LOD Surfer API: A Web API for LOD Surfing Using Class-Class Relationships in Life Sciences

Atsuko Yamaguchi[1], Kouji Kozaki[2], Yasunori Yamamoto[1], Hiroshi Masuya[3,4], and Norio Kobayashi[4,3]

[1] Database Center for Life Science (DBCLS),
Research Organization of Information and Systems,
178-4-4 Wakashiba, Kashiwa, Chiba, 277-0871 Japan
{atsuko,yy}@dbcls.rois.ac.jp
[2] The Institute of Scientific and Industrial Research (ISIR), Osaka University,
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
kozaki@ei.sanken.osaka-u.ac.jp
[3] BioResource Center (BRC), RIKEN
3-1-1, Koyadai, Tsukuba, Ibaraki, 305-0074 Japan
hiroshi.masuya@riken.jp
[4] Advanced Center for Computing and Communication (ACCC), RIKEN,
2-1 Hirosawa, Wako, Saitama, 351-0198 Japan
norio.kobayashi@riken.jp

Abstract. Linked Open Data (LOD) is being increasingly used when publishing life science databases. To facilitate the flexible use of such databases, we employ a method that uses federated query search along a path of class-class relationships. To demonstrate our strategy, we implemented a prototype system accessible via a web API as a preliminary trial. We have been collecting SPARQL Builder Metadata (SBM) that describe a data schema for each SPARQL endpoint. Employing the SBM, we constructed a graph of the class-class relationships in LOD. Using this graph, our system can provide the information required to construct a federated search query, such as the SPARQL endpoints that include a class–class relationship, and paths of the class-class relationships between two classes.

Keywords: Linked Open Data, class–class relationships, database integration, federated query search

1 Introduction

LOD Surfer is a search system that discovers data along paths of class–class relationships over life sciences Linked Open Data (LOD) provided by different SPARQL endpoints. A user can interactively obtain desired data by inputting data or class, and selecting an output class and a path of class–class relationships between the classes. In a previous study [1], we developed a system called SPARQL Builder (`http://www.sparqlbuilder.org/`) that allows users to build a SPARQL query for a SPARQL endpoint without thorough understanding of the RDF. For this system, we have been collecting metadata that describes a data schema for each SPARQL endpoint. These metadataa are called SPARQL Builder Metadata (SBM). By applying our SPARQL endpoint crawler program

to the SPARQL endpoints obtained from YummyData [2], we collected SBM for 76 datasets from 43 SPARQL endpoints (as of July 2017).

In this study, we implemented a web API for LOD Surfer as a preliminary trial, using technologies developed for SPARQL Builder such as SBM and class graphs. This enables flexible data acquisition from multiple SPARQL endpoints for users who do not have detailed information for data schema over LOD.

2   LOD Surfer API

The LOD Surfer API is designed to easily develop a search system, such as LOD Surfer. Using this API, a list of SPARQL endpoints, classes reachable using links from a given class, and paths between two

Fig. 1. Overview of the architecture of the LOD Surfer API

given classes can be obtained in JSON format along with a federated SPARQL query and result obtained for a given path by a simple HTTP GET request. Figure 1 shows the overview of the architecture of the LOD Surfer API. From the SBM collected by the LOD crawler, a single graph with 14147 nodes that corresponds to the classes appearing in LOD is constructed. Using this graph, the output generator can efficiently compute the output of this API. For example, from the connected components of the graph, a list of classes reachable from a given class can be instantly obtained. A typical example of the usage for this API is obtaining related data typed by some other classes (e.g., Disease, Protein, etc) from various datasets for a set of IDs typed by a class (e.g. Gene). The code for this web API can be accessed at `https://github.com/LODSurfer/lodsurfer-api`. This system is an early prototype and will be improved to make it more practical for users. In addition, the development for an application using this API will be the focus of our future work.

References

1. Yamaguchi, A., Kozaki, K., Lenz, K., Yamamoto, Y., Masuya, H., Kobayashi, N.: Semantic Data Acquisition by Traversing Class-Class Relationships Over Linked Open Data. The 6th Joint International Semantic Technology Conference (JIST 2016), LNCS 10055, 136–151 (2016)
2. Yamamoto, Y., Yamaguchi, A., Splendiani, A.: Umaka-Yummy Data: A Place to Facilitate Communication between Data Providers and Consumers. 9th International Conference Semantic Web Applications and Tool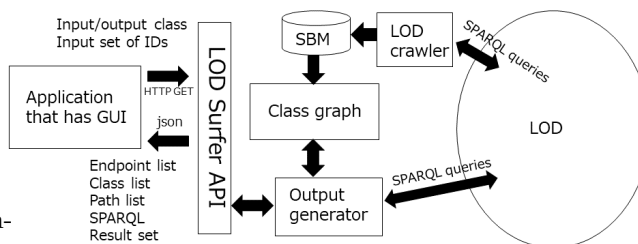s for Life Sciences, CEUR Workshop Proceedings 1795, 2016