SPARQL クエリ生成支援のためのクラス間関係提示とメタデータ設計

Class-class relationship suggestion and metadata design for assisting in writing a SPARQL query

山口敦子 [1]　小林紀郎 [2]　戀津魁 [2]　山本泰智 [1]　古崎晃司 [3]

Atsuko Yamaguchi [1], Norio Kobayashi [2], Kai Lenz [2], Yasunori Yamamoto [1] and Kouji Kozaki [3]

[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

[1]Database Center for Life Science, ROIS

[2] 理化学研究所 情報基盤センター

[2] Advanced Center for Computing and Communication, RIKEN

[3] 大阪大学 産業科学研究所

[3] The Institute of Scientific and Industrial Research, Osaka University

概要: .生命科学分野では，これまで数多くのデータベースが RDF 化されている．これらのデータベースは SPARQL エンドポイントと呼ばれる SPARQL 記述言語による検索が可能なウェブ API と共に提供されることが多い．多種多様な RDF データに対し，ユーザの興味に従って自由に SPARQL クエリを記述することで，RDF データの活用の可能性が大きく広がることが期待されるが，SPARQL クエリの記述は生物学者にとって技術的敷居が高い．この課題解決のために，著者らは SPARQL クエリ生成支援システム SPARQL Builder (http://sparqlbuilder.org/) を開発し，ウェブ上のサービスとして提供してきた．本発表では，SPARQL Builder を支える技術，特にクラス間関係の提示のためのクラス間パス計算とそれを可能にするメタデータ設計に焦点を当て紹介する．さらに，SPARQL Builder をより有用なサービスとするために，今後，必要と思われる技術についても議論する．

1. はじめに

1.1 背景

日々生産される多種多様かつ膨大なデータを統合的に扱うために，生命科学分野では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．同時期からサードパーティとして Bio2RDF が数多くのデータベースの RDF 化を進めている[3]．さらに 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，5 つのデータベースを RDF 化し公開した[4]．また，2014 年 1 月にはアメリカ最大の生命科学データベース機関である NCBI が運用管理する低分子化合物データベース PubChem が RDF 化され公開された[5].

RDF 化され SPARQL エンドポイントで公開される生命科学データベース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，研究者の要求に沿った SPARQL 記述が必要である．そのために，いくつかの SPARQL エンドポイントでは，典型的なクエリを例として予め用意しておく方法が取られている．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るため，予めすべての SPARQL クエリを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.

そこで，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 を構築し，2013 年 10 月からサービス運用を行ってきた[5,6]. 本発表では，SPARQL Builder を支える技術について，クラス間関係の提示のためのクラス間パス計算とそれを可能にするメタデータ設計を中心に紹介する．さらに，SPARQL Builder の今後の発展の方向性について議論し，その実現のために必要となる技術についても述べる．

