

RDF を用いた CDL (Concept Description Language) の表現と相互変換

山口 俊輔 石塚 満

東京大学 情報理工学系研究科 創造情報学専攻

1. はじめに

これからの次世代 Web の時代に向けて、あらゆるコンテンツやメディアの意味、内容、知識などの概念を表現できるように CDL (Concept Description Language) [1] が開発されている。CDL は各々のコンテンツやメディアにおいて個別に作られる。CDL の内でも Web を中心とする自然言語テキストが表す概念を対象としている CDL.ni (CDL natural language version) がある。CDL.ni はメタデータを用いたリソースの参照、コンピュータの理解可能を目指す次世代 Web を構築するセマンティック Web とは違い、Web の大半を占める自然言語テキストの概念記述によって次世代 Web 基盤の構築を目指す概念記述言語である。ただし、自然言語テキストの概念記述と Web コンテンツのメタデータという対象が異なる CDL.ni とセマンティック Web は競合する形ではなく、融合する形を目指している。

そこで本稿では、セマンティック Web との融合を実現させるため、リソースの関係を表現する RDF を用いた CDL.ni モデルの表現方法、RDF (CDL.ni 表現) と CDL.ni との相互変換について提案する。

2. RDF を用いた CDL.ni モデルの表現

本研究では、RDF のモデルを簡単に表現できるように提案されている N3 [2][3] という記法を採用した。CDL.ni のモデルのグラフ表現を図 1 に示す。

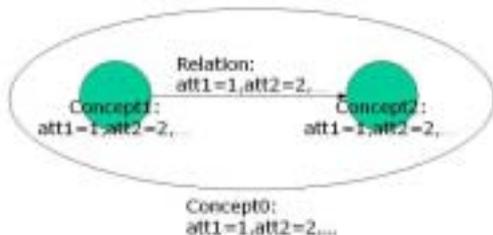


図 1: CDL.ni のグラフ表現

CDL.ni は Concept (概念) と Relation (関係) の 3 つ組み構造になっている。そして、その 3 つ組み構造自体も Concept の一つとして捉える。図 1 の CDL.ni をテキスト表現で表すと以下のようなになる。

```
Concept1:ID:attribute >Relation
                                Concept2:ID:attribute
```

この表現を用いて以下のような自然言語文と対応する CDL.ni の簡単な例を示す。

```
"John broke the door."
{#A001"break":01:temp=past.view=entry>agt
"John":02,>obj"door":03}:00:lang=en
(#A001 は事象などの複合概念を表す ID,
01,02,03 などはローカル ID, temp は時間性の属性,
属性値は past, view は状況, 事象を表す属性,
属性値は entry でこの文の主概念を表す。)
この例を使用して RDF を用いた表現にすると以下のように示される。
```

```
<break> <ID> <01>; att:temp <past>; att:view
<entry>.
<John> <ID> <02>.
<break> <agt> <John>.
<door> <ID> <03>.
<break> <obj> <door>.
<#A001> cdl:entity <break>, <John>, <door>.
<#A001> <ID> <00>.
```

属性を表すため att:xxx を導入、概念を表すために cdl:entity を導入、各々は名前空間接頭辞で宣言する。ローカルな ID や属性の表し方は以下のようなになる。

```
"break":01 <break> <ID> <01>.
"break":temp=past.view=entry <break>
<ID> att:temp <past>; att:view <entry>.
```

複合概念の場合、break などの要素概念とは表し方が異なり、概念を含むことを表すように cdl:entity を使用する。記述の方法は、複合概念 ID のノードを記述し、次に cdl:entity、その後複合概念に含まれる概念のノードを記述する。ノードが幾つかある場合は、”,”で続けて記述する。ただし、複合概念のローカル ID の表し方は要素概念のローカル ID の表し方と変わらず同様に表す。

Representation and Mutual Conversion of CDL (Concept Description Language) Using RDF
Shunsuke Yamaguchi, and Mitsuru Ishizuka
Department of Creative Informatics, Graduate School of Information Science and Technology, The University of Tokyo

