

## RDF 検索言語:xRQL の提案と実装

角野 宏光<sup>1</sup> 小俣 栄治<sup>1</sup> 石川 憲洋<sup>1</sup> ヨハン・イエルク<sup>2</sup> 朱 仲武<sup>3</sup> 宮津 和弘<sup>3</sup>

<sup>1</sup>株式会社 NTT ドコモ <sup>2</sup>エリクソンリサーチ <sup>3</sup>日本エリクソン株式会社

**あらまし:** セマンティックウェブで利用されるリソース情報の記述言語として、W3C により RDF が標準化されているが、RDF で記述されたデータに対する検索言語はまだ標準化されていない。現状 RDF の検索言語として、RQL や SquishQL などいくつかの検索言語が提案されているが、これらの言語はデータベースの操作機能が限られており、また XML のシンタックスを用いていないため、XML の汎用的なツールを利用することができない。本稿ではこれらの課題を解決するために我々が設計した新たな検索言語 xRQL (XML-based RDF Query Language) について紹介する。xRQL は既存の RDF の検索言語で不足していたデータベースの操作、検索結果の設定機能等を備えた XML 形式の言語である。また本稿では、我々の実装した xRQL データベースエンジンについても紹介する。

### 1. はじめに

セマンティックウェブを実現するためには、ウェブ上のリソースにマシンで処理可能な意味情報をメタデータとして付与することが基本となる。W3C ではこのようなリソースの記述形式として RDF (Resource Description Framework) [1,2]を標準化しているが、RDF を普及させるためには、RDF で記述されたリソースに対して標準的に検索を行う検索言語の確立も非常に重要となる。現状 RDF で記述されたデータに対する検索言語もいくつか提案されているが[3,4,5,15,16,21]、以下に示すような課題があり、標準として用いるのに十分な機能や構造を備えた検索言語はまだ存在しない。

- 非 XML 形式

既存の RDF 検索言語は SQL と同様なシンタックス

を用いているため、言語の処理を行うために XML パーサー以外の特別なパーサーが必要となる。また、階層的な構成をとる複雑な RDF のデータモデルを検索言語にて表現する場合、XML を用いたシンタックスに比べて SQL のようなプレーンなテキスト形式では表現が限られている。

- データ操作機能の不足

既存の RDF 検索言語は RDF メタデータの検索の機能はサポートしているが、RDF メタデータの作成、変更、削除等の機能をサポートしていない。リレーショナルデータベースで用いられる SQL などではこのような機能はサポートされており、データが分散環境する環境においては、共通に利用可能なデータ操作の機能が必要である。

- 検索結果の形式

既存の RDF 検索言語において得られる検索結果は結果の値が羅列されているだけであり、検索結果を利用するためには、結果を取得した後にさらなる処理が必要となる。検索の問合せにて検索結果の形式が明確に宣言されることにより、より効率的に処理することが可能である。

上記の問題を解決するために、我々は XML 形式で記述する RDF 検索言語 xRQL を提案する。この言語は RDF のグラフのデータモデルに基づく階層的なデータ表現が可能となるように設計した。また、xRQL は XML の検索言語[8]、オブジェクト指向の検索言語[7,11,14,15]、

---

1 株式会社 NTT ドコモ マルチメディア研究所

NTT DoCoMo, Inc., Multimedia Laboratories  
3-5, Hikarinooka, Yokosuka, Kanagawa, 239-8536, Japan  
{sumino, omata\_e, ishikawa}@mml.yrp.nttdocomo.co.jp

2 Ericsson Research

Torshamnsgatan 23, Kista SE-16480 Stockholm, Sweden  
Johan.Hjelm@era.ericsson.se

3 日本エリクソン株式会社

Nippon Ericsson K.K.  
Koraku Mori Building 1-4-14 Koraku, Bunkyo-ku Tokyo 112-0004, Japan  
{Zhong-Wu.Zhu, kazuhiko.miyatsu}@ericsson.com

グラフに対する検索言語[18,20,26]等の機能を用い、汎用かつ機能的なRDF検索言語の実現を目指すものである。

本稿では、まず2章でRDF検索言語の要求条件をあげ、3章にて我々が設計したxRQLの構成とシンタックスについて、いくつかの具体例をあげて説明する。4章では本研究において実装したxRQLを用いたデータベースエンジンについて簡単に紹介し、5章でRQL検索言語に関する現状の技術動向について紹介する。最後に、6章にて本研究のまとめおよび今後の研究について述べる。