1.2 SPARQL クエリ生成支援システム 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[5].

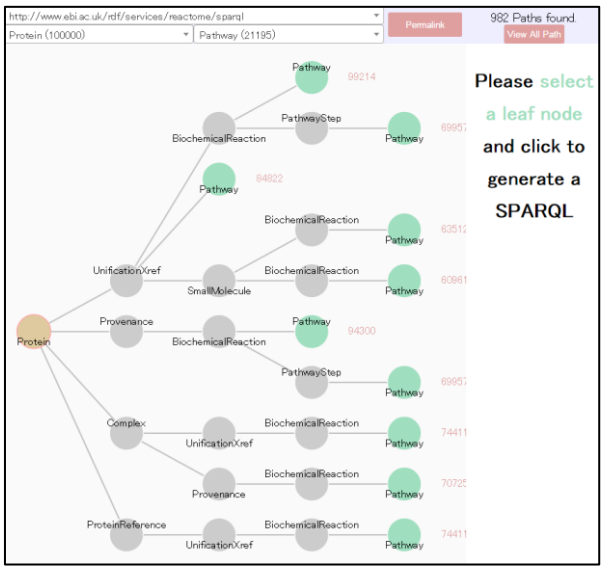

図 1: SPARQL Builder によるクラス間関係のパスの表示例

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．たとえば，ユーザがタンパク質のリストを持っており，それらと代謝経路の関係に興味がある場合は，入力クラスとして Protein，出力クラスとして Pathway を選ぶことになる．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．たとえば，代謝経路(Pathway)とその経路に登場するタンパク質(Protein)の関係は Protein -(left/right)- BiochemicalReaction -(pathwayComponnt)- Pathway という，BiochemicalReaction クラスを介した 2 段の関係で表される．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．図 1 は SPARQL Builder において，Protein-Pathway 間のクラス間関係を提示した例である．右端の頂点が Protein クラス，左側の緑の頂点が Pathway クラス，それらの頂点をつなぐパスがそれぞれクラス間関係を表す．ユーザがパスを一つ選び，選ばれたパスに対応する緑の頂点を押すと，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．生成されたクエリはそのまま検索へ利用したり，検索結果をダウンロードしたりできる．SPARQL の知識が多少でもあれば，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．

図 2 は SPARQL Builder のシステム構成の概要である．事前に対象となる SPARQL エンドポイントからクラスのリストやクラス間関係，さらに，インスタンス数やトリプル数などの統計情報など，必要なメタデータを取得し格納しておく(1)．ユーザが GUI からシステムにアクセスすると，メタデータからクラスのリストが取り出され(2)，ウェブ API を通じて(3)，GUI 上に提示される．ユーザがクラスのリストから入出力クラスを選択すると，メタデータから作られたクラス間関係を表すグラフであるクラスグラフを用いてクラス間パスが計算され(4)，クラス間パスのリストが GUI 上に提示される．ユーザがパスをひとつ選ぶと，パスから生成した SPARQL クエリが GUI 上に提示される．

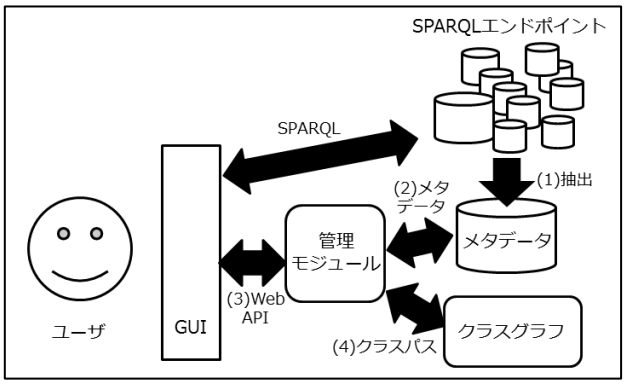

図 2: SPARQL Builder システム概要

ここで，システムの鍵となるのは，クラス間関係を管理しパスを計算するためのクラスグラフの構築およびクラスグラフ構築に必要なメタデータの設計の二点が挙げられる．先述したように，ユーザが指定する任意の 2 つのクラスに対し，入出力クラス間のパスを待たせない時間で計算する必要があり，そのために，入出力クラスに直接関係をもつクラスのみを扱うのではなく，対象となるデータセット全体のクラス間関係の構造を扱う必要がある．どのようにクラスグラフを定義し，構築するか，さらにどのようにクラス間パスを計算するかによって，大きく計算効率が異なる．また，ユーザにデータセットやクラスの情報を提示するデータ，および，クラスグラフを構築するために必要なデータを事前に取得し蓄積しておくために，メタデータの設計は重要である. そこで, 第 2 節ではクラスグラフ構築について，第 3 節ではメタデータ設計についてより詳しく述べる．

2. クラスグラフ構築とクラスパス

2.1 クラスグラフ

クラス間関係を効率的に取り扱うために，特にクラス間パスの計算や提示のため，SPARQL Builder では各データセットに対してクラスグラフを構築する．クラスグラフとは，クラスを頂点，プロパティを辺とするラベル付き有向グラフである．

ここで，より厳密にクラスグラフを定義する．R を RDF データセットとし, C を R に含まれるクラスの集合, P を R に含まれるプロパティの集合とする．このとき，R に対するクラスグラフはラベル付き有向グラフ $G_R = (V, E, c, p)$である．ただし，V は大きさ$|C|$の頂点集合，c は V から C への一対一関数である．E は $V \times V$ 上の多重集合であり，p は E から P への関数であり，R から次のように構成される: 2 つのクラス $class_d$, $class_r$，およびプロパティ $prop$ について，(条件 1) 2 つのトリプル($prop$, rdfs:domain, $class_d$), ($prop$, rdfs:range, $class_r$)が R に含まれる，(条件 2) 3 つのトリプル(s, $prop$, o), (s rdf:type $class_d$), (o rdf:type $class_r$)が R に含まれる，のいずれかが成り立つとき，またそのときに限って，頂点 $v = c^{-1}(class_d)$ から $u = c^{-1}(class_d)$の辺 e_{prop} を E に加え, $p(e_{prop}) = prop$ とする．