表 1 拡張 BNF 記法による CDL.n1 の生成規則

```

Concept ::=
( CompoundConcept |
  ConceptName ):ConceptID:Attributes

CompoundConcept ::=
{ GlobalID Concept Relation Concept
  (, Relation Concept)*
  (; Concept Relation Concep)* }

Relation ::= RelationName

RelationName ::= '>agt' | '>and' | ...

ConceptName ::= "a literal expression"

GlobalID ::= '#', 'A'-'Z'
  and three characters of '0'-'9'

ConceptID ::= any two characters
  of '0'-'9', 'a'-'z', 'A'-'Z'

Attributes ::= Attribute (.Attribute)*

Attribute ::= 'view=entry' | 'temp=past' |...

```

3. CDL.n1 から RDF への変換手法

CDL.n1 から RDF への変換手法は，CDL.n1 テキストの構文解析を行い，CDL.n1 の生成規則に対応させる．生成規則が正しく認識されたとき，その生成規則に対応して，変換処理を実行するようにした．表 1 に拡張 BNF 記法による生成規則を提示する．以下は生成規則の対応についてである．

Concept では，CompoundConcept か ConceptName かどうかを調べる．ここで複合概念が要素概念かに分かれる．複合概念の場合，CompoundConcept においてその概念を形成している要素概念，関係について生成規則の対応を調べる．その要素概念が複合概念である場合はさらに深く調べる．また，複合概念の概念 ID は GlobalID において字句解析を行い，生成規則に対応しているかを調べる．対応している場合は，複合概念の ID を要素概念の ConceptName と同様に扱う．ConceptName は要素概念の名称について字句解析を行う．Relation については，概念を結ぶ関係がどの関係を表しているかの対応を調べる．ConceptID では概念のローカル ID の字句解析を行う．Attributes では，Attribute の数

について調べる．Attribute では，概念の属性がどの属性であるかの対応をとる．

以上のような生成規則の対応後に 2 章で示した，RDF を用いた CDL.n1 の表現にそれぞれの対応に合わせて書き換える．この生成規則からわかるように CDL.n1 の書式がボトムアップ形式であることから一番小さい要素概念から RDF に書き換える．

4. RDF から CDL.n1 への変換

RDF を用いた CDL.n1 から元の CDL.n1 への変換手法は，RDF の書式がトップダウン形式であることから 1 文毎に対し条件による変換を行う．

条件分岐させる文の種類は，(1)要素概念のローカル ID や属性を表す文，(2)概念と概念を結ぶ関係を表す文，(3)複合概念にどんな要素概念が含まれているのかを表す文，(4)複合概念のローカル ID や属性を表す文の 4 種類である．

変換手順は，(2)文の各々の概念に合った要素概念の内容を(1)文，または(4)文から代入する．そして，(3)文の複合概念に(2)文の概念が含まれる場合(3)文の複合概念にその関係を書き込む．もし，その複合概念が更に上位の複合概念に含まれる場合，(3)文を(4)文の概念の内容として代入する．また，それぞれ文変換は代入動作のときに実行する．以上の手順を繰り返すことによって RDF から CDL.n1 に変換する．

4. 考察

CDL.n1 はまだ開発されて間もない概念言語であり，CDL.n1 をそのまま利用できるアプリケーションなども少ない．しかし，RDF を用いた CDL.n1 表現によって，RDF のアプリケーションを利用可能となる．また，CDL.n1 のままでは自然言語の意味，概念をリソースと扱うことができないが RDF に変換することでリソースとして扱うことが可能になると考えられる．また，その逆の利用も可能となると，これからの CDL.n1 に色々な利用法が考えられる．RDF，CDL.n1 の相互変換が可能となったことで，CDL.n1 とセマンティック Web の融合にも役立つと考えられる．

5. まとめ

CDL.n1 モデルの RDF を用いた表現方法を提案した．また，その RDF と CDL.n1 の相互変換システムの実装，考察を行った．

参考文献

- [1]Specification of CDL.n1, ISeC 資料(2005)
<http://www.instsec.org/tr>
- [2]Tim Berners-Lee, Notion3:Ideas about Web Architecture - yet another notation, (2006),
<http://www.w3.org/DesignIssues/Notation3>