The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

LOD Surfer API: クラス間関係に基づく LOD 探索のためのウェブ API

LOD Surfer API: A web API for LOD exploration based on class-class relationships

山口敦子[*1] 小林紀郎[*2] 桝屋啓志[*3] 山本泰智[*1] 古崎晃司[*4]
Atsuko Yamaguchi Norio Kobayashi Hiroshi Masuya Yasunori Yamamoto Kouji Kozaki

[*1] 情報・システム研究機構 ライフサイエンス統合データベースセンター
Database Center for Life Science, Research Organization of Information and Systems

[*2] 理化学研究所 情報基盤センター
Advanced Center for Computing and Communication, RIKEN

[*3] 理化学研究所 バイオリソースセンター
BioResource Center, RIKEN

[*4] 大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

In life-sciences domain, Linked Open Data (LOD) is being increasingly used when publishing databases. For 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. Using the SBM, we constructed a graph called a merged class graph based on 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.

1. はじめに

日々生産される生命科学のデータは巨大かつ多岐にわたる．実験によって得たデータを解釈し，生物学的に新たな知識を得るためには，これらのデータ全体を統合し，自在に組合わせることが必須となってきている．このような状況のもと，数多くの研究施設が生産した生命科学データの RDF 化をすすめ，Linked Open Data (LOD)として公開してきている．LOD 上でデータベースを公開する際には，RDF の標準的な検索言語である SPARQL を用いた検索が可能なウェブ API である，SPARQL エンドポイントが提供されることが多い．その結果，LOD 上の SPARQL エンドポイントは次々と新たに提供されており，ユーザがその全体像を理解するのは困難である．

一方，著者らはセマンティックウェブ技術にあまり詳しくないユーザを対象に，SPARQL クエリ生成支援システム SPARQL Builder を開発し，2013 年 10 月からサービス運用を行ってきた[1,2]．その際，高速にクラス間関係提示を行うため，予め LOD 上の SPARQL エンドポイントをクロールし，SPARQL エンドポイントで提供されるデータのデータスキーマを記述するメタデータを取得し蓄積してきた．

そこで，これらのメタデータを利用し，LOD 全体の構造，および，どのエンドポイントから必要な情報が得られるかをユーザが把握する必要がなく，柔軟に LOD からデータを切り出すことを可能とする情報基盤 LOD Surfer を提案する．本発表では特に，LOD Surfer を実現するにあたり，必要な機能と課題を検討するために，プロトタイプ実装を行ったウェブ API，LOD Surfer API について紹介する．

連絡先：山口敦子，情報・システム研究機構 ライフサイエンス統合データベースセンター，〒277-0871 千葉県柏市若柴 178-4-4-6F，atsuko@dbcls.rois.ac.jp

2. SPARQL Builder Metadata

2.1 SPARQL Builder

SPARQL Builder とは，SPARQL 言語の知識がなくとも，また対象データセットの構造を知らなくても，対話的な GUI を介して SPARQL クエリを生成することができることを目指して開発されたウェブ上のサービスである[4].

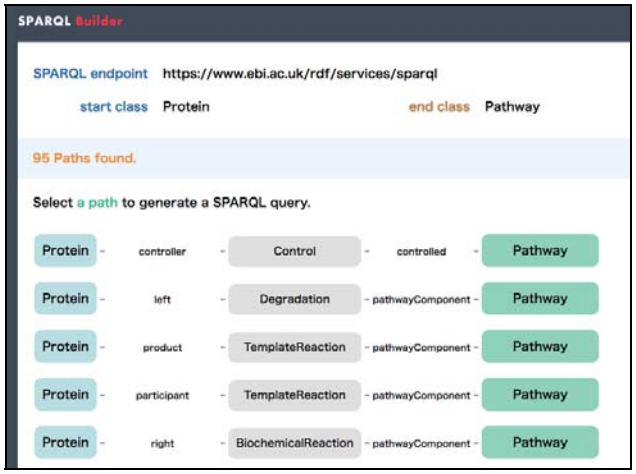

図 1 SPARQL Builder

SPARQL Builder はインスタンス間がどのようなプロパティでつながっているかというクラス間の関係をベースとして SPARQL クエリの記述補助を行う．ユーザはまず，入力クラスと出力クラスをそれぞれクラスのリストから選ぶ．入出力の二つのクラスが確定すると，それらのクラス間の関係でデータ内に含まれるものが