クラスグラフ G_R に対し，頂点 v_{start} から頂点 v_{end} へのクラスパスとは(n_0, e_1, n_1, …, n_k)，ただし，$n_i \in V$, $e_i \in E$, $n_0 = v_{start}$, $n_k = v_{end}$, $n_i \neq v_{end}$ ($i \neq k$)とする．クラスパス(n_0, e_1, n_1, …, n_k)に対し, k をパスの長さとよぶ．

クラスパスから SPARQL への変換は, パスに含まれる各辺 e_i について，WHERE 節へトリプルパターン(?s $p(e_i)$?o)を加えることで構成する. 辺 e_i が条件 2 によって辺集合 E に加えられた場合には，さらに，2 つのトリプルパターン(?s rdf:type $c(n_{i-1})$), (?o rdf:type $c(n_i)$)を加えてクラスの制限を行う．

2.2 無向クラスグラフとクラスパス探索

ユーザが望むクラス間関係をパスによって提示できる可能性を高くするためには，クラス間パスをできるだけ多く計算することが望ましい．しかしながら，通常のパス探索問題と違い，クラスグラフは多重辺グラフであり，さらに求めるパスもシンプルパスではないため，パスの長さを増やすにつれてパス数が爆発的に増大する．クラスグラフ上で幅優先探索を行うという素朴な手法をサービス初期には取っていたが，その手法では，ある程度以上の大きさのデータセットに対しては，長さ 3 がユーザと GUI を通してやりとりするシステムとしては限界であった. たとえば，Bio2RDF の NCBI-gene のデータに対し，各クラス間関係の長さ 3 のクラスパスを全て計算する平均時間は 3ms であるが, 長さ 4 では 129ms と大幅に増加し，長さ 5 では実験の環境ではメモリ不足で計算ができないクラス間関係が存在した[8].

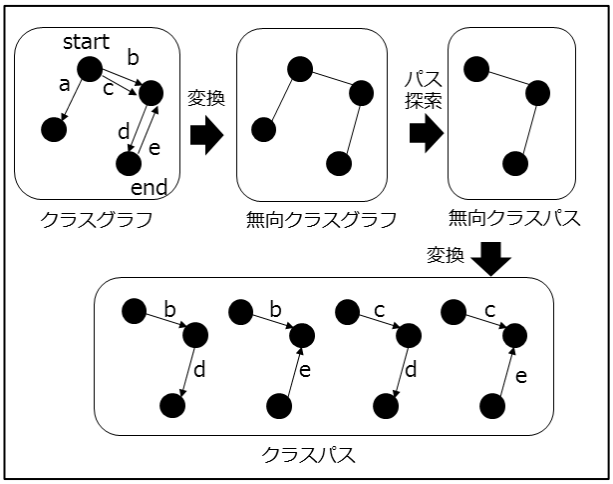

図 3: 改良後のパス探索手法

そこで，以下のようにアルゴリズムを改良した. まず, クラスグラフを無向でシンプルなグラフ(無向クラスグラフとよぶ)へ, 辺の方向とラベルを取り払い，多重辺は一つの辺へと束ねられたものとする．具体的にはクラスグラフ $G_R = (V, E, c, p)$に対し，$G_R' = (V, E')$，ただし，E' = { {u,v}| (u,v)∈E あるいは(v,u) ∈E }である．次に，G_R'上で幅優先である長さ以下

のパスを全探索する．探索の結果得られたラベルなしのパスに対し，各辺を多重辺に戻す．その際，多重辺のすべての組み合わせをパスに適用する．

この手法を使うことにより，パス探索の計算時間は大きく改良され，その結果，より多くのクラスパスを提示することが可能となった．例えば，この手法を適用した場合，先述の NCBI-gene データにおいて，各クラス間関係の長さ 4 のパス全ては平均 5ms で，長さ 5 のパス全ては平均 168ms で計算が可能である[8].