## 2. RDF 検索言語の要求条件

以下に我々の検討したRDF検索言語に対する要求条件を述べる。

- マシン可読性

分散された様々なメタデータを簡単かつ効率的に扱うために、人間にとって読みやすくマシンにおいても処理しやすい形式が求められる。

- RDF データモデルへの対応

RDF検索言語はRDFのデータモデル(リソース、プロパティ、値)に対応する必要がある。また、RDF Schema で定義されるオブジェクト指向のモデルと同様のクラスやプロパティの階層化、継承の概念に対応することが望まれる。

- スキーマ検索機能

分散されたRDFのメタデータを様々なアプリケーションで利用するためには、RDFのデータ構造を定義するスキーマについて把握する必要がある。そこで、RDFの検索言語はRDFのメタデータのスキーマに関しても同様に検索可能とする必要がある。これらのスキーマはRDF Schemaを用いて定義され、検索のクエリで用いられるプロパティやクラス間の関連についての情報を与える。

- 推論

RDFスキーマで定義される語彙の階層化に対応した処理を行うために、検索言語においてクラスやプロパティの関係を推論して処理する機能を含むことが望ましい。

- RDF メタデータの操作

基本的なRDFメタデータの検索機能に加え、検索言語はRDFメタデータの作成、変更、削除の機能をサポートする必要がある。これらの操作はRDFのメタデータ情報を作成したいユーザにとっては有効な機能であるといえるが、現状の検索言語ではサポートされていない。

- 検索結果フォーマット定義

RDFの検索結果を取得したユーザにとって、検索結果を簡単に処理するために、ユーザにとって利用しやすい形式を定義できるよう検索結果のフォーマットの指定する機能が要求される。

## 3. xRQL

xRQLは機能を用いた論理的な言語である。xRQLは操作の宣言、RDFデータに関する記述、結果のフォーマットの記述より構成される。RDFのグラフ形式のデータモデルに基づき、xRQLはRDFのデータを記述する方法として、GOQL[23]と同様の見やすい形式でパスを表現する。データ操作を行うためにオブジェクト指向のデータモデル[8]に対応した作成、変更、削除の機能を適用する。また、XQuery[7]と同様の手法を用い、結果の出力形式をユーザの望むXMLの形式で定義することが可能である。さらに、RDFスキーマのプロパティやクラスの階層化を考慮した一連のRDF操作の機能を提供する。

### 3.1. 設計方針

あるコンテンツに関連するRDFリソースの構成例を図1に示す。

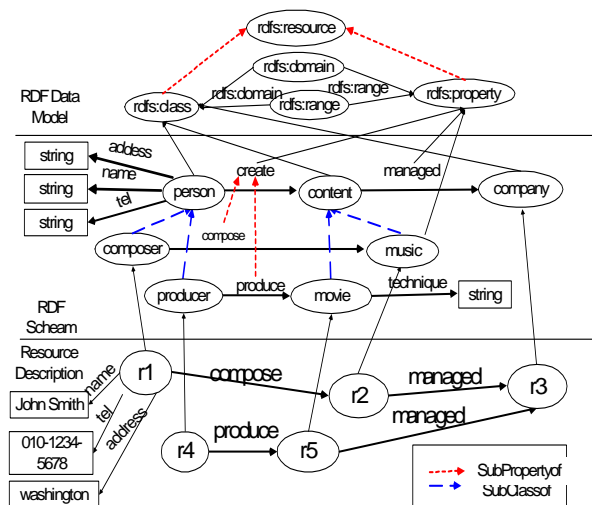


図1: RDF リソースの構成例

図1の例を用い、以下にxRQLの設計方針について述べる。図1の最下層は、2人(r1, r4)によって作成された2つのコンテンツ(r2, r5)についての記述である。中間の層はコンテンツの各オブジェクトが対応するクラスを定義する部分である。本スキーマにおいて、create プロパティはドメインがperson、レンジがcontentのような関係が定義される。最上層のグラフはRDFのデータモデルを示している。

図1の構成により、クラスとプロパティの階層的な意味を含んだリソース間の関連やリソースの属性が表現される。例えば、music (又は movie)クラスは content クラスの subclass として定義され、compose(又は produce)プロパティは create の subproperty である。

### 3.1.1. 階層化グラフモデル

図2に示すように、我々は RDF メタデータのアーキテクチャを RDF データのリソースのレベル、スキーマのレベル、データモデルのレベルの3つの階層に分けて扱うこととした。

RDF のリソースレベルはリソースとリテラルをノードとし、ノード間のプロパティをラベルで記述したラベル付き有向グラフで表現される。また、RDF のスキーマレベルは、person クラスや create プロパティのように、ノードやラベルを定義するものである。RDF のデータモデルのレベルでは、rdfs:Class や rdfs:Property のような RDF schema の定義により構成される。それぞれのレベルの各要素は、その上のレベルのインスタンスとして与えられる。

