SPARQL Builder: 生物学研究者が SPARQL を使いこなすための補助ツール

SPARQL Builder: A tool for biologists to build a SPARQL query

山口敦子[1]　古崎晃司[2]　呉紅艶[1]　小林紀郎[3]

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

[1] ライフサイエンス統合データベースセンター
[1]Database Center for Life Science
[2] 大阪大学 産業科学研究所
[2] The Institute of Scientific and Industrial Research, Osaka University
[3] 理化学研究所 情報基盤センター
[3] Advanced Center for Computing and Communication, RIKEN

概要: .生命科学で扱う多様なデータを有機的に扱い統合するために, セマンティックウェブ技術を活用したデータベース構築が広く行われるようになってきた．RDF 化され，SPARQL エンドポイントを公開する生命科学系データベースの数は年々増加している．そこで，生物学研究者の多種多様なユースケースに対応したデータ処理を行うためには, SPARQL エンドポイントを有効に活用したアプリケーション開発が必要となる．しかし，SPARQL クエリは RDF データのグラフ構造を熟知していなければ記述することができず, 生物研究者が欲しいデータの取得やさらに取得したデータを効果的に処理するアプリケーションの構築は一般に困難である．そこで，あらかじめ RDF データ構造を知らなくても生物学研究者が欲するデータを取得する SPARQL クエリを生成できるツール SPARQL Builder の試作を行った．本発表では，システム設計と，試作で得られた問題点や知見について述べる．

1. はじめに

1.1 背景

生命科学研究においては，分野の細分化や計測機器の発展に伴い，多種多様かつ膨大なデータが産出される．これらのデータを集約し公開するデータベースは，生物学研究者に必要不可欠なインフラとして利用されている．しかしながら，従来，プロジェクトや目的ごとなど個別にデータベースが作成，公開されてきたため，データベースの所在や利用方法がわかりにくいなど多くの問題を抱えていた．

近年では，生命科学の多種多様かつ膨大なデータを統合的に扱うために，セマンティックウェブ技術に基づいたデータの記述や公開が推進されてきた．タンパク質配列データベース UniProt [1,2] では，2008 年頃から大量のデータベース間のリンク情報を扱うために RDF データモデルを採用している．2013 年 10 月にはヨーロッパ最大のバイオインフォマティクスの拠点である EMBL-EBI が，UniProt に加えて，ChEMBL, Expression Atlas など生物学でよく使われるデータベースを RDF 化し, その RDF データにアクセスできる SPARQL エンドポイントを公開した[3].

このような状況の下，RDF 化され SPARQL エンドポイントで公開されるデータを一般の生物学研究者が有効活用できるようにするためには，SPARQL に準拠した研究者の要求に沿ったアプリケーション開発が必要である．そのために，アプリケーションで必要と思われる SPARQL クエリを予め用意しておく方法が現在は主流である．例えば，BioGateway では，データセットと共に SPARQL クエリのセットが用意されている[4]．しかしながら，生物学研究者の要求するデータは非常に多種多様であり，RDF のグラフ構造もデータ毎に異なり，データ統合によってグラフ構造も順次拡張される．このため，予めすべての SPARQL クエリをアプリケーション側で用意することは不可能である．また， SPARQL クエリ

の構築に当たっては，RDF データの構造や仕様を熟知し，自分の欲しいデータ構造を精緻に記述することが求められるが，一般の生物学研究者にとってこの作業は技術的に難しい．

我々は，RDF に準拠したアプリケーション TogoTable を例に取り，TogoTable への SPARQL クエリの半自動的追加を目的として，一般の生物学研究者が自分の欲しいデータを SPARQL で記述できる手法を確立する研究を行っている．本発表ではこれまでに検討した手法を述べ，それを SPARQL Builder と呼ぶツールとして試作した結果，実装に伴う問題点，さられその解決方法について述べる．

1.2 表データ注釈付加ツール「TogoTable」

TogoTable はユーザがアップロードした表形式のデータに対して，公開されている SPARQL エンドポイントから関連するデータを付加できるウェブツール(図 1)である[3].

