

CHISE の階層的素性名の RDF 化の試みについて

守岡 知彦¹

概要：「CHISE IDS 漢字検索」や「東洋学文献類目データベース Ver.7」の詳細画面表示に用いている「E₅T」に対して RDF/XML 形式での出力機能の追加を試みた。ここでは階層的素性名の RDF/XML における述語へのマッピングに焦点を当てて概説する。

1. はじめに

CHISE project [1] では汎用文字符号に対して独立な次世代文字処理技術の確立を目指して、文字、特に、漢字に関するさまざまな知識を機械可読な形式で記述した「CHISE 文字オントロジー」、および、その処理系である CHISE 実装の開発を続けている。[2] CHISE は素性の集合によってオブジェクトを表現する『Chaon モデル』[3]に基づき、文字をオブジェクトとして扱うことで、文字の内部表現を隠蔽するとともに、文字オブジェクトを扱うためのさまざまなインターフェースを提供できるようにしている。

CHISE ではオブジェクト間の関係を表すために『関係素性』と呼ぶ素性を用いている。これはオブジェクト間の名前付き有向グラフのリンクとなるものであり、これによって CHISE は文字に関するさまざまな知識を意味ネットワークによって記述するための枠組となっている。CHISE は当初は文字専用のものであったが、後に、任意の対象を扱えるように一般化した Concord [4] を開発し、CHISE 自体も Concord を使って再実装を行った。また、Concord を使って「東洋学文献類目データベース」のリファクタリング作業を行い [5]「東洋学文献類目データベース Ver.7」[6]として公開するなど、実際に文字以外の大規模データベースの開発・運用を行っている。

一方、Web の世界ではメタデータを記述するための形式として Resource Description Framework (RDF) [7] が標準化されており、さまざまなメタデータやセマンティックな情報の交換のための標準形式として盛んに使われるようになってきた。特に、RDF を XML 形式で表現した RDF/XML [8] が標準的に使われている。Concord 上のデータをインターネット上で公開されているさまざまな情報資源と円滑に連携させたり、Concord ベースで開発・公

開されたデータを素材として第三者が Web サービスを作れるようにするためには、Concord 上のデータを RDF/XML 形式で配信できるようにすることが重要であるといえる。

そこで、Concord 形式のデータを RDF/XML 形式で出力するために、Concord/CHISE の表示・編集用 Web サービスである E₅T[9] に RDF/XML 形式での出力機能の追加を試みた。本稿では Concord 形式を RDF 形式に一対一対応させる上での問題点や E₅T で採用した手法について、階層的素性名とメタデータ素性の表現に焦点を当てて、概説する。

2. Concord のデータモデルと形式

Concord は CHISE を一般化したシステムとなっており、そのデータモデルは『Chaon モデル』に基づき、素性（素性名）と素性の値（素性値）の対（素性対）の集合として表現されたオブジェクトを基礎とする。素性には、オブジェクト間の関係を示す『関係素性』や『構造素性』があり、オブジェクトをノードとした有向グラフによる知識表現形式の一種と見なすことができる。

2.1 素性、素性名、素性対

Concord における素性 (feature) は記述対象となるオブジェクトの性質を示すためのものであり、ある名前を持つオブジェクトである。

Concord では、この素性（素性名）を使ってオブジェクトを記述する訳であるが、この時、素性名とその値（素性値；feature value）の対を『素性対』(feature pair)と呼ぶ。Concord では各オブジェクトは、この素性対の集合によって表現される。ここで、この素性対の集合を『オブジェクト仕様』(object-spec)と呼ぶ。

素性は概念的には名前（文字列）によって指されるオブジェクトであり、*¹ ある素性が複数の名前から指され得る。

¹ 京都大学人文科学研究所
Institute for Research in Humanities, Kyoto University

*¹ Concord における実装上もそうである。

即ち、『別名』(alias)を持つことができる。^{*2}よって、素性
の名前(素性名; feature name)と素性は厳密には別物で
あるが、特に問題がない限り、話を簡単にするために、素
性と素性名を同一視して述べることにする。

なお、素性名によって指される『オブジェクトとしての
素性』という側面を強調したい場合、『素性オブジェクト』
(feature object)という語を用いることにする。素性オブ
ジェクトはある素性を持つオブジェクトの集合であり、オ
ブジェクトと素性値の対応関係を管理するデータベースオ
ブジェクトと看做すこともできる。この意味では、素性は
素性名と素性値の対を包含するものとも言える。^{*3}

2.2 素性の種類

Concord の素性は大別すると

基礎素性 数値や識別子(シンボル)といったアトミック
なデータ、または、それらのリストや配列を値として
取る素性

構造素性 オブジェクトを要素として持つリストや配列を
値として取る素性

ID 素性 オブジェクトに対する ID(素性名においてユ
ニークな数値または識別子)を値として取る素性。あ
る ID 素性において、その素性を持つ各オブジェクト
とその素性値である ID は 1 対 1 対応していなければ
ならない(これにより、値である ID をキーにオブジェ
クトを得るための索引を作ることができる)

