2P-2

# 概念記述言語 CDL データの意味的検索法

## An Efficient Method for Semantic Search of CDL(Concept Description Language) Data

堀内 暢之† 高山 智史‡ 石塚 満‡

† 東京大学工学部電子情報学科 The University of Tokyo

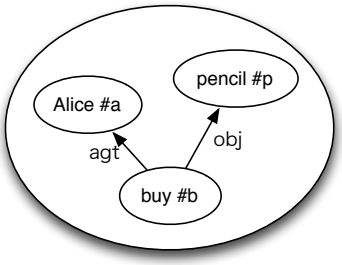
‡ 東京大学大学院情報理工学系研究科 The University of Tokyo

## 1. まえがき

ISeC[1]が開発と標準化活動を行っている，概念記述言語 CDL(Concept Description Language)に対して検索する手法を提案する．まず，CDL について紹介し，次に先行研究である「SPARQL を利用した CDL データの検索」についてふれ，その高速化を狙ったリレーショナルデータベース MySQL を利用する手法を提案する．

## 2. CDL とは

CDL のデータは Entity, Relation, Attribute からなる有向ハイパーグラフである．[石塚 09, CWL 08] Entity と Relation は入れ子構造を持ち，Arc によって Entity と Entity が Relation で結ばれる．図 1 に示した CDL の例は”Alice bought penciles”を表している。




```
{#
  <#a:Alice>
  <#b:buy.@past>
  <#p:pencil.@pl>
  [#b agt #a]
  [#b obj #p]
}
```

図 1 CDL の例

## 3. SPARQL を用いた CDL データの検索

CDL のデータを RDF に変換，リレーショナルデータベースに格納し，CDQL クエリを入力とし，SPARQL[2]クエリを生成，SPARQL のエンジン Jena[3]内で SQL を生成して，CDL データの検索を行うシステム(図 2)を，高山氏が設計、実装した.[高山 10] このシステムによって CDL に対する構造的・意味的な検索が可能となるが，5 万トリプルからなるデータへの検索に十数秒～数分ほどかかるため，実用性に問題があった．このシステムの問題は CDQL クエリ→SPARQL クエリ→SQL クエリと多段階に変換・生成がなされるがために最適ではない冗長な問い合わせになってしまうことと，意味的拡張のために正規表現を使用したことにある．SPARQL から独自エンジンに切り替えたシステムにおいて 1 秒前後の検索時間を実現している.[高山 10]

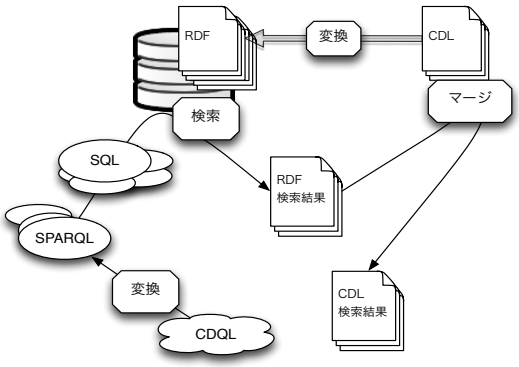



図 2 SPARQL を利用した CDL データの検索

## 4. 提案手法

前出のシステムを高速化するため，中間に RDF や SPARQL の使用をせず，CDL を直接リレーショナルデータベースへ格納し，CDQL クエリから SQL クエリに直接変換して検索を行うこととした.

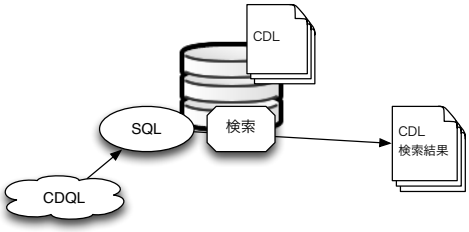



図 3 アーキテクチャ図

[1]特定非営利活動法人セマンティック・コンピューティング研究開発機構 http://instsec.org/

[2]SPARQL Protocol and RDF Query Language http://www.w3.org/TR/rdf-sparql-query/

[3] http://incubator.apache.org/jena/



Copyright ©2012 Information Processing Society of Japan.
All Rights Reserved.

