Demonstration of Comunica, a Web framework for querying heterogeneous Linked Data interfaces

Joachim Van Herwegen, Ruben Taelman, Miel Vander Sande, Ruben Verborgh

IDLab, Department of Electronics and Information Systems, Ghent University – imec

Abstract. Linked Data sources can appear in a variety of forms, going from SPARQL endpoints to Triple Pattern Fragments and data dumps. This heterogeneity among Linked Data sources creates an added layer of complexity when querying or combining results from those sources. To ease this problem, we created a modular engine, Comunica, which has modules for evaluating SPARQL queries and supports heterogeneous interfaces. Other modules for other query or source types can easily be added. In this paper we showcase a Web client that uses Comunica to evaluate federated SPARQL queries through automatic source type identification and interaction.

1. Introduction

There are a multitude of ways we can access Linked Data these days. Some of the more commonly used ones are SPARQL endpoints [1], Triple Pattern Fragments (TPF) [2] and its variations [3, 4], Linked Data documents [5] and data dumps. These all have their own methods on how they can be accessed and help solve SPARQL queries [6]. While a SPARQL endpoint can execute queries on its own and can require a significant amount of server effort, data dumps will require client-side processing to produce more granular results and is less intensive for servers. This trade-off is measured as client cost on the Linked Data Fragments axis [2].

Having all these heterogeneous interfaces greatly complicates federated queries. While resolving such a query, different actions have to be taken depending on the source that is being accessed. Different solutions might also be required depending on the combination of sources. In case of a single SPARQL endpoint, a single query will suffice. On the other hand, if all sources are data dumps they all have to be downloaded and parsed client-side. But what if some sources are SPARQL endpoints and some are data dumps?

To this end we created a modular Linked Data client called Comunica [7]. In our ISWC 2018 article we describe how this client can easily be extended to support a variety of sources and algorithms. This allows everyone to quickly set up a federated SPARQL client without having to worry about the sources, and to easily extend it should more types be required.

In this article we describe how we will showcase the heterogeneous features of Comunica. We created a Web client capable of executing federated SPARQL queries over heterogeneous interfaces, as an extension of the Triple Pattern Fragments Web

client [8]. Additionally, this client will automatically identify the type of all sources. This way the end-user only has to provide the URLs and query; Comunica will take care of the rest.

2. Comunica

Comunica [7] is a modular meta engine that enables the instantiation of specific engines with their functionality described by modules through semantic configuration files. We released 80+ modules *(https://github.com/comunica/comunica)* that can be combined to fully replicate all features of the original TPF client *(https://github.com/ LinkedDataFragments/Client.js)*. Comunica is not limited to simply solving SPARQL queries: new modules can be added to solve new problems and add additional features. Similarly, existing functionality can easily be switched out for others to quickly compare different implementations and ideas.

Every module consists of two parts: the source code and its semantic description. The second part is a collection of Linked Data documents describing the functionality of the corresponding module. These are used by the Components.js [9] dependency injection framework to instantiate and link all modules together.

The current collection of Comunica modules offers more functionality than the original TPF client. The default configuration allows users to query different kinds of Linked Data besides TPF interfaces. We also provide support for SPARQL endpoints, Linked Data documents and HDT [10] files. These can all be combined in a single federated query by making use of the federated TPF algorithm and utilizing Comunica modules to allow triple pattern queries on those different source types.

3. Demonstration overview

In this demonstration, we offer the possibility for executing SPARQL queries over a federation of heterogeneous interfaces. This demonstration can be used directly within the browser, and is available on the Comunica website *(http:// comunica.linkeddatafragments.org/)*.

This demonstration is an adaption of the Triple Pattern Fragments Web client [8], with the main difference that instead of using the Triple Pattern Fragments engine for querying, it uses the Comunica engine. The implementation of this Web client is available on GitHub *(https://github.com/comunica/jQuery-Widget.js)*, under the open MIT license so that it can be reused for different use cases.

We provide a collection of example queries with a predefined set of sources, where some queries federate over different heterogeneous sources. Fig. 1 shows an example query that federates over a Triple Pattern Fragments interface and a Linked Data document. Additionally, users can also write custom queries, and add more datasources by their URL.
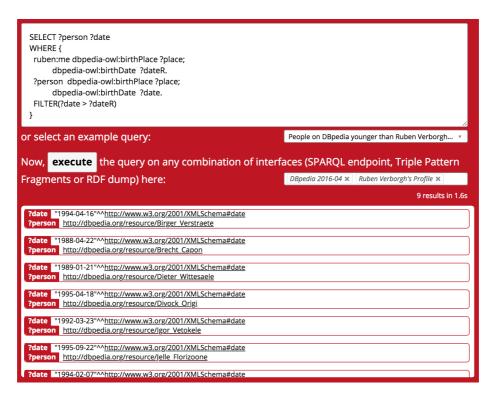
Fig. 1: Example SPARQL query in the Comunica Web client that federates over the DBpedia Triple Pattern Fragments interface and a FOAF profile.

At the time of writing, SPARQL endpoints, Triple Pattern Fragments interfaces and raw RDF files can be queried. Internally, Comunica will identify the source type through a set of heuristics. SPARQL endpoints are tested using a simple `ASK` query through the SPARQL protocol [1]. Triple Pattern Fragments interfaces are tested by checking if the required set of hypermedia controls is available. Finally, RDF files are tested with the lowest priority by checking their content type.

References

1. Feigenbaum, L., Todd Williams, G., Grant Clark, K., Torres, E.: SPARQL 1.1 Protocol. W3C, https://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/ (2013).
2. Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., Colpaert, P.: Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web. Journal of Web Semantics. 37–38, (2016).
3. Hartig, O., Buil-Aranda, C.: Bindings-Restricted Triple Pattern Fragments. In: Proceedings of the 15th International Conference on Ontologies, DataBases, and

Applications of Semantics. pp. 762–779 (2016).

4. Taelman, R., Vander Sande, M., Verborgh, R., Mannens, E.: Versioned Triple Pattern Fragments: A Low-cost Linked Data Interface Feature for Web Archives. In: Proceedings of the 3rd Workshop on Managing the Evolution and Preservation of the Data Web (2017).

5. Berners-Lee, T.: Linked Data. https://www.w3.org/DesignIssues/LinkedData.html (2009).

6. Harris, S., Seaborne, A., Prud'hommeaux, E.: SPARQL 1.1 Query Language. W3C, https://www.w3.org/TR/2013/REC-sparql11-query-20130321/ (2013).

7. Taelman, R., Van Herwegen, J., Vander Sande, M., Verborgh, R.: Comunica: a Modular SPARQL Query Engine for the Web. In: Proceedings of the 17th International Semantic Web Conference (2018).

8. Verborgh, R., Hartig, O., De Meester, B., Haesendonck, G., De Vocht, L., Vander Sande, M., Cyganiak, R., Colpaert, P., Mannens, E., Van de Walle, R.: Low-Cost Queryable Linked Data through Triple Pattern Fragments. In: International Semantic Web Conference (Posters & Demos). pp. 13–16 (2014).

9. Taelman, R.: Components.js. http://componentsjs.readthedocs.io/en/latest/

10. Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary RDF Representation for Publication and Exchange (HDT). Web Semantics: Science, Services and Agents on the World Wide Web. 19, 22–41 (2013).