ユーザに提示される. ここで, ユーザが望むクラス間の関係は, データ内ではどのプロパティを用いて記述されているか分からないことに加え, 複数のトリプルを用いた多段の関係により記述されている可能性もある. そこで, ユーザの欲しい関係を提示できるよう, 入出力クラス間の多段の関係, つまりクラス間関係のパスを提示する. 現在の SPARQL Builder では, 最大 4 段のクラス間関係パスをすべて提示する. ユーザがパスを一つ選ぶと, そのパスのクラス間関係に対応する SPARQL クエリが自動生成される. 生成された SPARQL クエリはそのまま検索へ利用することができ, また, 生成された SPARQL クエリを編集して利用することも可能なシステムとなっている.

2.2 SPARQL Builder Metadata の概要

ユーザにデータセットやクラスの情報を提示するために必要なデータ, および, パスの計算を高速に行うために必要なデータは, 事前に SPARQL エンドポイントから取得し蓄積しておくことが望ましい. この目的で設計されたメタデータが SPARQL Builder Metadata (SBM)である. SBM はそれ自身が RDF で記述される.

SBM のスキーマ仕様は, データセットのメタデータ記述語彙 VoID[1] および, SPARQL エンドポイントのウェブ上のサービスとしてのメタデータ記述語彙 SPARQL 1.1 Service Description[2] の拡張として定義されている. 大まかには, SPARQL エンドポイント→エンドポイントに含まれるデータセット→データセットのメタデータの階層構造になっており, 各メタデータ部分にはクラスリスト, プロパティリスト, クラス-プロパティ-クラス関係, さらにそれらに関連するトリプル数等の統計情報が含まれる. 詳しくは http://www.sparqlbuilder.org/doc/sbm_2015sep/ を参照されたい.

SPARQL Builder を運用するにあたり, 現在, 生命科学分野の 43 の SPARQL エンドポイントから 76 のデータセットのメタデータを取得し蓄積している. メタデータ取得は主に, SPARQL エンドポイントに対し, 一連の SPARQL クエリを投げ, その結果から SBM を構成する. SBM 取得プログラムは GitHub にて公開されており, https://github.com/sparqlbuilder/metadata から入手可能となっている.

3. LOD Surfer API

3.1 LOD Surfer 概要

L OD Surfer は, LOD から自分のデータに紐付いたデータの中から, 欲しいデータがどの SPARQL エンドポイントに含まれるか気にすることなく, 興味がある部分を柔軟に切り出すことを可能とすることを目指して設計されたシステム基盤である. 対象データはウェブ上に分散しているため, LOD Surfer を実現するためには, データの所在を自動的に見つけ, さらにそのデータを適切かつ現実的な時間で取得する必要がある.

まず, データが所在する SPARQL エンドポイントを見つけるためには, 先述の SBM が利用可能である. SPARQL エンドポイントから必要なデータを適切かつ現実的な時間で取得することを担保するためには, Umaka-YummyData[3]を利用する. Umaka-YummyData は, 生命科学データを中心とした SPARQL エンドポイントのリストに対し, 毎日アクセスを行い, サービスの死活, 更新頻度, VoID 等のメタデータ提供状況, クエリ返答速度などを監視し, その結果をスコア付けするシステムである. このシステムのウェブ API を利用し, 稼働率が低い SPARQL エンドポイントやクエリ返答速度が遅い SPARQL エンドポイントをフィルタすることで, 検索先のサービスが返答せず, データが取得できないことを回避する.