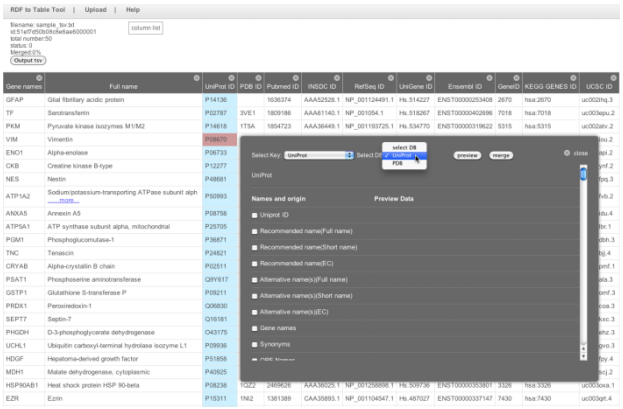

図 1: TogoTable

典型的な使い方としては，(1) 自分の実験データを表形式でアップロードする，(2) アップロードした表の中からデータベースの ID をデータとして含むカラムを 1 つ選び，データを取得したいデータベースをリストから選ぶ，(4) 取得可能なデータの種類が表示されるので，取得したいデータの種類を選ぶ，(5) ユーザが選んだカラム(= ID の集合)とデータの種類を利用して，予め作成された SPARQL クエリリストからクエリを選び，ユーザが選んだデータベースの SPARQL エンドポイントから取得して，表の左端に新たなカラムとして表示する，である．

データベースごとに取得可能なデータの種類はそれを取得可能な SPARQL クエリとペアで予めアプリケーション内に登録されている．この SPARQL クエリのリストは，段階(4)での取得可能なデータの種類の表示，および，段階(5)での SPARQL エンドポイントからのデータ取得に利用されている．

2. SPARQL Builder

2.1 開発要件

SPARQL を全く書けないユーザに任意の SPARQL エンドポイントから任意の SPARQL クエリが書けるような補助をするのは様々な面で困難である．例えば，エンドポイントごとに RDF データをブラウズさせ，入力データと目的のデータをユーザが示すことで SPARQL クエリを自動記述する方法を考える．しかしながら，予めユーザがデータベースの全体像を知らなければ，目的のデータがそのデータベースにあるかどうかさえ不明であり，あるとしても，どちらへどうブラウズすれば目的のデータにたどり着けるか不明である．

そこで，本発表では，目的を「ユーザが TogoTable の SPARQL クエリリストを半自動的に拡充可能とするツール」と状況を限定することで，必要とされる SPARQL クエリの形式の限定を試みた．

表形式データに対して，各カラムは何らかのクラスに対応すると仮定する．この場合，ユーザが TogoTable インタフェイスを通じて指定するのは，SPARQL エンドポイント，入力クラス(前述した TogoTable の段階(2)に相当)および出力クラス(TogoTable の段階(4)に相当)である．しかしながら，ある RDF グラフにおいて，入力クラスや出力クラスを rdfs:domain，rdfs:range にもつことができる Property は一般に複数あり，また，途中に複数のクラスを挟んで関連をもつこともありえる．そして，入力クラス内のインスタンスと出力クラスのインスタンスがグラフ上でどの経路を通じてつながっているかによって，インスタンス間の関係性は異なる．したがって，ユーザが要求する関連データを出力させるには，SPARQL エンドポイント，入力クラス，出力クラス，入力クラスと出力クラスの間の経路，の 4 つを指定させる必要がある．

本発表では，この 4 つをユーザに指定させ，それをもとに SPARQL クエリを作成するシステム SPARQL Builder について考える．

2.2 システム概要

前述した開発要件を満たすため，SPARQL Builder は，(1)ユーザが SPARQL エンドポイント，入力クラス，出力クラス，クラス間経路を指定する画面，

(2) 経路計算モジュール，(3) 経路から SPARQL クエリを作成するモジュールの 3 つの部分から構成される．

図 2 は SPARQL Builder のフローを表したものである．まず，画面から SPARQL エンドポイント，入力クラス，出力クラスをユーザより取得する．取得した情報は経路計算モジュールに引き渡される．経路計算モジュールは，入力クラスの隣接クラスをあるステップ数まで SPARQL エンドポイントに問い合わせて取得し，その情報から入力クラスを始点とし出力クラスを終点とする経路をすべて計算する．計算された経路群は，画面に表示される(図 3)．ユーザが表示された経路群の中から一つの経路を選ぶと，その経路から SPARQL クエリが作成され，画面に表示される．また，作成されたクエリは SPARQL エンドポイントに投げられ，得られた結果も画面に表示される(図 4)．

