Prototype implementation of SPARQL Builder for Life-science Databases by intelligent schema analysis on RDF datasets

Norio Kobayashi[1], Kai Lenz[1], Hongyan Wu[2], Kouji Kozaki[3], and Atsuko Yamaguchi[2]

[1] Advanced Center for Computing and Communication (ACCC), RIKEN,
2-1 Hirosawa, Wako, Saitama, 351-0198 Japan
{norio.kobayashi, kai.lenz}@riken.jp
[2] Database Center for Life Science (DBCLS),
Research Organization of Information and Systems,
178-4-4 Wakashiba, Kashiwa, Chiba, 277-0871 Japan
{wu, atsuko}@dbcls.rois.ac.jp
[3] 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

Abstract. Metadata publication in accordance with the semantic web as a database is a trend for providing and integrating various life-science data. These metadata are published as SPARQL endpoints, a standardised API for RDF datasets. As life-science data are very widely diverse and described using various ontologies and data classes, writing an efficient SPARQL query for SPARQL endpoints is difficult for biologists. To address this problem, we propose an intelligent SPARQL query builder that enables users to build a query without knowledge of SPARQL or the data schema. We have developed a prototype version of the SPARQL builder accessible via users' web browsers. The system crawls SPARQL endpoints in advance to analyse the data schema of large amounts of data, and the resultant crawled data are stored as RDF datasets. This paper focuses on the implementation including the system overview, and the data structure of the resultant crawled data.

Keywords: SPARQL, RDF schema, metadata of RDF datasets, life-science databases

1 Introduction

With the development of life-science research fields and measurement technologies for biological phenomena, the diversity of research data has been increasing. For efficient circulation, intelligent analysis and integration of such heterogeneous data, semantic web technologies including RDF and SPARQL have been adapted, and life-science metadata datasets have already been published as SPARQL endpoints such as the European Bioinformatics Institute (EBI) RDF platform [1], Bio2RDF [2] and BioPortal [3]. However, because such various metadata in RDF are described using specialised ontology terms or data classes in subdivided research fields, writing an efficient SPARQL query that requires complete understanding of the data schema of RDF metadata is a difficult task for biologists as well as bio-informaticians. Efforts to make building a SPARQL query easier have been accomplished; for instance, many SPARQL endpoints provide typical example queries and figures of data schemata. However, they are not enough to cover the wide-ranging interests of biological researchers.

To address this problem, we propose an intelligent web tool named SPARQL Builder that enables users to build a SPARQL query without understanding RDF data schema or SPARQL. We have implemented a prototype version of SPARQL Builder that enables users to build a SPARQL query for existing life-science SPARQL endpoints, including EBI's service. This paper reports the implementation issues of the prototype system.

2 System overview

SPARQL Builder is an intelligent tool that assists a user with no knowledge of SPARQL to generate a query on the basis of a triple path. To be more precise, a triple path $i_1 \xrightarrow{p_1} i_2 \xrightarrow{p_2} \ldots \xrightarrow{p_n} i_{n+1}, \quad (1 \leq n,$ and $n = 3$ is our default) is a sequence of instances $i_1, i_2, \ldots i_{n+1}$ of classes $C_1, C_2, \ldots, C_{n+1}$ respectively, connected by properties $p_1, p_2, \ldots p_n$. A list $C_1, C_2, \ldots, C_{n+1}$ of classes is called a class path if a triple path for the list exists. When a SPARQL endpoint, a start class C_1 and an end class C_{n+1} are specified by a user on the system, the system analyses the metadata of the SPARQL endpoint obtained by a crawler in advance (*cf.* Section 3) and displays possible class paths $C_1, C_2, \ldots, C_{n+1}$. A user further selects a class path. Then the system generates a SPARQL query that searches a triple path corresponding to the selected class path.

