

Agentic exploration of knowledge graph data schemas

How data schemas can help AI agents understand knowledge graphs

Uldis Bojārs^{1,*}, Lelde Lāce¹, Mikus Grasmanis¹ and Kārlis Čerāns¹

¹*Institute of Mathematics and Computer Science, University of Latvia, Raina bulvaris 29, Riga, LV-1459, Latvia*

Abstract

The semantic web consists of a large number of knowledge graphs (KG), a subset of which are available for querying as SPARQL endpoints.

To fully explore and query these KGs, a user needs to understand the structure of data, which can be represented via data schemas containing information on each endpoint's classes, properties, and the relations between them, along with relevant statistics. While this information can be retrieved by querying SPARQL endpoints, this process can be time- and resource-consuming and requires a good knowledge of SPARQL.

In this demo we present ongoing work on collecting and exploring data schemas of SPARQL endpoints. The presented system consists of a data shape server (DSS) that contains information about more than 200 SPARQL endpoints, an API for accessing this information from the command line, and a Model Context Protocol server that makes this information available to large language models (LLMs).

We demonstrate how the data schema information can be accessed both from the command line and from LLMs. Accessing schema information from an LLM allows users to engage in conversations about the data structure of SPARQL endpoints that DSS contains information about. The system lets users get an overview of what data is contained in these SPARQL endpoints, identifying their largest classes and most-used properties, and exploring how they are connected together.

Keywords

knowledge graphs, SPARQL endpoints, data schemas, Model Context Protocol, large language models, agentic AI

1. Introduction

Knowledge graphs contain valuable structured information about various topics including public sector data, cultural heritage information and life sciences. Users with knowledge of SPARQL can explore these datasets and retrieve information from one or more endpoints by writing SPARQL queries. They would normally start with exploratory queries to get an understanding of what information a SPARQL endpoint contains – what are the classes and properties used in the dataset, how often they are used and how they are connected.

Running these exploratory queries, however, can be time- and resource-consuming and not all users know what exploratory queries to run. To make this information more accessible, we have collected data schema information from 200+ knowledge graphs [1] and made it available via the DSS API. It can now be accessed using a command-line API and a Model Context Protocol (MCP) server that makes this information available to LLMs.

The primary use of DSS information is to support ViziQuer visual queries [2] and schema diagrams [3]. Its custom schema format (cf. [1]) collects more information than what is usually stored in the more widely used SHACL [4] or VoID schema representations (such as class-to-class and property-to-property relations). Still, the methods presented in this paper could, in principle, be also applicable to data schemas represented in SHACL or VoID. VoID Generator [5], Agentic Data Catalogue support tools [6] and QSE [7] already offer partial support for creating schemas in these formats.

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

*Corresponding author.

✉ uldis.bojars@gmail.com (U. Bojārs); lelda.lace@lumii.lv (L. Lāce); mikus.grasmanis@lumii.lv (M. Grasmanis); karlis.cerans@lumii.lv (K. Čerāns)

🆔 0000-0001-7444-565X (U. Bojārs); 0000-0001-7650-2355 (L. Lāce); 0000-0002-0668-0970 (M. Grasmanis); 0000-0002-0154-5294 (K. Čerāns)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

LLMs allow users to engage in conversations about SPARQL endpoints and may provide them with insights and comments about the KGs that they might not get by looking at the data alone. Thus users can participate in the agentic exploration of the data schemas of SPARQL endpoints that DSS has information about. This requires supplying LLMs with factual information about the data schemas of these endpoints, as described in the next section.

2. DSS and Data Schemas

The data shape server (DSS)¹ provides API access to a knowledge base that contains information about the data schemas of various public SPARQL endpoints. It was initially developed to support visual query building in ViziQuer [2] and is now also used for visualizing SPARQL endpoint data schemas in ViziQuer [3] and for DSS-assisted SPARQL autocompletion. We have previously published information about the resulting ViziQuer schema visualizations but this is the first time that we have made the DSS public and available for use by LLMs.

Our DSS instance² is available online and contains information about more than 200 distinct SPARQL endpoints collected from the LOD Cloud and other open lists of SPARQL endpoints, published using an HTTP API. These SPARQL endpoints cover various areas including generic endpoints (e.g. Wikidata), nature and life sciences (e.g. RDF portal³), scholarly communication (e.g. ORKG [8]) and cultural heritage (e.g. the Sampo series of portals [9]).

Since DSS is open-source software, anyone can also run their own DSS instance which can be useful, for example, for exploring non-public data schemas inside an enterprise.