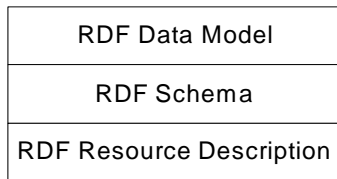


図2：階層化データモデル

### 3.1.2. RDF リソースデータの操作

xRQL の機能として、我々は 3.1.1 節のデータモデルにおける RDF リソースデータを扱うための4つの基本的な操作（検索、削除、作成、変更）について定義した。

xRQL では RDF のリソースを定義しているスキーマに関する知識を用いて検索のクエリを記述することとする。検索の操作において、まず我々は RDF のデータモデルに従ったパス表現を定義した。RDF のグラフ構造に対応し、さらに人が理解可能なように、パスの表現はグラフにおけるノードと矢印を順番に並べる形式とした。これはオブジェクト指向の検索言語 GOQL と同様の表現である。以下に典型的なパスの表現を示す。

**L = C1 {p1} C2 {p2} C3 ...**

ここで、p1, ..., pn はプロパティであり、C1, ..., Cn は

クラスである。これらは RDF データに対応するスキーマで定義されるものである。一つのデータのパスはノードと矢印のシーケンスである。ここで、Ci はクラスであり、Ci と Ci+1 の間にプロパティ Pi がある。

また、スキーマで表現されたパスにおいて、関連する RDF リソースを結びつけるために、パス中にリソースに対応した変数を用いる。一例として、図1のリソースにおいて、あるコンテンツ(content) を作成(create) した作者(person) のリソースを x とするとパスは以下のように表現される。

**person:x{create}content**

ここで、変数 x は person クラスのインスタンスであることを示す。このパス表現を基に x に対応するリソースを検索する場合、もし person クラスの下に定義される subclass があれば、その subclass に属するリソースも取得可能とする。例えば、図1に示す RDF スキーマによれば、composer と producer の subclass に属するインスタンスについても person クラスのインスタンスとして取得可能である。

RDF のメタデータ操作（作成、削除、変更の操作）については、OQL[7]が用いる手法と同様に、オブジェクト指向のコンセプトを用いて定義する。スキーマで定義されるクラスにリソースは属し、リソースは唯一の ID (URI) で識別されるオブジェクトである。図1の例では、r1 は name, tel, address, compose というプロパティをもつオブジェクトである。name, tel, address の各プロパティの値は文字列である。一方 compose プロパティの値は別のオブジェクト r2 を示している。r1 は composer クラスに属し、r2 は図1の中間層にて定義される RDF スキーマ music クラスに属すオブジェクトである。

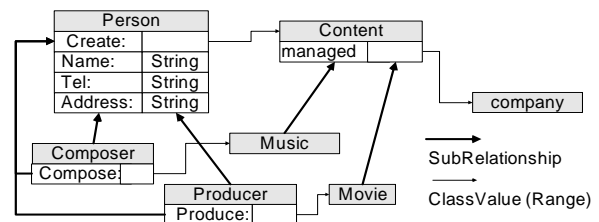


図3：オブジェクト指向のスキーマ

削除の操作は RDF のレポジトリから RDF のリソースを取り除くために用いる。この操作により RDF のリソースそのものだけでなく、リソースに関連付けられたプロパティを含むステートメントも取り除かれる。例えば、図1の例において、r3 のリソースを削除する

場合はリソース r3 と r3 についての全ての関連が削除され、r2 と r5 のプロパティ"managed"の値は Null となる。

作成の操作はRDF レポジトリ内に新たにRDFのリソースを作成する場合に用いる。図3に示すように、それぞれのRDF リソースはプロパティとプロパティ値で構成される一つのオブジェクトとして表現される。プロパティ値は他のオブジェクトもしくはリテラルである。RDF リソースが継承するプロパティはその parent class で定義される。例えば、composer オブジェクト parent クラスである person クラスより name, tel, address というプロパティを継承する。

変更の操作はプロパティを変更するために用いられ、プロパティ値として新たにリテラルまたは関連するリソースを与える。図4に示すように、この操作は別々に定義されたRDF グラフのリソースをプロパティにより関連付けする場合にも用いられる。