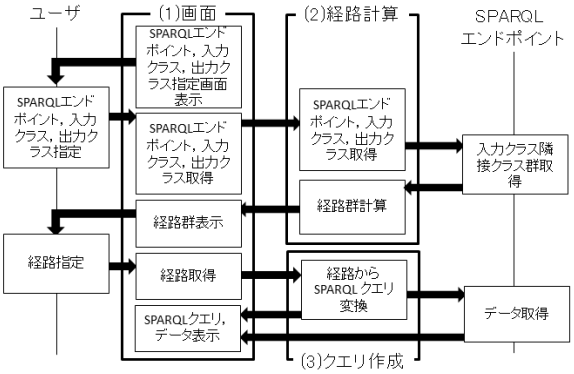

図 2: SPARQL Builder フロー図

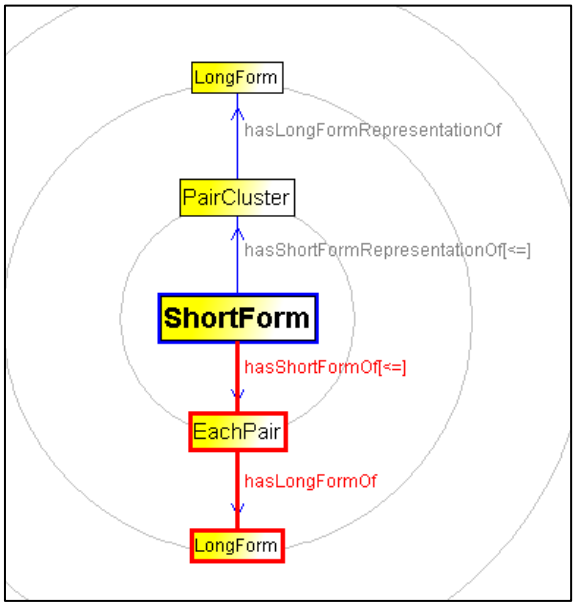

図 3: 経路選択画面

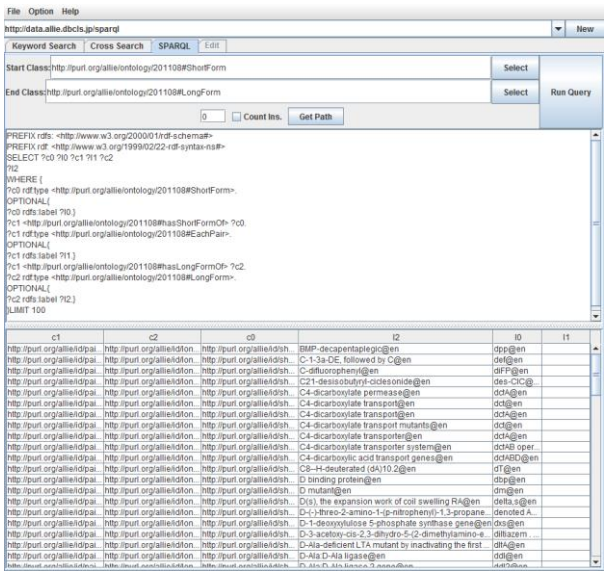

図 4: 検索結果表示画面

2.2 経路計算モジュール

経路計算モジュールでは，ユーザが指定する任意の SPARQL エンドポイントにおいて，入力クラスを始点とし，出力クラスを終点とし，クラス間のインスタンスをつなぐプロパティを辺とするグラフ内のクラス間経路を計算する．

任意の SPARQL エンドポイントをターゲットとするため，対象の SPARQL エンドポイントから経路計算に必要なデータを取得する必要がある．しかしながら，通常の経路探索問題と異なり，同じクラスにたどり着いた別の経路をカットすることができないため，データによっては経路数が爆発的に増えてしまう．そのため，現時点では，経路の長さの最大数を指定し(デフォルトは 3)，経路数の爆発的増大を抑え，その範囲内で出力クラスを終点とする経路を探索する．

