Beyond the Blob: Demonstrating MeGraS for Multimodal Knowledge Graph Interaction

Luca Rossetto[1] and Florian Ruosch[2]

[1] Dublin City University, Dublin, Ireland, luca.rossetto@dcu.ie
[2] Zurich, Switzerland, ruosch@ifi.uzh.ch

Abstract. Knowledge Graphs are a powerful tool for the organization of knowledge, especially when it is easily represented in textual form. For knowledge best represented using other modalities, such as vision or sound, Multimodal Knowledge Graphs incorporate multimedia documents into the graph, using either BLOB literals or URI references to documents external to the graph. This makes the media documents opaque to the graph query engine and limits accessibility to the information contained within.
In this paper, we demonstrate **MeGraS**, the **Me**dia**Gra**ph **S**tore, a storage and query engine for Multimodal Knowledge Graphs. MeGraS stores documents directly and makes them available as graph nodes through RDF-compatible URIs. It further offers facilities to address not only complete documents but also arbitrary spatiotemporal segments as part of the graph and dynamically derive information from them at query time. MeGraS can be queried using SPARQL and supports several custom extensions to enrich the query functionality when working with multimodal data.
MeGraS is available as open-source software via `https://megras.org/`, and a demonstration video showing MeGraS in action can be seen on `https://megras.org/MMM2026demo`.

Keywords: Multimodal Knowledge Graphs · Multimodal Media Segmentation · Multimodal Graph Store · Multimodal Graph Query Engine

1 Introduction

Knowledge Graphs are a powerful way of representing information in a structured way, efficiently capturing the interrelations of different concepts. Concepts are represented as nodes in a directed named graph and are commonly labeled with short, human-readable textual labels, indicating their meaning. This is effective for notions that can be concisely represented using a single noun or a short sequence of words. Since *Knowledge* is not limited to atomic elements efficiently represented by language, Multimodal Knowledge Graphs extend the concept by incorporating documents of other media types, such as visual or audio, into the graph. These documents are, however, commonly stored outside the graph and hence remain inaccessible to a graph query engine, which limits the expressiveness of queries in multimodal graphs.

In this paper, we demonstrate **MeGraS** [4], the **Me**dia**Gra**ph **S**tore, a storage and query engine for Multimodal Knowledge Graphs. Contrary to other graph stores, MeGraS not only captures the graph in the form of <subject, predicate, object> triples but also stores the multimodal documents directly as part of the graph, making their content available to the graph query engine. The documents stored by MeGraS become accessible via a consistent URI scheme, which makes them usable directly as nodes of a graph in the Resource Description Format (RDF).[1] Since documents often contain more than one concept, arbitrary spatiotemporal segments can be addressed through the Unified Media Segmentation Model [5]. MeGraS can be queried via the semantic query language SPARQL[2] and supports a range of extensions relevant to multimodal graphs.

2 MeGraS's Capabilities

This section provides an overview of the capabilities of the MediaGraph Store.

2.1 RDF and SPARQL

In addition to several RESTful API endpoints for graph data management and querying, MeGraS provides a standard SPARQL API based on Apache Jena.[3] Beyond the traditional SPARQL data types, MeGraS supports a *vector* literal type, which can encode vector values of arbitrary dimensions. Several operations can be performed on such vectors, as detailed in Section 2.4. Additionally, MeGraS supports the extensions defined in SPARQL-MM [1], as well as some other custom extensions.

2.2 Storing Media Documents as Graph Nodes

The main feature that sets MeGraS apart from other RDF graph stores is that it also serves as a media document store. Media files stored in MeGraS are accessible via HTTP through URIs of the form `http://<host>/<hash>` where `<host>` refers to the resolvable name of the MeGraS instance (using localhost as a default, in case the graph is not to be made publicly accessible) and `<hash>` refers to a file hash of the document itself, represented as a MultiHash.[4] These URIs, when resolved, return the document in its *canonical representation*. This representation can differ from the original format of the document while preserving its content, but allowing for certain storage and accessibility optimizations. For example, adding an uncompressed bitmap image and resolving the URI will return the same image in PNG format. The URIs also serve as graph nodes, and relevant metadata (such as file type, MIME type, etc.) is stored directly in the graph when a document is added. Having the documents accessible and

