an image repository

exists to manage these images, and each simulation service can be built by linking the image repository with it. As shown in Figure 1, the services of the External Interface Layer are designed to be executed by linking the services of the Application Layer to the MCP, rather than linking with the LLM service.

B. *Ontology based Large Language Model*

An ontology-based LLM service is a software architecture that uses a domain ontology (OWL/RDF) and a knowledge graph (KG) as the layers for schema, facts, and rules, while employing an LLM for querying, summarization, reasoning, and extraction; RAG, tool calls, and validation (via a reasoner/SHACL) are then used to generate, constrain, and verify the LLM's outputs grounded in the ontology[8].

In other words, it can be used in conjunction with the linguistic capabilities of the LLM service to increase reliability and reusability by combining the structural verifiable knowledge of the Knowledge Graph and ontology[9-11]. If the results of the text data of the RAG service are extracted and mapped to an ontology instance, it can be generated in a specific regular result format, and a linked service can be built to map the data created by these rules to the MCP and execute it according to the platform execution process. Among the open sources that can map text to ontology instances, there is OntoGPT, and the project can create an instance object that follows the LinkML schema and the corresponding instance and perform the mapping, and can register and manage the ontology object according to a configuration file optimized for the platform, structure, and execution environment.

III. Design for HPC Platrom

Building an ontology-based LLM service requires the integration of various service servers. This process begins with a web UI interface that accepts user queries. This initial process analyzes the query and processes metadata. The query and user actions are analyzed, converted into a vector database, and retrieval is performed based on this data. Finally, a response text is generated based on the Large Language Model. This series of steps is identical to the existing RAG service process. A simplified version of the process is shown in Figure 2.

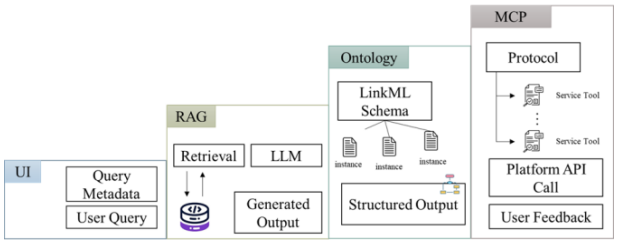

Fig. 2. Ontology based LLM Service Process

As shown in Figure 2, the entire process is designed to automatically connect from user input on the left to the right. Therefore, users enter queries for the simulation platform, much like querying existing LLM services. These queries are analyzed and the resulting text is generated. Based on this resulting text, the platform's predefined schema can be mapped to extract structured result data. The MCP server, linked to existing service tools, is responsible for analyzing each user's structured request and executing services within the platform.

A. *Platform tool-linked LLM simulation service*

Tools intended to be integrated into the HPC simulation platform with MCP can be defined as three services: simulation execution, science application registration and management, and simulation scheduler. MCP servers operate in conjunction with each service, and platform functions are defined to be executed by linking with appropriate MCP protocols using structured prompt RAG service process results data.

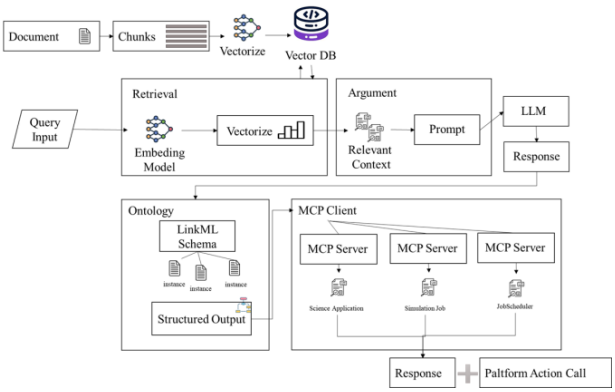

Fig. 3. Design of Ontology based LLM Simulation Platfrom

As defined in Figure 3, user input queries are vectorized to generate LLM model responses. The platform's internal linkML schema, defined as an ontology, is defined, and each data instance can be linked to the schema. The data format required for simulations used on the platform is defined, and each simulation cluster has its own specific instance type based on its execution environment. This allows for the generation of structured data by generating ontology based on user queries. The MCP architecture is a protocol defined for linking with interconnected service tools. By receiving structured data as input through the ontology process, more reliable data results can be analyzed and linked.

IV. Conclusion

Recently, various web-based platforms are offering additional LLM services using RAG. Following this trend, we analyzed various service types and designed an ontology-based LLM service to provide this service on the EDISON platform, an HPC platform currently in service. The targeted HPC platform provides specialized simulation execution environments for computational science and engineering simulations, and general users require basic knowledge to run simulations. Therefore, we analyzed the underlying technologies for building an LLM-based service that can search and execute required simulations based on user queries. We also conducted an ontology-based LLM query analysis to link it with the MCP, which can run the actual platform.

Overall, the ontology-based LLM service architecture generates text-based output data based on unstructured user input derived from the RAG service. The resulting data is mapped to an ontology schema and generated as instance-based structured data. For the MCP, which consists of an MCP server and client, the design was designed to provide structured data that can be recognized by the MCP server and integrated with actual service tools. These results may suggest the possibility of effective design of ontology-based data analysis that can be linked to specific services of the MCP-based HPC simulation platform.

ACKNOWLEDGMENT

This research was supported by the Ministry of Science and ICT through the National Research Foundation of Korea (NRF), under the Digital Convergence R&D Platform Program (No. NRF-2022M3C1A6090416), and by the Global TOP Strategic Research Group Program of the National Research Council of Science & Technology (No. GTL24031-700).

REFERENCES

[1] Xiong, Haoyi, et al. "When search engine services meet large language models: visions and challenges." IEEE Transactions on Services Computing, vol 17, pp. 4558-4577, 2024.

[2] Yang, Rui, et al. "RAGVA: Engineering retrieval augmented generation-based virtual assistants in practice." arXiv preprint arXiv:2502.14930, pp. 1-16 2025.

[3] Hou, Xinyi, et al. "Model context protocol (mcp): Landscape, security threats, and future research directions." arXiv preprint arXiv:2503.23278, pp. 1-20, 2025.

[4] Ray, Partha Pratim. "A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions." Authorea Preprints, pp. 1-44, 2025.

[5] Ma, Jin, et al. "Design and implementation of information management tools for the EDISON open platform." KSII Transactions on Internet & Information Systems, vol 11.2, 2017.

[6] Han, Sunggeun, et al. "Data Framework Design of EDISON 2.0 Digital Platform for Convergence Research." KSII Transactions on Internet & Information Systems vol. 17.8, 2023.

[7] Ahn, Sunil, et al. "EDISON-DATA: A flexible and extensible platform for processing and analysis of computational science data." Software: Practice and Experience 49.10, pp.1509-1530, 2019.

[8] Pan, Shirui, et al. "Unifying large language models and knowledge graphs: A roadmap." IEEE Transactions on Knowledge and Data Engineering vol. 36.7, pp. 3580-3599 2024.

[9] Shimizu, Cogan, and Pascal Hitzler. "Accelerating knowledge graph and ontology engineering with large language models." Journal of Web Semantics 85, 100862., pp. 1-6,2025

[10] Kollapally, Navya Martin, et al. "Ontology enrichment using a large language model: Applying lexical, semantic, and knowledge network-based similarity for concept placement." Journal of Biomedical Informatics, 104865, 2025

[11] Erickson, John S., et al. "LLM experimentation through knowledge graphs: Towards improved management, repeatability, and verification." Journal of Web Semantics 85 100853, 2025.

[12] Caufield, J. Harry, et al. OntoGPT v0.3.15. Zenodo, 2024.