The information loaded into the DSS instance is collected using the OBIS Schema Extractor tool⁴, which extracts data schemas from SPARQL endpoints and returns them as JSON files [1]. Figure 1 shows a UML-based representation of the information stored in DSS. This metadata schema allows us to represent detailed information about SPARQL endpoint data schemas including classifier (class) information with class instance counts (`:Classifier`), class intersection and subclass information, property information with RDF triple counts (`:Property`) and class-property relations (`:ClassPropertyPair` – where a class instance is a source or a target of a property). Further information about this metadata schema is available in [10].

Table 1 contains a list of the top SPARQL endpoint data schemas in DSS ordered by the number of classes. The largest data schemas present in DSS are DBpedia with 483,453 classes⁵, Wikidata with 85,800 classes and the MMI Ontology Registry and Repository with 6,708 classes. Wikidata and DBpedia data schemas are particularly difficult to acquire due to the size of these KGs. Additional information about these data schemas can be found in [11] for DBpedia and in [12] for Wikidata.

The next section describes the API interfaces that can be used to access the DSS information.

3. API and MCP Access

This demo presents an MCP server that can be used to access the DSS data schemas from LLMs along with a Python command-line API for programmatic access to the DSS information. Both tools are open source and are available on GitHub⁶.

The command-line API (CLI API) makes the DSS HTTP API commands available on the command line, returning the results as JSON. Table 2 lists the commands for exploring DSS data schemas⁷. These

¹<https://github.com/LUMII-Syslab/data-shape-server>

²<https://dss.semttech.lv> – this is an API endpoint; it does not provide a browsable web interface.

³<https://rdfportal.org/about/>

⁴<https://github.com/LUMII-Syslab/OBIS-SchemaExtractor>

⁵Note: this is an older version of DBpedia’s schema. For a more recent schema please refer to the *s26_https_dbpedia_org_sparql* schema that has 1086 classes.

⁶<https://github.com/LUMII-Syslab/dss-ai-tools>

⁷In some cases the DSS API refers to data schemas as ontologies. For consistency, the CLI API and the MCP server follow the same convention.

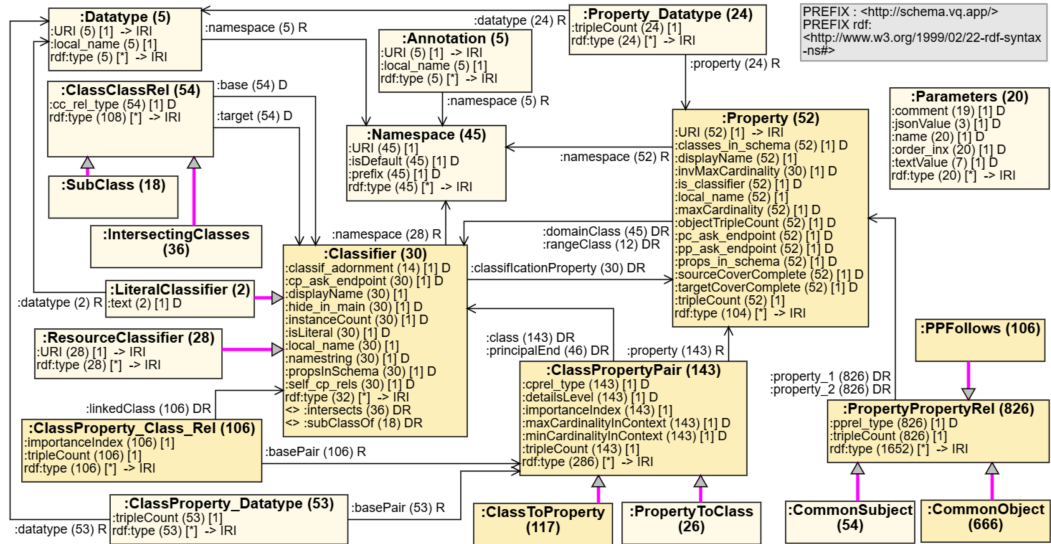


Figure 1: The metadata schema used by DSS (with example data from the Nobel Prizes dataset) [10].

Table 1

Top SPARQL endpoint data schemas in DSS, ordered by the number of classes.