### 4.1 CDL データの保存

Entity, Relation, Attribute を図 4 に示したそれぞれのテーブルに保存する. Realization Label が一意な識別子とはならないため, システム側で id を別に付与する.

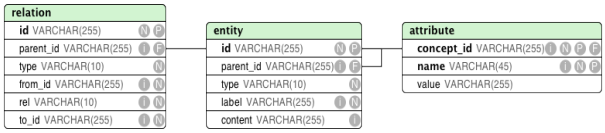



図 4 ERD 図

### 4.2 CDL データの復元

entity テーブルの parent_id, relation テーブルと attribute テーブルの concept_id を用いて CDL データを復元することができる.

### 4.3 CDQL (CDL Query Language)

CDL に対するグラフマッチ・意味的検索のために, 問い合わせ言語 CDQL を拡張した. 基本的な構文は GET ENTITY WHERE {CDL'}であり、{CDL'}の部分では#realization ラベルの代わりに?realization を用い、definition ラベルの意味的な拡張のために+を後置する.

```
GET ENTITY WHERE
{?
  <?07:cure+>
  <?0O:end>
  <?27:likely>
  [?0O aoj ?07]
  {?01
    <?0X:medical condition>
  }
}
```

図 5 CDQL クエリの例

### 4.4 CDL データに対するグラフマッチング

CDL のグラフにマッチする SQL クエリを生成する手順を以下に示す.

1. 各 entity, relation, attribute に名前をつける.
2. FROM 句では各 entity, relation, attribute を INNER JOIN する. それぞれは AS 句で 1 でつけた別名を使用する.
3. WHERE 句で definition label に対する制約をかける. (例として、entity_0000.content LIKE 'cure%' )
4. ルートのエンティティの id を SELECT する.

手順 1〜4 によって生成された SQL クエリにより、問い合わせに適合する CDL エンティティの id の一覧が得られる. そして、id より元の CDL データを得る.

95 万トリプルのデータに対しての検索にかかる時間は数 ms 程度であり, CDL データの復元にかかる時間は 100ms 程度(1 件のみヒットした場合)である. 1 つの CDQL クエリが 1 つの SQL クエリに変換されることが特徴的である. 実行されるのは単一の SQL クエリであり, MySQL のクエリオプティマイザによって最初の 1 つか 2 つの条件によって候補が絞り込まれるため, 検索実行時間はクエリの複雑さからほとんど影響を受けない. CDL データの取得はマッチした Entity の数だけイテレートされるため、マッチした件数と CDL データ取得の実行時間は比例する. CDL データの取得は CDL データをテキストとして保存しておくことにより省略可能である.

### 4.5 CDL データに対する意味的検索

Definition label に対するマッチングにおいて, 語句との厳密な一致だけで無く、下位語との一致も認める. 下位語の抽出には WordNet3.0[4]を用いた.

## 5. むすび

CDL 向けのクエリ言語 CDQL を拡張し, CDL に対する高速検索を実現した. 意味的なマッチングに関しては, 語義情報の利用, relation の曖昧な一致, クエリに対する不完全な一致など工夫の余地が多く残されている.

本システムは CDL データを対象としているが, RDF をはじめとする, 任意のトリプルで表せられるグラフへの応用が可能である.

## 参考文献

[石塚 09] 石塚満, 内田祐士, 横井俊夫 : 自然言語テキスト意味概念の共通的記述による次世代 Web 基盤, 知能と情報(日本知能情報ファジィ学会誌) Vol.21, No.4, pp.519-526 (2009)

[CWL 08] Common Web Language, http://www.w3.org/2005/Incubator/cwl/XGR-cwl/

[高山 10] 高山智史, 石塚満, 内田祐士 : 概念グラフマッチングによる自然言語テキストの意味的検索, 電子情報通信学会・信学技報「人工知能と知識処理」, vol. 110, no. 172, AI2010-14, pp. 25-29 (2010.8)

[4] http://wordnet.princeton.edu/



Copyright ©2012 Information Processing Society of Japan.
All Rights Reserved.