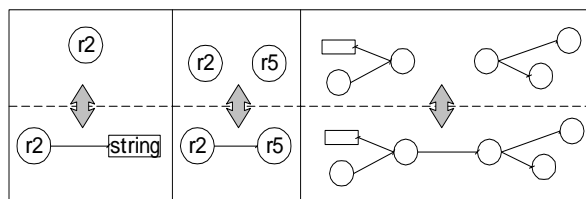


図4：オブジェクト指向のスキーマ

### 3.1.3. 検索言語のシンタックス

W3C の標準化により XML[22]は機械により処理可能な言語として広く普及している。RDF に関して、データモデルを記述するために XML 形式の記述を用いているため、機械で処理する際には XML パーサーは必要なツールである。従来の XML 形式でない RDF の検索言語では、言語を処理するための特別なパーサーが必要であったが、xRQL では検索言語の解析に必要な処理を減らすために、RDF のリソース記述と同様の XML 形式で RDF の検索言語を定義した。

### 3.1.4. 結果のフォーマット定義

既存の RDF の検索言語は検索結果のフォーマットについて考慮されておらず、結果を取得し自動的に処理するためには、結果を機械で処理しやすい形式に変換するツールを用意する必要があった。そこで、xRQL では XQuery で用いられている手法と同様に、XML 形式を基にしたユーザの望む結果のフォーマットをユーザが自由に定義可能となるように設計した。例えば図1のデータにおいて、パス表現

person:x{create}content における変数 x の値を結果として取得する場合について述べる。結果である x 値が <personinfo>要素の値に記述された形式で検索結果を取得したい場合には、以下のように記述する。

<result>

<personinfo> x </personinfo>

</result>

結果として複数値が得られた場合には、結果の入った <personinfo>要素が複数返される。結果を xHTML 形式で表示した場合や、他のフォーマットに変換してその先のプロセスで利用する場合などにも、XSLT などを用いて簡単かつ効率的に変換することが可能である。

## 3.2. xRQL シンタックス定義

xRQL は XML 形式により定義され、以下の3つの部分により構成される。最初の部分はユーザの要求する操作命令である。次の部分はオブジェクト指向に基づくパスの表現と検索の条件式が記述される部分である。最後の部分は検索の問合せを行うユーザが受け取りたい検索結果のフォーマットを指定する部分である。

### 3.2.1. 操作命令

xRQL では操作を示す要素が必ず1つ宣言される。我々は xRQL データを操作する命令として、検索、作成、変更、削除の4つの命令を定義した。RDF のリソースに対してはこれら4つの操作により問合せや変更が行われ、スキーマに対しては検索の操作が用いられる。

### 3.2.2. RDF データ記述

操作対象となる RDF のデータの記述には、object, objectvalue, location, condition, namespace の各要素が含まれる。

#### • object 要素、objectvalue 要素

3.1.2 節で述べた設計方針に基づき、我々は RDF リソースを記述するために2つの要素を定義した。object 要素は対象となる RDF リソースのクラスを指定するために用いられる。objectvalue 要素は object 要素で指定されたクラスに含まれるリソースに対してユニークな ID とプロパティの値を付与するために用いられる。これらのエレメントがどのように用いられるかは図10の例に示す。

#### • location 要素

location 要素はパスの表現を記述するための要素である。パスの記述においては、3.1.2 節で述べたよ

うに、RDF のスキーマのクラスとプロパティを並べることでグラフデータモデルを表現する。

- **condition 要素**  
condition 要素は location 要素内のパス表現の部分に定義された RDF のリソースに対して、検索、変更、削除の操作における条件を指定するために用いられる。
- **namespace 要素**  
複数のスキーマに対応するため、namespace 要素を定義した。URI で記述される namespace は長いいため、namespace 要素を用いて記述を省略することが可能である。また、異なるスキーマにて利用される同じ名前のクラスやプロパティに対して、名前の重複を解決するために namespace 要素が用いられる。

### 3.2.3. 結果フォーマット定義

検索の操作では、結果のフォーマットを定義する機能が提供される。フォーマットは XML 形式に基づいているため、ユーザは検索結果を取得した後の処理で用いるための好みのフォーマットを取得することができる。

## 3.3. RDF スキーマ操作

xRQL は RDF のリソースに関する操作に加え、スキーマの検索にも対応する。

### 3.3.1. スキーマ変数とパス表現

RDF データのリソースとして用いられる変数と、スキーマで用いられる変数を識別するために、xRQL ではスキーマの変数に対してはプレフィックス"\$"を付与することとした。スキーマの変数を用いたパスの表現は以下のように記述される。

**<location>\$x{managed}\$y</location>**

3.1.1 節で述べたような階層化されたデータモデルでは、RDF のスキーマレベルは RDF データモデルレベルのインスタンスとして表現される。ゆえに、このパス表現における \$x および \$y は `rdfs:Class` に属し、このプロパティに対して \$x はドメイン、\$y はレンジに相当する。