[1] `https://www.w3.org/TR/rdf12-concepts`
[2] `https://www.w3.org/TR/sparql11-query`
[3] `https://jena.apache.org/`
[4] `https://multiformats.io/multihash/`

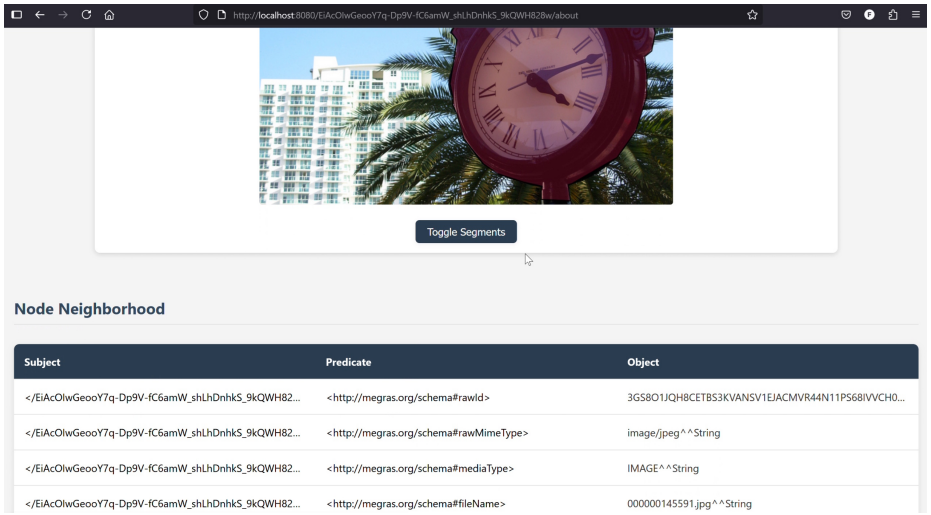

Fig. 1. The about page for an imported image, showing the content, its segments, and node neighborhood in the graph. The example is taken from the COCO dataset [2].

understandable by the graph store also enables further functionality, such as the use of arbitrary media segments as graph nodes.

2.3 Addressing Document Segments

MeGraS implements the Unified Media Segmentation Model [5] to address arbitrary spatiotemporal segments of media documents and to make them available as graph nodes. Such segments can be addressed with URIs of the form

`http://<host>/<hash>/segment/<type>/<definition>`

where the `<type>` corresponds to the way the segment boundaries are defined (e.g., *rect*, *polygon*, *bezier*, etc.) and `<definition>` describes the actual bounds, as defined in [5]. When such a URI is resolved, MeGraS will perform the necessary operations to segment the document and return only the content contained within its bounds, also resolving possible duplicates to avoid redundancies.

An example is shown in Figure 1, where the red marked area corresponds to a segment that is also its own node in the knowledge graph. Furthermore, Figure 2 displays the user interface for applying the segmentation to images, allowing to choose the identifier of the parent, the type, and the definition.

2.4 Vector Operations

As mentioned above, MeGraS supports vector literals in addition to the standard scalar literal values commonly used with SPARQL. This is useful to store embeddings of media documents as part of the graph to be used for similarity search. To support this, MeGraS offers primitives for k-nearest neighbor (kNN)

search and linear support vector machines that are accessible through specialized API endpoints, as well as custom SPARQL functions.

2.5 Implicit and Derived Relations

In addition to basic materialized (explicit) graph triples, MeGraS can dynamically create certain types of triples at query time without the need for prior materialization. This can be the case for two types of relations, which we distinguish as *implicit* and *derived*.

Implicit relations are always defined as between two graph nodes and can be derived from other information available to the query engine without the need for materialization. Examples of such relations include the spatial and temporal relations between segments of the same media document, as defined in SPARQL-MM. Querying for a segment that is spatially to the left of another one or starts after another has ended in time can be done using a standard SPARQL graph pattern using the corresponding relation. The query behaves as if all possible relations between all segments had been previously materialized in the graph.

