Agentic exploration of knowledge graph data schemas

How data schemas can help AI agents understand knowledge graphs

Uldis Bojārs[1,*], Lelde Lāce[1], Mikus Grasmanis[1] and Kārlis Čerāns[1]

[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 300 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 300+ knowledge graphs and made it available via the DSS API. This information can be accessed using a command-line API and using a Model Context Protocol (MCP) server that makes this information available to LLMs.

LLMs allow users to engage in a conversation with information about SPARQL endpoints and may provide users with insights and comments about the KGs they are exploring that they might not get by looking at the data alone. Thus they can participate in the agentic exploration of the data schemas of SPARQL endpoints that DSS has information about. In order to enable LLMs to let their users explore information about SPARQL endpoints we need to provide LLMs with factual information about these endpoints' data schemas.

SEMANTiCS 2026: 22nd International Conference on Semantic Systems, 15–17 September 2026, Ghent, Belgium

*Corresponding author.

uldis.bojars@gmail.com (U. Bojārs); lelde.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)

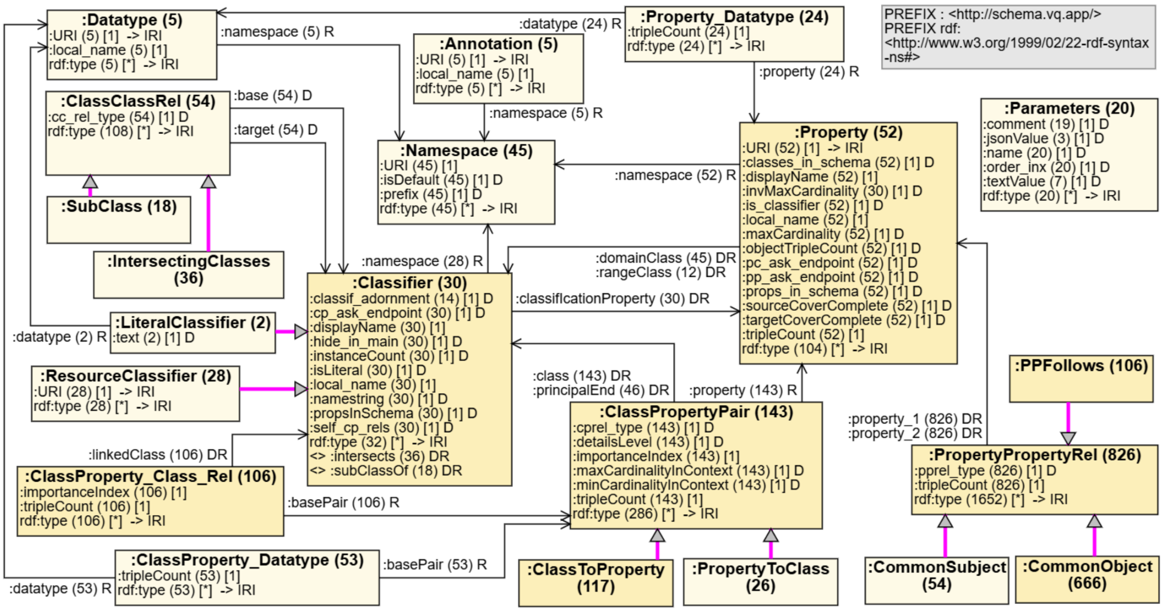

Figure 1: The metadata schema used by DSS (with example data from the Nobel Prizes dataset) [3].

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 [1] and is now also used for visualizing SPARQL endpoint data schemas in ViziQuer [2] and for DSS-assisted SPARQL autocompletion[2]. 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 instance[3] is available online and contains information about 317 distinct SPARQL endpoints collected from the LOD Cloud and other open lists of SPARQL endpoints (the total number of DSS schemas is 401 but some of these schemas are duplicates) published using an HTTP API. Since DSS is open-source software, anyone is free to also run their own DSS instance which can be useful, for example, for exploring non-public data schemas inside an enterprise. DSS currently stores its data in a PostgreSQL server. A visual representation of the metadata schema used by DSS is shown in Figure 1. Further information about this schema is available in [3].

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 classes[4], 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 [1] for DBpedia and in [4] for Wikidata.

These SPARQL endpoints cover various areas including generic endpoints (e.g. Wikidata), nature and life sciences (e.g. RDF portal[5]), scholarly communication (e.g. ORKG [5]) and cultural heritage (e.g. the Sampo series of portals [6]).

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

[1]https://github.com/LUMII-Syslab/data-shape-server
[2]https://yasgui.semtech.lv
[3]https://dss.semtech.lv
[4]Note: this is an outdated version of DBpedia's schema. It also does not include information on class-property-class relations. For a more recent schema please explore the *s26_https_dbpedia_org_sparql* schema that has 1086 classes.
[5]https://rdfportal.org/about/

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

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

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 GitHub[6].

The command-line API (CLI API) takes DSS web API commands and makes them available on the command line, returning the results as JSON. It provides the commands for exploring DSS data schemas shown in Table 2 such as commands for listing the data schemas, and the classes and properties in them[7]. The CLI API also provides commands for retrieving information about the incoming and outgoing properties used together with a class, and information about classes used together with a given property. There is also a generic `call` command that allows users to make direct calls to the DSS HTTP API for cases when the included commands are not enough.

Command output is indented JSON, but there is also an option to output it as single-line JSON that can be easily "piped" to other command-line tools.

The DSS Model Context Protocol (MCP) server[8] is built upon the command-line API and makes the DSS API available to LLMs. It allows an LLM to run the same commands that are available via the CLI API and provides instructions on how an LLM can use them.

[6]https://github.com/LUMII-Syslab/dss-ai-tools
[7]Note: in some cases the DSS API refers to data schemas as ontologies. For consistency, the CLI API follows the same convention.
[8]https://github.com/LUMII-Syslab/dss-ai-tools/tree/main/mcp-server

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 (confernment_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.

The source code repository also includes a Claude Code skill[9] 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.

4. LLM and DSS Usage Example

Figure 2 shows an example of a brief LLM conversation with Claude Desktop tool running the Opus 4.8 model about the Sampo series of SPARQL endpoints. The prompt used was *"Show me information about Sampo data schemas"*. The LLM displays a summary about 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 conversation[10] 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. This allows users to "talk" with SPARQL endpoint data schemas instead of running potentially time-consuming SPARQL queries.

[9]https://support.claude.com/en/articles/12512176-what-are-skills
[10]https://claude.ai/share/dd13bf64-83aa-412a-a7f1-3b353386abd9

5. Conclusions and Future Work

In this paper we presented a public data shape server that contains information about the data schemas of 317 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 and to engage in “conversations” with these data schemas.

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.

Acknowledgments

This work has been partially supported by a Latvian Science Council Grant lzp-2024/1-0665 "What is in Your Knowledge Graph?".

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, L. Lāce, M. Grasmanis, J. Ovčiņņikova, A uml-style visual query environment over dbpedia, Communications in Computer and Information Science 1537 CCIS (2022) 16 – 27.

[2] L. Lāce, A. Romāne-Ritmane, M. Grasmanis, A. Sproģis, J. Ovčiņņikova, 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.

[3] 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.

[4] K. Čerāns, J. Ovčiņņikova, 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.

[5] 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.

[6] E. Hyvönen, Digital humanities on the semantic web: Sampo model and portal series, Semantic Web 14 (2023) 729–744.