### 3.3.2. 操作機能

xRQL は RDF のスキーマで定義されたクラスやプロパティの関係を検索するために、以下の機能を提供する。

- **subclassof**  
subclassof 機能は所望のクラスのサブクラスを取

得するために用いられる。例えば、`subclassof(person)` という記述では `person` クラスのサブクラスである `composer` クラスと `producer` クラス、さらに `person` クラス自身を結果として返す。

- **subpropertyof**  
subpropertyof 機能はあるプロパティのサブプロパティを取得するために用いられる。例えば、`subpropertyof(create)` は `compose` プロパティ、`produce` プロパティおよびそれ自身である `create` プロパティを返す。
- **domain および range**  
これらの2つの機能はあるプロパティのドメインとレンジを取得するために用いられる。例えば、`create` プロパティのドメインとレンジを取得したい場合には、それぞれ `domain(create)`、`range(create)` と記述する。
- **typeof**  
typeof はある RDF のリソースまたはプロパティの属する class や property を取得するための定義である。例えば、図 1 のデータでは `typeof(r1)` により `composer class` が返される。

## 3.4. xRQL の記述例

本節では xRQL の操作で用いられるいくつかの命令に関してそれぞれ記述例を示す。

### 3.4.1. 検索

図 5 は、`company` により管理される ( `managed` ) `content` の作成者 ( `creator` ) に関して問い合わせる例である。

```
<xrql:xrql>
<xrql:operator>select</xrql:operator>
  <xrql:location>person:x{create}content{managed}company</xrql:location>
<xrql:result>
  <description>x</description>
</xrql:result>
<xrql:usingnamespace>
  <xrql:item>ns1="www.docomo.co.jp/DE/metaschema"</xrql:item1>
</xrql:usingnamespace>
</xrql:xrql>
```

図 5 : 検索操作例

location 要素では、3つのクラス ( `person`、`content`、`company` ) と2つのプロパティ ( `create`、`managed` ) を用

いてパスを表現し、変数 *x* が **person** クラスのインスタンスを示している。この例では、条件を与える **condition** 要素を用いていないので、パス表現で示された条件に適合するリソース *x* が結果として得られる。**result** 要素において結果出力のフォーマットを定義しており、図 6 が本検索における結果である。

```
< xrql:result>
<description> "www.person1.co.jp" </description>
<description> "www.person2.co.jp" </description>
<description> "www.person3.co.jp" </description>
</ xrql:result>
```

図 6：検索操作例(図5)の結果

いくつかのパス表現と条件を組み合わせることで、RDF レポジトリに対するより複雑なクエリを記述することが可能である。例として、図 7 に "John Smith" という名前を持つ人の電話番号を問い合わせる場合のクエリを示す。本例では、この問合せを 3 つの変数と 2 つのパスを用いて表現する。2 つのパス表現はどちらも同じ **person** クラスの変数 *x* から始まり、同一のリソースに関する 2 つのプロパティについて条件を与えている。**condition** 要素はデータマッチングにて用いられる条件を与える。また、結果のフォーマットはそれぞれ 3 つの変数に対して異なるタグを与えるように設定されている。図 8 には検索結果の例を示す。

```
< xrql:xrql>
< xrql:operator>select</ xrql:operator>
< xrql:location>
  person:x{name}STRING;y{person{tel}STRING;z
</xrql:location>
< xrql:condition>
  < xrql:item>y="john smith" </ xrql:item>
</ xrql:condition>
< xrql:result>
  <resourcedescription> x <resourcedescription>
    <name>y</name>
    <tel>z</tel>
</ xrql:result>
< xrql:usingnamespace>
  < xrql:item>ns1="www.docomo.co.jp/DE/metaschema"</
xrql:item1>
</ xrql:usingnamespace>
</ xrql:xrql>
```

図 7：複雑な検索操作の例

```
< xrql:result>
<resourcedescription>"www.person.co.jp/person1"
</resourcedescription>
<name> john smith</name>
<tel> 010-1234-5678 </tel>
</ xrql:result>
```

図 8：検索操作例(図7)の結果

### 3.4.2. 削除

図 9 に削除の操作の例を示す。本例では、RDF レポジトリより **company** クラスのインスタンスである "www.company1.co.jp" を削除する場合の例を示す。3.1.2 節で述べたように、この **delete** の操作では **condition** 要素に記述された **company** クラスのオブジェクトが削除される。

```
< xrql:xrql>
< xrql:operator>delete</ xrql:operator>
< xrql:object>company</ xrql:object>
< xrql:location>company: x</ xrql:location>
< xrql:condition>
  < xrql:item>x="www.company1.co.jp"</ xrql:item>
</ xrql:condition>
< xrql:usingnamespace>
  < xrql:item>ns1="www.docomo.co.jp/DE/metaschema"</
xrql:item1>
</ xrql:usingnamespace>
</ xrql:xrql>
```