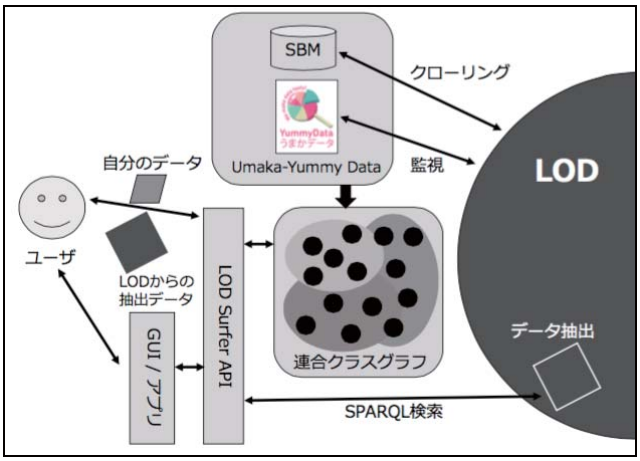

図 2 LOD Surfer システム構成

図 2 は LOD Surfer のシステム構成設計の概要である. Umaka-Yummy Data によりフィルタされクエリ応答が担保された SPARQL エンドポイントのリストに対し, 対応する SBM 形式のメタデータを利用し, LOD 全体のクラス間関係を示すグラフ, 連合クラスグラフを構築する. ユーザが LOD Surfer API を通じて自分のデータを入力すると, そのデータのクラスを始点としたクラス間関係パスを, 連合クラスグラフを用いて計算し, 到達可能なクラスのリストを示す. それらのクラスのリストから, ユーザが興味をもったクラスを指定すると, クラス間パスからフェデレート検索クエリを生成し, クエリを LOD 上で実行し, その結果をユーザへ返すことができる.

計算機に詳しくなく, LOD Surfer API を直接利用することが難しいユーザに対しては GUI やアプリケーションソフトウェアを通じて API を利用するように, 今後開発を進める予定である.

3.2 連合クラスグラフと連結成分

LOD をクロールした結果, SBM には LOD 内のクラスおよびクラス間関係情報が蓄積される. そこで, 蓄積した全ての SBM に対し, 一度でも現れるクラスを頂点にし, クラス間関係を辺としたグラフを考える. このグラフを連合グラフという.

LOD Surfer におけるフェデレート検索クエリ構築もクラス間関係をベースとするため, 効果的なフェデレート検索の設計のためには連合クラスグラフの構造の解析が必要である. そこで, 現在 SPARQL Builder で利用中の 43 の SBM を利用し, 連合クラスグラフを構築し, その解析を試みた. 特に, クラス間の到達可能を考慮するために連結成分を中心に解析を行った. ちなみに, グラフの連結成分とは, グラフを互いにパスを持つ極大頂点集合に分割したものである. 言い換えるならば, 同じ連結成分内のクラス間には何らかのパスがあるといえる.

SBM から構築した連合グラフは, 頂点数 14,147, 辺 18,738 のグラフとなった. このグラフの連結成分を計算し, その頂点数(=クラス数)の分布を調べた. 連結成分の数は 6,857 個であり, そのうち, 6,814 個が孤立点であった. 一方, 最大の連結成分は 5,327 個のクラスを含み, 孤立しない 7,333 個のクラスのかなりの部分を最大の連結成分が占めることがわかる.

[1] https://www.w3.org/TR/void/
[2] https://www.w3.org/TR/sparql11-service-description/

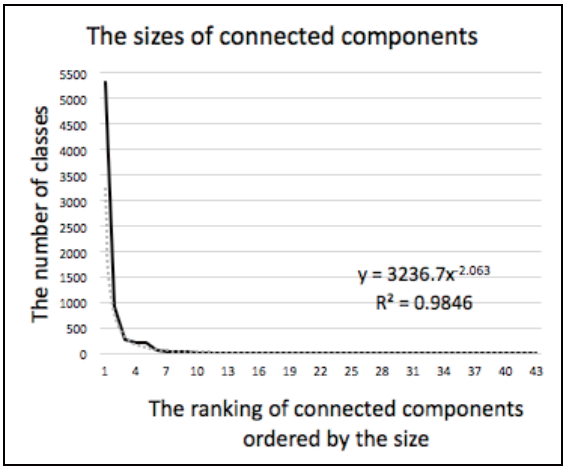

図 3 連合グラフの連結成分数の分布

図 3 は，孤立点でない連結成分について，連結成分の頂点数をサイズが大きいものから順に示したものである．上下二つのグラフは Y 軸に関し，上は線形軸，下は対数軸となっている．分布は冪乗則に従うように見え，事実，最小二乗法でフィッティングすると $y=3236.7x^{-2.063}$ に対し，その決定係数 R2 は 0.9846 と十分 1 に近い値を取る．

このことから，(1)連合クラスグラフは連結成分を予め計算し，連結成分ごとに別に管理することで，探索空間を大幅に減らすことができる，(2)孤立点となるクラスが多いので，上位クラスでまとめる工夫が必要である，ということができる．

3.3 API 実装

LOD Surfer API はユーザがクラス間関係を利用して LOD から効率良くかつ柔軟にデータを切り出すためのウェブ API である．この API を利用して，対象となる SPARQL エンドポイント一覧，クラス一覧，あるクラスから到達可能なクラスの一覧，二つのクラスに対し，連合クラスグラフ上のクラス間パスのリスト，クラス間パスに対し，SERVICE 句付きのフェデレート検索クエリを出力することができる．