#	Classes	SPARQL Endpoint
1	483,453	https://dbpedia.org/sparql
2	85,800	https://query.wikidata.org/sparql
3	6,708	https://mmisw.org/sparql
4	5,798	http://sparql.glygen.org:8880/sparql
5	5,015	https://forestexplorer.gsic.uva.es/sparql
6	4,074	https://rdfportal.org/primary/sparql
7	3,915	https://metadb.riken.jp/sparql
8	3,782	http://85.254.199.72:8890/sparql
9	3,691	https://sparql.agroportal.lirmm.fr/sparql
10	2,461	https://lov.linkeddata.es/dataset/lov/sparql

commands let users retrieve the list of available data schemas and the main contents of a data schema: classes and properties used in a SPARQL endpoint, the incoming and outgoing properties used together with a class, and the classes used together with a given property. There is also a generic `call` command that allows users to make direct calls to the underlying DSS API for cases when the included commands are not enough.

Model Context Protocol (MCP) is an open-source standard for connecting AI applications to external systems⁸. The DSS MCP server⁹ builds on the CLI API and makes DSS API calls available to LLMs along with instructions on how to use them.

The source code repository also includes a Claude Code skill¹⁰ that lets the Claude Code agentic coding tool access the DSS API. The agentic coding tool provides its users with more flexibility in accessing the DSS API as it can explore source code (e.g. the DSS API code) and develop tools for processing the output of the DSS API.

⁸<https://modelcontextprotocol.io/>

⁹<https://github.com/LUMII-Syslab/dss-ai-tools/tree/main/mcp-server>

¹⁰<https://support.claude.com/en/articles/12512176-what-are-skills>

Table 2

Commands provided by the DSS command-line API

Command	Description
ontologies	List known data schemas
schema-tags	List known schema tags
public-ns	List public namespaces
namespaces	List namespaces for an ontology
classes	List classes in an ontology
properties	List properties in an ontology
resolve-class	Look up a class by name (prefix:local)
resolve-property	Look up a property by name
class-out-properties	List properties outgoing from a class (by class id)
class-in-properties	List properties pointing into a class (by class id)
class-pairs	List (source class, target class) pairs for a property
call	Generic POST /api/ontologies/<ont>/<fn> API call

4. LLM and DSS Usage Example

Figure 2 shows a fragment of an LLM conversation about the Sampo series of SPARQL endpoints with the Claude Desktop tool running Opus 4.8 model. The prompt used was “*Show me information about Sampo data schemas*”. The LLM displays a summary of the various Sampo data schemas that the DSS has information about. Continuing the conversation (for brevity not shown in Figure 2), the LLM offers to “dig” deeper and provide information about classes and properties of one of these data schemas.

To demonstrate these tools in practice, we include a transcript of a larger Claude Desktop conversation¹¹ about the AcademySampo KG. In this conversation a user asks an LLM for information about the largest classes in the AcademySampo KG and then requests information about the use of a particular class (:Timespan) and a property (biocrm:has_family_relation). The LLM responds with detailed information about this class and property including details such as the incoming and outgoing properties of this class. This conversation demonstrates how users can explore DSS data schemas using LLMs, allowing them to “talk” with SPARQL endpoint data schemas rather than having to write SPARQL queries that they might not be familiar with.

5. Conclusions and Future Work

In this paper we presented a public data shape server that contains data schemas of more than 200 distinct SPARQL endpoints. Users can explore this information using the tools included in this demo: a command-line API and a DSS MCP server that makes this API available to LLMs. These tools let users explore the SPARQL endpoint data schemas using LLMs, engaging in “conversations” about these schemas. Users can benefit from this natural language interface by getting a better understanding of what information is available from a SPARQL endpoint and how it is structured. This functionality can be complementary to the data schema visualization approach implemented by ViziQuer. Schema information can also be useful to LLMs by providing the vocabulary needed to generate SPARQL queries over unfamiliar endpoints.

A limitation of the approach described in this paper is that DSS schemas are snapshots-in-time and might not reflect the current state of a SPARQL endpoint. This limitation can be mitigated by refreshing DSS data schemas at regular intervals and providing provenance information on when the dataset was last inspected.

Future work includes improving DSS’s command-line API and MCP server based on user feedback. There is also a potential in adapting these tools to work with data schema information represented as RDF, with the MCP server using SPARQL queries instead of DSS API calls.

¹¹<https://claude.ai/share/dd13bf64-83aa-412a-a7f1-3b353386abd9>

The DSS registry holds several "Sampo" data schemas — these are part of the Finnish [Sampo series](#) of linked open data portals (Semantic Computing Research Group, Aalto University). Here's what's loaded:

Schema	Domain	Classes	SPARQL endpoint
WarSampo	WWII Finnish war history (people, events, units)	90	ldf.fi/warsa
AcademySampoX	Academic people & university history	268	ldf.fi/yoma
PromootioSampo (conferment_sampo)	Doctoral conferment ceremonies	270	ldf.fi/promootiosampo
KirjaSampo (book_sampo)	Fiction literature / books	96	ldf.fi/kirjasampo
CoinSampo	Numismatics / coins	25	ldf.fi/coinsampo
FindSampo	Archaeological finds	33	ldf.fi/findsampo
LawSampo	Legislation & case law	47	ldf.fi/lawsampo
OperaSampo	Opera performances	22	ldf.fi/operasampo

Figure 2: An example of an LLM conversation exploring information about Sampo SPARQL endpoints.