図 9：削除操作の例

### 3.4.3. 作成

図 10 に示す例は、**person** クラスのリソース "www.person4.co.jp" を作成し、その名前(name プロパティ)が "Rechard Lim"、電話番号(tel プロパティ)が "010-1122-3344"、作成したコンテンツ(create プロパティ)が "www.content.co.jp/content1" であるという情報を作成する例である。3.1.2 節および 3.2.2 節で述べたように、この操作は **person** クラスに属するオブジェクトに対する操作として表現される。**objectvalue** 要素内で作成対象のオブジェクトのリソース "www.person4.co.jp" を **resource** 要素で記述し、このリソースに関するプロパティとその値を **propertyvalue** 要素にて記述する。

```

< xrql:xrql>
< xrql:operator>create</ xrql:operator>
< xrql:object>person</ xrql:object>
< xrql:objectvalue>
  < xrql:objectitem>
    < xrql:resource>"www.person4.co.jp"</xrql:resource>
    < xrql:propertyvalue>
      < xrql:item>name = "Richard Lim" </xrql:item>
      < xrql:item>tel = "010-1122-3344" </xrql:item>
      < xrql:item>create = "www.content.co.jp/content1"
    </xrql:item>
    </xrql:propertyvalue>
  </xrql:objectitem>
</xrql:objectvalue>
< xrql:usingnamespace>
  < xrql:item>ns1="www.docomo.co.jp/DE/metaschema"</
xrql:item1>
</xrql:usingnamespace>
</xrql:xrql>

```

図 10：作成操作の例

#### 3.4.4. 変更

図 11 は"www.content1.co.jp"のリソースを  
"www.company1.co.jp"へ変更する例を示す。

```

< xrql:xrql>
< xrql:operator>update</ xrql:operator>
< xrql:object>content</ xrql:object>
< xrql:location>content:x</ xrql:location>
< xrql:condition>
  < xrql:item>x="www.content1.co.jp"</xrql:item>
</xrql:condition>
< xrql:objectvalue>
  < xrql:objectitem>
    < xrql:propertyvalue>
      < xrql:item>managed = "www.company1.co.jp" </
xrql:item>
    </xrql:propertyvalue>
  </xrql:objectitem>
</xrql:objectvalue>
< xrql:usingnamespace>
  < xrql:item>ns1="www.docomo.co.jp/DE/metaschema"</
xrql:item1>
</xrql:usingnamespace>
</xrql:xrql>

```

図 11：変更操作の例

#### 3.4.5. スキーマの検索

スキーマ問合せの例として、図 12 に create プロパティのドメインとレンジの問合せ例を示す。location 要素で RDF スキーマに基づくパス表現はプロパティ "create" と 2 つのクラスの変数 \$x1 と \$y1 が宣言される。あるオブジェクト指向のデータモデルでプロパティはクラスのサブクラスとして継承が可能である。そこで、この操作では create プロパティのドメインのサブクラスも検索対象として含む。図 13 は図 12 の操作の結

果を示すものである。

```

< xrql:xrql>
< xrql:operator>select</ xrql:operator>
< xrql:location> $x1{create}$y1</ xrql:location>
< xrql:result>
  < domain>$x1</domain>
  < range>$y1</range>
</xrql:result>
</xrql:xrql>

```

図 12：スキーマ検索操作の例

```

< xrql:result>
  < domain>person</domain>
  < range>content</range>
  < domain>person</domain>
  < range>music</range>
  < domain>person</domain>
  < range>movie</range>
  < domain> composer </domain>
  < range>content</range>
  < domain> composer </domain>
  < range>music</range>
  < domain> composer </domain>
  < range>movie</range>
  < domain> producer </domain>
  < range>content</range>
  < domain> producer </domain>
  < range>music</range>
  < domain> producer </domain>
  < range>movie</range>
</xrql:result>

```

図 13：スキーマ検索操作例（図 12）の結果

#### 4. 実装

我々はネイティブの RDF データベース上に xRQL のプロセッサを実装した。現状他のほとんどの RDF データベースはリレーショナルデータベース上で実装しているが、本システムでは RDF のデータを text ファイルとして保存している。本システムの構成を図 14 に示す。RDF データベースは RDF パーサー、API、グラフ照合処理、結果出力処理とストレージにより構成される。我々の xRQL システムは本データベースの API を用いて実装した。