2.2 経路表示画面

経路表示画面では，システムが提示した経路群からユーザが必要とするデータを表す一つの経路を選べるような表示をする必要がある．その一方で，全ての経路をリスト化すると重複した情報が多く，経路をユーザが細かく見比べて違いを見ながら選ぶ必要がある．そのため，現在は，経路群を樹状に表し，葉(出力クラスに対応)をユーザが選ぶことで，経路を指定できるようにした．樹状に表すことで，重複した情報はまとめられ，かつ，根から各葉への経路は一つであるため，経路を一意に指定することができる．

3. 考察

SPARQL Builder の試作に伴い, いくつかの問題点が明らかとなった．これらの問題点は，おおまかにシステム上の問題点とデータベース側の問題点に分けることができる．

システム上の最も大きな問題は経路数の爆発により，経路群表示までの時間がかかりがちであり，かつユーザが選ぶ際にも候補が多すぎて選びにくいという点である．たとえば，全てのクラスは rdfs:Class につながり, かつ, rdfs:Class はそれ自身が rdfs:Class のインスタンスであることより，クラス間グラフ上で rdfs:Class は非常に巨大な次数を持つ頂点となっている．そのため，素朴にクラス間経路を探索すると，一度 rdfs:Class を経由する経路が大量に出現することとなる. また, 次数が巨大な頂点については，経路数爆発の問題がおこるだけではない．巨大な次数を持つ頂点を通過した経路は，情報がその頂点を経由する際に失われるため，クエリとしては意味がないことが多い．そのため，適切な方法でこのような経路をユーザに提示する前に削除していく必要がある．この削除の基準をどう決めていくかは現在も検討中である．

データベース側の問題点としては，入力クラスの隣接クラス群取得の際，クラスが定義されていないインスタンスは探索できない，また，プロパティに rdfs:domain, rdfs:range が定義されていない場合，インスタンスを辿る必要があるため，隣接クラスの取得に時間がかかる，などがある．後者の問題点は今後の計算機やトリプルストアの性能向上，あるいは隣接クラス取得クエリの改良によって，改善の可能性があるものの，前者については，システム側で解決するのは非常に困難である．したがって，クラスが定義されていないインスタンスに対しては，現在は探索対象から外すしかなく，データはあるのに取得できない，あるいは経路はあるのに，途中で分断されてしまうことが起きてしまう状況になっている．この問題を解決するため，データベース探索法の研究を目下進めている．

4. まとめ

生物学研究者で SPARQL に習熟していないデータベースユーザを対象に，任意の SPARQL エンドポイントに対して，ユーザが求めるデータを取得できる SPARQL を記述する補助ツール SPARQL Builder を，TogoTable 上での利用に限定し，SPARQL の形式を制限したうえで試作した．

今後の課題としては，前節で述べた問題点，特に経路探索の問題点について改良をすすめていきたいと考えている．経路の枝刈りの基準として，情報量などの統計的基準と，実データを見てその傾向から示される基準の両方から検討をすすめる予定である．

また，クラス指定や経路指定の画面の設計について， SPARQL に不慣れなユーザにより使いやすいように変更を加えていきたいと考えている. 例えば，入力クラスと出力クラスの指定の際，何らかの基準でランキングを行って候補クラスを表示するなど，よりスムーズにユーザが欲しいデータの指定が行われるような画面システム設計が必要であると考えている．

参考文献：

[1] UniProt, http://www.uniprot.org/

[2] Redaschi, Nicole and Consortium UniProt: UniProt in RDF: Tackling Data Integration and Distributed Annotation with the Semantic Web. Nature Precedings, <http://dx.doi.org/10.1038/npre.2009.3193.1> (2009)

[3] RDF Platform < EMBL-EBI, http://www.ebi.ac.uk/rdf/

[4] Erick Antezana, Ward Blondé, Mikel Egaña, Alistair Rutherford, Robert Stevens,Bernard De Baets, Vladimir Mironov and Martin Kuiper: BioGateway: a semantic systems biology tool for the life sciences, BMC Bioinformatics, 10(Suppl 10):S11, (2009)

[5] TogoTable http://togotable.dbcls.jp/