Acknowledgments

This work has been partially supported by a Latvian Science Council Grant lzp-2024/1-0665 “What is in Your Knowledge Graph?”. This publication has been partially supported by COST Action CA24121 – Knowledge Graphs in the Era of Large Language Models (KGELL), supported by COST (European Cooperation in Science and Technology, <https://www.cost.eu>).

Declaration on Generative AI

During the preparation of this work, the authors used Claude Code with the Opus 4.8 model in order to: paraphrase and reword; for formatting assistance. The authors also used Claude Code to assist with developing the DSS CLI and the MCP server presented in this paper. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] K. Čerāns, A. Romane-Ritmane, M. Grasmanis, L. Lāce, OBIS extractor: Create a schema for your knowledge graph, in: Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, CEUR-WS.org, Ghent, Belgium, 2026. Accepted for publication.
- [2] K. Čerāns, A. Šostaks, U. Bojārs, J. Ovčinnikova, L. Lāce, M. Grasmanis, A. Romāne, A. Sprogis, J. Bārzdiņš, ViziQuer: A web-based tool for visual diagrammatic queries over RDF data, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11155 LNCS (2018) 158 – 163. URL: https://viziquer.lumii.lv/papers/2018_ESWC_ViziQuer_Demo.pdf.
- [3] L. Lāce, A. Romāne-Ritmane, M. Grasmanis, A. Sprogis, J. Ovčinnikova, U. Bojārs, K. Čerāns, Visual presentation and summarization of linked data schemas, in: S. Tiwari, B. Villazón-Terrazas, F. Ortiz-

- Rodríguez, S. Sahri (Eds.), Knowledge Graphs and Semantic Web, Springer Nature Switzerland, Cham, 2025, pp. 290–305.
- [4] H. Knublauch, D. Kontokostas, Shapes constraint language (SHACL), W3C Recommendation, 2017. URL: <https://www.w3.org/TR/shacl/>.
 - [5] Bolleman, Jerven, et.al, A large collection of bioinformatics question-query pairs over federated knowledge graphs: Methodology and applications, 2024. URL: <https://arxiv.org/abs/2410.06010>. arXiv: 2410.06010, arXiv preprint arXiv:2410.06010.
 - [6] The Catalogue for the Agentic SPARQL, Web catalogue, 2025. URL: <https://catalogue.ai.wu.ac.at/>, A registry of SPARQL endpoints, ontologies and semantic wikis optimized for agentic AI. Accessed: 2026-08-14.
 - [7] K. Rabbani, M. Lissandrini, K. Hose, Extraction of validating shapes from very large knowledge graphs, Proceedings of the VLDB Endowment 16 (2023) 1023–1032.
 - [8] S. Auer, A. Oelen, M. Haris, M. Stocker, J. D’Souza, K. E. Farfar, L. Vogt, M. Prinz, V. Wiens, M. Y. Jaradeh, Improving access to scientific literature with knowledge graphs, Bibliothek Forschung und Praxis 44 (2020) 516–529.
 - [9] E. Hyvönen, Digital humanities on the semantic web: Sampo model and portal series, Semantic Web 14 (2023) 729–744.
 - [10] S. Rikačovs, K. Čerāns, Visual data and schema queries over knowledge graphs, in: Joint Proceedings of Posters, Demos, Workshops, and Tutorials of the 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW-PDWT 2024), volume 3967 of *CEUR Workshop Proceedings*, CEUR-WS.org, Amsterdam, Netherlands, 2024. URL: https://ceur-ws.org/Vol-3967/PD_paper_188.pdf.
 - [11] K. Čerāns, L. Lāce, M. Grasmanis, J. Ovčinnikova, A UML-style visual query environment over DBpedia, Communications in Computer and Information Science 1537 CCIS (2022) 16 – 27.
 - [12] K. Čerāns, J. Ovčinnikova, M. Grasmanis, L. Lāce, Towards UML-style visual queries over Wikidata, Lecture Notes in Computer Science 13384 LNCS (2022) 11 – 15. URL: https://viziquer.lumii.lv/papers/2022_ESWC_Demo_Wikidata.pdf.