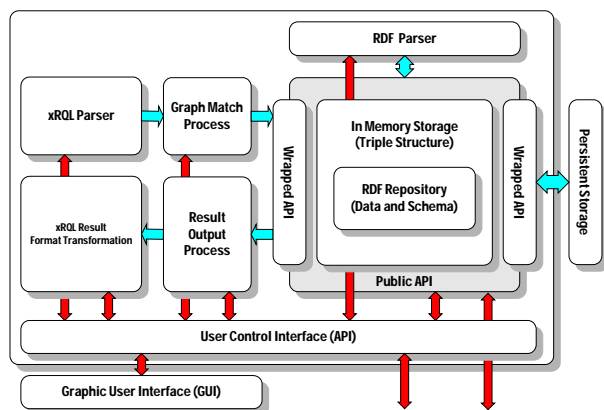


図 14 : xRDF データベースエンジンの構成

#### 4.1. ユーザ操作画面

本システムではデータベース操作のための API に加え、データベースを直接操作する際の検索問合せ作成のアシストやデータ入出力機能を備えたツールを GUI として提供する。

#### 4.2. xRQL パーサー

xRQL のパーサーはユーザインタフェースにおいて入力された xRQL 問合せの記述を解釈し、query オブジェクトを生成する。この query オブジェクトは GraphMatch 処理モジュールに渡されて処理される。本パーサーは以下の 3 つのコンポーネントで構成した。

- **preprocessor**  
preprocessor は入力されたクエリの文字列を lexical アナライザと syntax アナライザで処理可能な形式へ変換する。
- **lexical アナライザ**  
lexical アナライザは simple XML parser(SAX)[24]を利用して構成した。また、xRQL の定義において、まだ XML 化していない部分（パス表現等）は、JavaCC で開発した小さなパーサーを用いて処理する。
- **syntax アナライザ**  
syntax アナライザはトークンキューからトークンを一つ一つ抽出する。syntax が適合した場合には、対応する query オブジェクトが生成される。最後に、全てのクエリを表現する一つの query オブジェクトが構成される。

#### 4.3. xRQL 結果フォーマット変換

結果フォーマットの情報は syntax アナライザにより解析される。また、その情報を保存するために結果

のフォーマットのオブジェクトが生成され、オブジェクトは最終的な query オブジェクトとして統合される。この変換モジュールは query オブジェクトより結果フォーマットのオブジェクトを抽出し、グラフ適合処理から取得した結果を処理するために用いる。

### 5. 関連研究

本章では、RDF のリソースの検索を目的として提案された代表的な検索言語について簡単に説明する。

Versa[4]は様々なアプリケーションにて用いる RDF のツールとして、開発者からのニーズに基づいて開発された検索言語である。Versa はノードと矢印で示される RDF のグラフ構造に基づいて設計されているため、RDF リソースとスキーマの階層的な構成についての考慮が不足している。

Squish[5]は SQL に似た RDF の query 言語である。単純なグラフ構造に対する検索言語であり、RDF 検索言語としてのいくつかの機能要求をテストするために用いることができる。クエリは subject, predicate, object の triple を表現して行われ、複数のグラフの連結についても対応している。Versa と同様に SquishQL は階層化された RDF のリソースとスキーマの関連には対応がなされていない。

RQL[3]は OQL に似た形式機能言語であり、ラベルグラフのモデルに基づく RDF の構造を考慮した検索言語である。SQL に似たシンタックスを用いており、RDF で記述されたリソースとスキーマの両方とも検索可能である。

上記のように、現状の検索言語は非 XML の形式を用いており、XML 形式と比較すると RDF で記述される複雑な形式を表現する上で制限がある。また、プレーンなテキスト形式で RDF データのような階層的なモデルを記述するのは難しい。これらの検索言語は RDF データやスキーマに関する検索機能のみを定義しており、RDF のリソースデータに対する作成、削除、変更といった操作はサポートされていない。また、既存の RDF 検索言語は検索結果のフォーマット定義を行っておらず、このような検索結果のフォーマットはマシンによる自動処理が容易ではない。

### 6. おわりに

本稿では、我々の提案する XML 形式の RDF 検索言語 xRQL について紹介した。xRQL の記述形式として、RDF のような XML データが分散された環境において利用しやすいように XML 形式を採用した。xRQL の機能としては、オブジェクト指向のコンセプトに基づ



いて検索、削除、作成、変更操作の機能を設計した。また、結果のフォーマット定義をクエリにおいて指定可能とし、結果の処理を柔軟に行えるようにした。本稿では、我々の実装した xRQL のデータベースエンジンについても紹介している。今後は今回作成した xRQL データベースエンジンを実際のアプリケーションで利用して評価し、さらに必要な機能や改善点について検討を行う予定である。例えば、演算機能やオントロジーへの対応に関しては新たに検討する必要があると考えている。