Figure 1 shows part of a screen capture of the SPARQL Builder client performing on a web browser. Our prototype system is implemented as a Java servlet, and a user can access through its client written in JavaScript using a web browser. To obtain possible class paths between the user's start and end classes in a practical time, we use a data schema for the SPARQL endpoints called *endpoint metadata*. To construct the endpoint metadata for a SPARQL endpoint, SPARQL Builder throws small but numerous SPARQL queries to the endpoint in advance. The endpoint metadata are written in the vocabulary called *SPARQL Builder metadata*, published at http://sparqlbuilder.org/doc/, and they are stored in the servlet server in RDF. As of September 2014, we have retrieved endpoint metadata from EBI's five SPARQL endpoints of large-scale databases used by the most cutting-edge research, including Expression Atlas[1], BioModels[2], BioSamples[3], ChEMBL[4] and Reactome[5].

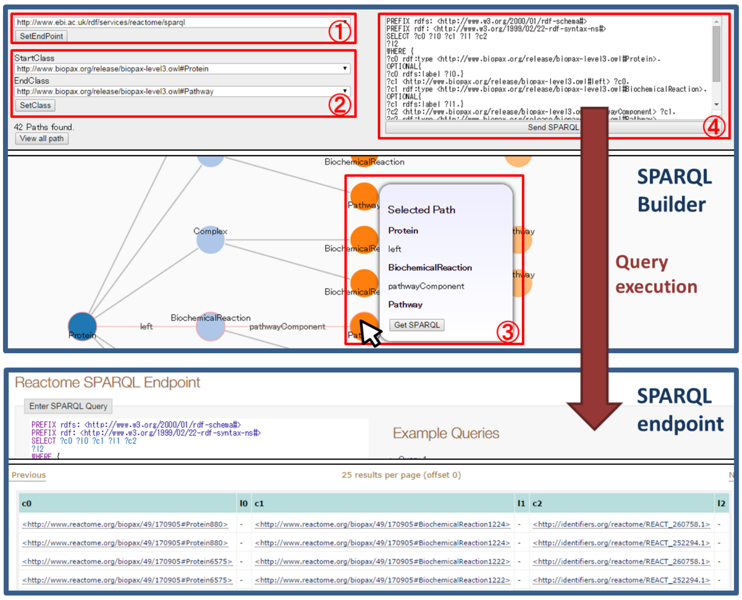
[1] http://www.ebi.ac.uk/rdf/services/atlas/sparql