Derived relations are, in contrast, defined between a graph node as the subject and a literal as the object. The relation has an accompanying function that can be applied to the subject to derive the object literal value, which is then persisted. Hence, the function is only evaluated when the relevant triple is not already present in the graph. In this case, MeGraS would compute the result (e.g., embedding vectors) the first time they are needed to compare a given subject. Subsequent comparisons would then use the previously derived values.

2.6 Interaction and Manipulation

The interaction with the multimodal knowledge graph inside MeGraS happens in three different ways. The first, a command-line interface, offers methods to add multimedia documents and to import information in tab-separated RDF triples. Second, MeGraS has various user interfaces to add to or manipulate the graph. The core UIs are the *file* and the *triple adder*, the *segmenter* (Figure 2), as well as the *about pages* for multimedia documents (Figure 1).

For programmatic access, MeGraS provides a RESTful API, which allows for further and more fine-grained interactions. Mainly directed at client applications and designed for ease of use and interoperability, its most extensive endpoints are those concerning queries. Apart from dedicated methods to filter for subjects, predicates, objects, and their combinations, MeGraS also exposes a SPARQL endpoint. Likewise, its extended functionalities can be accessed, allowing for derived and implicit relations as well as optimized operations with the vector literal type. Furthermore, the Unified Media Segmentation [5] is also covered by MeGraS's API, enabling precise spatiotemporal access to the contents of the multimedia documents.

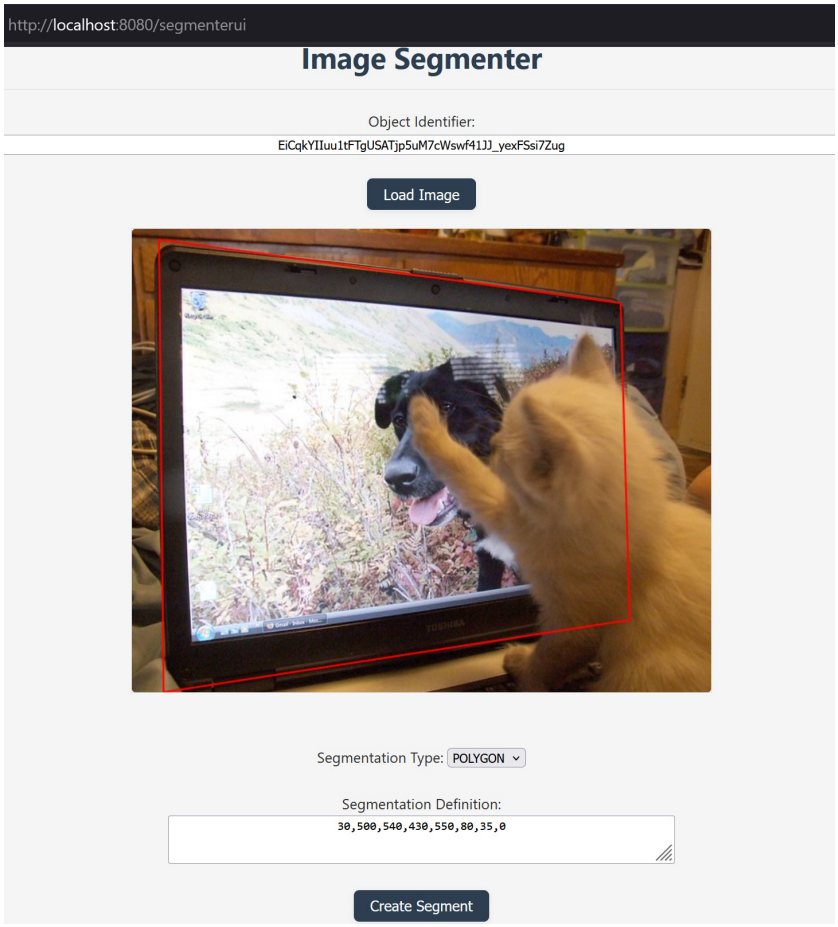

Fig. 2. The user interface for segmenting an image, including a dynamically generated preview. The example is taken from the COCO dataset [2].