## 参考文献

- [1] Ora Lassila and Ralph R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification", W3C Recommendation, 22 February 1999.
- [2] Dan Brickley and R.V. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema", W3C Working Draft, 30 April 2002.
- [3] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, Michel Scholl. "RQL: A Declarative Query Language for RDF", WWW2002, Honolulu, Hawaii, USA. ACM 1-58113-449-5/02/0005. May 7-11, 2002.
- [4] Mike Olson, Uche Ogbuji. "RDF Query and Inference: Versa", Available at <http://rdfinference.org/versa.doc?xslt=db.xslt>, 2002
- [5] L. Miller. "RDF Query using SquishQL", Available at <http://swordfish.rdfweb.org/rdfquery/>, 2001.
- [6] Lee Jonas, Stefan Kokkelink. "RDF Path Requirements", Available at: <http://zoe.mathematik.uni-osnabrueck.de/RDFPath/req/>
- [7] ODMG "Object Query Language (OQL). User Manual", Release 5.0 - February 1998
- [8] D. Chamberlin, D. Florescu, J. Robie, J. Simeon, and M. Stefanescu. "XQuery: A Query Language for XML", Working draft, World Wide Web Consortium. Available at <http://www.w3.org/TR/xquery/>, June 2001
- [9] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener. "The Lorel Query Language for Semistructured Data", International Journal on Digital Libraries, 1(1):68{88, April 1997.
- [10] Jeen Broekstra, Arjohn Kampman. (Sesame) "Query Language Definition", EU-IST Project IST-1999-10132 On-To-Knowledge 1999.
- [11] Serge Abiteboul. "Towards a deductive object-oriented database language", the First International Conference on Deductive and Object-Oriented Databases, pages 453-472. North Holland, Amsterdam, 1989.
- [12] Simon Brown. "The semantic of object-oriented databases", The University of Sheffield Department of Computer Science July 1999
- [13] Daniel Kim Chung Chan. "Object-oriented query language design and processing", Doctor thesis. UNIVERSITY OF WISCONSIN-MADISON 1993
- [14] Zeki O. Bayram, Barrett R. Bryant, Faik Hakan Bilgen. A "Deductive Declarative Object-Oriented Data Model and Query Language based on Narrowing", Proceedings of ISCS XI, the 11th International Symposium on Computer and Information Sciences, Antalya, Turkey. November 6-8, 1996
- [15] Flavius Frasincar Geert-Jan Houben Richard Vdovjak Peter Barna. "RAL: an Algebra for Querying RDF", The Third International Conference on Web Information Systems Engineering(WISE'00), p. 173. Singapore. December 12 - 14, 2002
- [16] Joachim Peer. "A Logic Programming Approach To RDF Document And Query Transformation", Workshop on Knowledge Transformation for the Semantic Web at the 15th European Conference on Artificial Intelligence. Lyon, France (2002)
- [17] Nitish Manocha, Diane J. Cook, and Lawrence B. Holder. "Structural Web Search Using a Graph-Based Discovery System", FLAIRS Conference: P133-137. 2001
- [18] Jeonghee Kim, Taisook Han Kyu, Young Whang. "Visualization of Path Expressions in a Visual Object-Oriented Database Query Language", 6th International Conference on Database Systems for Advanced Applications (DASFAA '99) Hsinchu, Taiwan, p. 99. April 19 - 22, 1999
- [19] Martin Erwig, Ralf Hartmut Güting. "Explicit Graphs in a Functional Model for Spatial Databases", IEEE Transactions on Knowledge and Data Engineering, Volume 6. Number 5, P.787-804. October 1994
- [20] R. Fikes. "DAML+OIL query language proposal". Available at <http://www.daml.org/listarchive/joint-committee/0572.html>, August 2001
- [21] "Edutella RDF Query Exchange Language", Available at: <http://www.kbs.uni-hannover.de/Diverses/edutella-archive/discussion/pdf00004.pdf>, 2002
- [22] T. Bray, J. Paoli, and C.M. Sperberg-McQueen. "Extensible markup language (XML) 1.0", W3C Recommendation, February 1998. Available at <http://www.w3.org/TR/REC-xml/>.
- [23] L. Sheng, Z. M. Özsoyoglu, G. Özsoyoglu. "A Graph Query Language and its Query processing", IEEE ICDE Conf., Australia, March 1999
- [24] SAX Available at :<http://www.saxproject.org/> 2002