[2] http://www.ebi.ac.uk/rdf/services/biomodels/sparql

[3] http://www.ebi.ac.uk/rdf/services/biosamples/sparql

[4] http://www.ebi.ac.uk/rdf/services/chembl/sparql

[5] http://www.ebi.ac.uk/rdf/services/reactome/sparql



http://www.ebi.ac.uk/rdf/services/reactome/sparql

StartClass: http://www.biopax.org/release/biopax-level3.owl#Protein

EndClass: http://www.biopax.org/release/biopax-level3.owl#Pathway

42 Paths found.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?c0 ?l0 ?c1 ?l1 ?c2
?l2
WHERE {
?c0 rdf:type <http://www.biopax.org/release/biopax-level3.owl#Protein>.
OPTIONAL{
?c0 rdfs:label ?l0.}
?c1 <http://www.biopax.org/release/biopax-level3.owl#left> ?c0.
?c1 rdf:type <http://www.biopax.org/release/biopax-level3.owl#BiochemicalReaction>.
OPTIONAL{
?c1 rdfs:label ?l1.}
?c2 <http://www.biopax.org/release/biopax-level3.o        ayComponent> ?c1.
```

25 results per page (offset 0)

c0	l0	c1	l1	c2	l2
<http://www.reactome.org/biopax/49/170905#Protein880>	-	<http://www.reactome.org/biopax/49/170905#BiochemicalReaction1224>	-	<http://identifiers.org/reactome/REACT_260758.1>	-
<http://www.reactome.org/biopax/49/170905#Protein880>	-	<http://www.reactome.org/biopax/49/170905#BiochemicalReaction1224>	-	<http://identifiers.org/reactome/REACT_252294.1>	-
<http://www.reactome.org/biopax/49/170905#Protein6575>	-	<http://www.reactome.org/biopax/49/170905#BiochemicalReaction1222>	-	<http://identifiers.org/reactome/REACT_260758.1>	-
<http://www.reactome.org/biopax/49/170905#Protein6575>	-	<http://www.reactome.org/biopax/49/170905#BiochemicalReaction1222>	-	<http://identifiers.org/reactome/REACT_252294.1>	-

Fig. 1. Graphical user interface of SPARQL Builder. (1) A user first selects a SPARQL endpoint, and then class lists for selecting a start class and end class are displayed. (2) When the user selects start and end classes, all possible class paths are displayed as a tree. (3) The user then selects a path, and the system generates the corresponding SPARQL query. (4) Finally, when the user clicks the SPARQL button, the system sends the generated SPARQL query to the SPARQL endpoint, and the result is displayed in a new window of the web browser.

Though our SPARQL Builder itself is an individual application, it is designed to work in conjunction with TogoTable [4], a web application that enables biological researchers to upload a table from a user's data and to add annotations obtained from SPARQL endpoints. SPARQL Builder assists users in obtaining annotations from SPARQL endpoints without knowledge of SPARQL. The TogoTable service built with SPARQL Builder will be released to the public as the next version and will be evaluated regarding practicality of the tool.

3 SPARQL Builder metadata

SPARQL Builder metadata briefly and comprehensively describes an RDF graph schema of SPARQL endpoint datasets. Other specifications defined for a similar purpose include the vocabulary of interlinked datasets *VoID*[6] and the vocabulary for describing SPARQL services *SPARQL 1.1 Service Description*[7]. SPARQL Builder metadata is based on these existing specifications but is defined by adding our original vocabularies that describe metadata for constructing class paths and statistics to determine comprehensiveness of the data that can

[6] http://www.w3.org/TR/void/

[7] http://www.w3.org/TR/sparql11-service-description/

be handled by our search method on the basis of class paths. For instance, our original class *ClassRelation* is used to describe a relationship of two classes correlated with property p that is essential to build a class path. In order to improve comprehensiveness of our triple path search, the class–class relationship as a ClassRelation is not only the domain and range classes of property p but also the classes of subject and object instances of triples having property p.

As described above, SPARQL Builder metadata is our original specification, but it is defined for arbitrary SPARQL endpoints. Some life-science SPARQL endpoints provide metadata for their datasets. EBI publishes such metadata in their framework called Lodestar[8], and Bio2RDF publishes Bio2RDF Dataset Metrics[9]. We hope these metadata specifications are integrated as a global standard and promote distribution of metadata for advanced intelligent semantic web data processing.

4 Conclusions

We discussed our prototype version of the SPARQL Builder tool, which enables users to discover a sequentially connected triple path for arbitrary SPARQL endpoint without knowledge of the data schema or SPARQL. In order to a build SPARQL query by interaction with a user in a practical time, the metadata of the datasets provided by SPARQL endpoints are retrieved in advance and the results are stored as RDF datasets followed by a SPARQL Builder metadata specification. Our future work includes support for SPARQL queries not only for triple paths of a sequence of instances but also general structures such as trees, verification and improvement of practicality of our prototype system.

Acknowledgments. We thank Dr Yasunori Yamamoto for useful comments for improvement of the SPARQL Builder metadata specification.

References

1. Jupp, S., Malone, J., Bolleman, J., Brandizi, M., Davies, M., Garcia, L., Gaulton A., Gehant, S., Laibe, C., Redaschi, N., Wimalaratne, S. M., Martin, M., Le Novére, N., Parkinson, H., Birney, E., Jenkinson, A. M.: The EBI RDF platform: linked open data for the life sciences. Bioinformatics 30(9), 1338–1339 (2014)
2. Belleau, F., Nolin, M. A., Tourigny, N., Rigault, P., Morissette J.: Bio2RDF: towards a mashup to build bioinformatics knowledge systems. J. Biomed. Inform. 41(5), 706–716 (2008)
3. Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C., Tudorache, T., Musen, M. A. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucl. Acids Res. 39(Web Server issue), W541–545 (2011)
4. Kawano, S., Watanabe, T., Mizuguchi, S., Araki, N., Katayama, T., Yamaguchi, A.: TogoTable: cross-database annotation system using the Resource Description Framework (RDF) data model. Nucl. Acids Res. 42(W1), W442–W448 (2014)

[8] http://www.ebi.ac.uk/fgpt/sw/lodestar/

[9] https://github.com/bio2rdf/bio2rdf-scripts/wiki/Bio2RDF-Dataset-Metrics