3 Demonstration

MeGraS can be compiled from source.[5] Alternatively, it can be downloaded conveniently as a Docker container.[6] Furthermore, a video is available that showcases MeGraS's capabilities.[7] It provides a practical illustration of how MeGraS facilitates the integration and querying of multimodal data.

The demonstration highlights MeGraS's core mechanism for media ingestion. A user interface can be used to add documents to the knowledge graph, resulting

[5] `https://megras.org/`
[6] `https://megras.org/docker`
[7] `https://megras.org/MMM2026demo`

in an *about* page, as shown in Figure 1. Here, the preview of the media file is displayed as well as the relevant subgraph in triples (i.e., the node neighborhood).

Furthermore, to enable the precise addressing and retrieval of only the relevant content from complex media files, a user interface for segmenting these documents is provided, depicted in Figure 2. Creating the segment produces a distinct node in the graph, which only represents the relevant content and annotations. The *about* page of such segments also displays the subgraphs of the ancestor nodes to provide context to the media excerpts.

The second key aspect demonstrated is the seamless process of adding semantic information to these media segments. MeGraS provides a user interface for the ingestion of triples. This capability directly integrates fine-grained annotations, like those from object detection algorithms, into the knowledge graph, making the internal contents of media files queryable at an unprecedented level.

Finally, we also show how to leverage MeGraS's support for implicit and derived relations, and, relatedly, vector literal types. This illustrates a significant advantage: MeGraS computes and persists properties internally when attached to a predefined transformation function (e.g., CLIP [3] embeddings). These values can then be used either for other relations or downstream applications.

In essence, the video provides a tangible walkthrough of MeGraS's innovative approach to making multimodal data first-class citizens within a knowledge graph, enabling rich semantic querying and analysis. Conference attendees will be able to interact with MeGraS directly and experience the various query functionalities. They will be able to add further media documents of their choice to convince themselves of MeGraS's capabilities.

4 Conclusion

In this paper, we presented **MeGraS** [4], the **Me**dia**Gra**ph **S**tore, a novel system for storing, managing, and querying Multimodal Knowledge Graphs. Contrary to traditional approaches, which treat multimodal documents as opaque blobs or external links, MeGraS elevates them to first-class citizens by storing them directly in the graph and making their contents accessible to the query engine.

Our demonstration showcased MeGraS's key capabilities, including its unified URI scheme for document and segment addressing, the support for spatiotemporal segmentation in keeping with the Unified Media Segmentation Model [5], and the extended SPARQL functionalities. MeGraS can ingest and manage multimedia documents and allows queries for explicitly and implicitly stored information, enabling dynamic computation where necessary, such as k-Nearest Neighbor or embeddings. These features enhance query expressiveness and allow for complex analysis across disparate media types not possible in traditional graph stores.

MeGraS is available as open-source software. It offers various interaction methods, including a command-line interface, a RESTful API, and user interfaces for core functionalities. MeGraS can serve as a foundational tool for researchers and developers handling large multimodal datasets for knowledge management and retrieval in the increasingly multimodal landscape.

Acknowledgments. This work was partially funded by the Swiss National Science Foundation through Project "MediaGraph" (Grant Number 202125).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Kurz, T., Schaffert, S., Schlegel, K., Stegmaier, F., Kosch, H.: SPARQL-MM - extending SPARQL to media fragments. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (eds.) The Semantic Web: ESWC 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8798, pp. 236–240. Springer (2014). https://doi.org/10.1007/978-3-319-11955-7_26
2. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V. Lecture Notes in Computer Science, vol. 8693, pp. 740–755. Springer (2014). https://doi.org/10.1007/978-3-319-10602-1_48
3. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021), http://proceedings.mlr.press/v139/radford21a.html
4. Rossetto, L., Ruosch, F.: MeGraS: An Open-Source Store for Multimodal Knowledge Graphs. In: Proceedings of the 33rd ACM International Conference on Multimedia (MM '25). MM '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3746027.3756872
5. Willi, J., Bernstein, A., Rossetto, L.: Unified multimedia segmentation - A comprehensive model for uri-based media segment representation. TGDK **2**(3), 1:1–1:34 (2024). https://doi.org/10.4230/TGDK.2.3.1