3. メタデータ設計と利用

3.1 SPARQL Builder Metadata

クラス間関係の提示のためには，クラスグラフ構築に必要な情報を SPARQL エンドポイントから取り出す必要がある．サービス開始時は必要な情報を必要なだけ動的に SPARQL エンドポイントから取り出すことも検討したが，現実的な時間でクラスグラフを構築するためには事前に抽出し蓄積する方法が妥当であるという結論に至った．

そのため，クラスグラフ構築に必要なデータを事前に SPARQL エンドポイントに過剰な負担をかけずに取得できることが望ましい．その目的のもと，取得すべきメタデータを洗い上げてスキーマを設計し，さらにそれらのメタデータを取得するための SPARQL 文を定義した．

設計したメタデータのスキーマ仕様(SPARQL Builder Metadata)について，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，[9]を参照されたい．

3.2 メタデータ抽出

設計した SPARQL Builder Metadata および定義した SPARQL 文に沿って，SPARQL エンドポイントからメタデータを抽出した．現在，EBI や Bio2RDF のエンドポイントを中心に，38 のエンドポイントからメタデータを抽出し蓄積して利用している．一方で，UniProt 等の巨大な RDF データに関しては，SPARQL エンドポイントからメタデータを取得することに成功しておらず，次節で議論するようなメタデータ取得方法の検討が必要な状況となっている．

4. 今後の展望と考察

生命科学データのさらなる統合へ向けて，フェデレート検索への対応は必要性が高いと思われる．幸いなことに，クラスパスから SPARQL へ変換することを基本とする SPARQL Builder は，現在，データセットごとにもっているクラスグラフを，すべてのデータセットに拡張することで原理的には自然にフェデレート検索に対応可能だと考えられる．しかしながら，クラスグラフが大規模化し，クラスパスの数も現状よりさらに爆発的に増えることが予想されるため，クラスパス探索手法の改良はもちろん，クラスパスのランキング手法等で数多くのクラスパスをどのように見せるかという検討と開発が必要になるであろう．

また，前節の最後に述べたメタデータ抽出に関しても，データのダウンロードを併用するなど SPARQL エンドポイントに負担をかけすぎない手法を考えていく必要がある．また，メタデータをプロバイダ側でできるだけ用意してもらう方向性も考える必要があるかもしれない．例えば，クラウド上のトリプルストアのサービス Dydra[10]では，データをアップロードすることにより自動的に SPARQL Builder Metadata が生成され，HTTP GET で取得可能となっているため，全く SPARQL エンドポイントに負担をかけずにメタデータを抽出できる．

5. まとめ

SPARQL に習熟していないユーザを対象に，SPARQL 記述を支援する SPARQL Builder を開発し，サービス運用をするにあたり，必要となった技術開発について述べた．特に，メタデータの事前取得蓄積とクラスグラフ構築およびパス探索方法の改善により，ユーザが待てる程度の時間でクラスリストやクラス間パスの提示が可能となった．

今後の課題としては，前節で議論したクラスパスのランキングやメタデータ取得方法改良の改良，そして，複数の SPARQL エンドポイントを利用したフェデレート検索についても対応していきたい．

参考文献：

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

[2] Redaschi, N. 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] 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)

[4] 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)

[5] Fu, G., Batchelor, C., Dumontier, M., Hastings, J., Willighagen E., and Bolton, E.: PubChemRDF: towards the semantic annotation of PubChem compound and substance databases. Journal of Cheminformatics, 7(34), doi:10.1186/s13321-015-0084-4 (2015)

[6] SPARQL Builder, http://sparqlbuilder.org/

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

[8] Yamaguchi, A., Kozaki, K., Lenz, K., Wu, H., Yamamoto Y., and Kobayashi, N.: Efficiently Finding Paths Between Classes to Build a SPARQL Query for Life-science Databases, 5th Joint International Semantic Technology (JIST2015) Conference, to appear in LNCS, Yichang, China

[9] SPARQL Builder Metadata (Version Sep. 2015), http://www.sparqlbuilder.org/doc/sbm_2015sep/

[10] Dydra, http://dydra.com/