これらの機能を実現するウェブ API を下記のように設計した．

- /eplist?class1={class1 URI}
  class1 をデータ内に含む SPARQL エンドポイントのリストが JSON 形式で出力される．class1 が指定されない場合，全ての対象 SPARQL エンドポイントのリストが出力される．
- /clist?class1={class1 URI}
  class1 をデータ内に含む SPARQL エンドポイントのリストが JSON 形式で出力される．class1 が指定されない場合，全ての対象 SPARQL エンドポイントのリストが出力される．
- /path?class1={class1}&class2={class2}
  class1 から class2 へのパスのリストが JSON 形式で出力される．
- /sparql?path={path}
  path に対応する SERVICE 句付き SPARQL クエリがテキスト形式で出力される．

これらのウェブ API は，先述の連合クラスグラフおよびその連結成分を用いて高速に実行できる．例えば，到達可能性は連結成分の頂点のリストと一致するため，これを用いて瞬時に計算できる．パスも連結成分内のみに探索空間が限定されるため，現実的な時間で計算できる．

LOD Surfer API の有効性を確認するため，現在 SPARQL Builder で利用中の 43 の SBM に対し，プロトタイプを作成した．実装にあたり，このウェブ API の諸機能のための計算は直接 RDF 形式の SBM を利用するわけではなく，パス探索や連結成分計算などのグラフアルゴリズムが適用しやすいように，一旦，連合クラスグラフを表す隣接リスト構造へ変換したものを利用した．さらに，パス探索の効率性を上げるために，予め連結成分を計算したうえで，成分ごとに頂点のハッシュ集合で保持した．これらの構造上で，上記の機能を持つウェブ API を作成した．

LOD Surfer API をプロトタイプ実装したサービスは http://lodsurfer.dbcls.jp/api/で試すことができる．例えば，全クラスのリストは http://lodsurfer.dbcls.jp/api/clist で出力することができる．また，LOD Surfer API において使われているシステムのコードは https://github.com/LODSurfer/lodsurfer-api から取得可能である．

4. まとめと今後の課題

クラス間関係をベースとした一つのグラフを構築することで，LOD 上のデータを柔軟に切り取る手法を提案した．また，この手法のプロトタイプ実装として，SPARQL Builder 運用において収集した SBM を用い，連合クラスグラフを構築して，連合クラスグラフ上でパス探索などの諸機能を実現し，ウェブ API を開発した．

今後の課題は，現在の枠組みで検索対象にならない，あるいはなりにくいデータへの対応である．連合クラスグラフの解析結果より，インスタンスが少ないクラスや孤立したクラスが数多くあることがわかった．これらのクラスに対しては，より上位クラスで下位クラスをまとめたうえでクラス間関係を検索することが望ましいと思われる．しかしながら，現時点での SBM は rdfs:subClassOf への対応が十分ではなく，SBM の情報だけでは，下位クラスを適切に上位にまとめることが難しい．そのため，今後，SBM を LOD Surfer 向けに設計を直す必要があり，また，そのためのクローラシステムの開発も必須である．

また，今回の LOD Surfer API のプロトタイプ実装では，最終的にフェデレート検索クエリの出力で留まり，クエリの処理は他の SPARQL エンドポイントへ任せる設計になっている．今後，SBM 内の統計情報などを利用することにより，データの結合を効率的に行える順序を計算することができると思われる．その手法を使ったクラス間関係ベースのフェデレート検索システムの提案も考えたい．

謝辞

本研究のは独立行政法人科学技術振興機構(JST)、バイオサイエンスデータベースセンター (NBDC) の助成，および科学研究費補助金基盤(C) 17K00434，17K00424，基盤(B) 17H01789 の助成による．

参考文献

[Yamaguchi 2014] Yamaguchi, A., Kozaki, K., Lenz, K., Wu, H., Kobayashi N.: An Intelligent SPARQL Query Builder for Exploration of Various Life-science Databases, The 3rd International Workshop on Intelligent Exploration of Semantic Data (IESD 2014), CEUR Workshop Proceedings 1279, 2014

[Yamaguchi 2016] 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

[Yamamoto 2018] Yamamoto, Y., Yamaguchi, A., Splendiani, A.: YummyData: providing high-quality open life science data. Database, doi: 10.1093/database/bay022, 2018