関係素性 オブジェクトの集合を値として取り、値に取
った各オブジェクトと素性を持つオブジェクトの関係を
表す素性。これは、オブジェクトをノードとした有向
グラフのリンクとなるものである。関係素性対を持つ
オブジェクト(主語)とその値の各要素(目的語)の
間には、その主語を値の要素とする逆関係素性対がそ
の目的語に付く

メタデータ素性 元になる素性に対するメタデータを記述
するための素性
に分類することができる。

ID 素性、関係素性、メタデータ素性の名前は、Concord
(CHISE)における素性名の命名規則によって、特定の形式
を用いることになっている。

ID 素性は '=' から始まる名前を持つ。また、CHISE の
『グリフ ID 素性名の命名規則』[10] [11] に従い、例示オブ
ジェクトの集合を指すための名前から、それを抽象化した
オブジェクトの集合を指すための派生名を接頭辞によって

機械的に決定できるようにしている。

関係素性は '->' もしくは '<-' から始まる名前を持つ。
両者は互いに逆関係となっており、あるオブジェクト A が
関係素性 '->foo' を持ち、その値が (B₁ B₂ ...) である時、
オブジェクト B_i は逆関係素性 '<-foo' を持ち、オブジェ
クト A はその素性値の要素のひとつとなる。

メタデータ素性は、言及対象となる素性名の後ろに '*'
から始まる文字列を付けた名前を持つ。ここで、'*' の後
に続く文字列を『メタデータ識別子』と呼ぶ。

2.3 階層的素性名

Concord では、ある素性に対して、出典や学説の違いや
用途等によって、異なる素性値を記述できるようにするた
め、素性名に対してその値が属するドメインを示す識別子
(これを『ドメイン識別子』と呼ぶ)を付けて構造化する
『階層的素性名方式』を用いている。

この方式では、ベースとなるドメインを持たない素性名
が foo の時、ドメイン識別子 bar を付けたドメイン付き素
性名は foo@bar で表される。

また、ドメインを持つ素性名 foo@bar に対してドメイン
識別子 baz を付ける場合、素性名は foo@bar/baz となる。
以下、セパレータとして '/' を用いることで、再帰的にド
メイン付き素性名を構成することができる。

2.4 類型

Concord には本質的な意味でのクラスは存在しない。し
かしながら、あらゆるオブジェクトを単一の空間に置くの
は不合理であることも少なくないので、オブジェクトの種
類毎に異なる空間を設定することが可能である。これを
『類型』(genre)と呼ぶ。

Concord のオブジェクトは必ずどれかの類型に属する。
例えば、文字オブジェクトは character 類型に属し、素性
オブジェクトは feature 類型に属する。つまり、Concord
のオブジェクトは類型とオブジェクト仕様の組によって定
義できる訳である。

3. RDF へのマッピング

Concord のデータモデルを RDF と比較した場合、素性
対を持つオブジェクトが『主語』(subject)、素性名が『述
語』(predicate)、素性値が『目的語』(object)に対応する。
また、素性対を持つオブジェクトと素性対の組がトリプル
(triple)に対応する。

このように Concord と RDF のデータモデルはインス
タンスレベルでは同様であるが、その基礎となるデータモ
デルにおいては、RDF がクラスベースであるのに対し、
Concord はプロトタイプベース(クラスがない)とい
う違いがある。即ち、メタデータやオントロジー等を記述
するための語彙の定義のレベルに関しては Concord にお

^{*2} 実際、CHISE では幾つかの符号化文字集合を指すための素性が
別名を持っている。

^{*3} この意味で、『オブジェクトは素性の集合によって示される』と
いうように表現されるが、ここでの「素性」は素性オブジェクト
を示しており、実際には素性名と素性値の対からなる『素性対』
(feature pair)の集合によってオブジェクトが表現されているの
であって、素性名の集合による訳ではない。

ける素性の定義と RDF のスキーマ定義 [12] はかなり異なったものとなってしまふと考えられる。

また、階層的素性名やメタデータ素性、あるいは、CHISE におけるグリフ ID 素性名の命名規則といった、Concord の構造化された素性名を RDF でどのように表現するかも問題となる。これは構造化されていない RDF の述語に Concord の素性名と同様な構造を導入するにはどうしたら良いかということであり、RDF の述語は URI で表現されるため、構文論的には、URI へのマッピング問題という風にとらえることもできる。また、情報交換用の標準形式として XML を使うことを鑑みれば、XML の構文形式もその制約条件となる。

3.1 ID 素性名の RDF での表現

RDF の述語が URI で表現されることを鑑みれば、CHISE-wiki [13] (EgT) における ID 素性名の URI 表現形式を用いることが考えられる。即ち、例示オブジェクト (字形) 素性 =foo は「rep.foo」、抽象字形素性 =>>>foo は「gi.foo」、字体素性 [11] =>>foo は「g.foo」、統合字体素性 [11] =>foo は「o.foo」、抽象文字素性 =>foo は「a.foo」、超抽象文字素性 ==>foo は「a2.foo」という表現によって表せば良い訳である。

但し、Concord の階層的素性名のセパレーターとして用いている “@” は XML の識別子 (name; 『名前』) では使えないため、階層的素性名の表現に関しては別の方法を用い、素性名の内のドメイン部を取り除いた部分に関してのみこの手法を用いることとする。

3.2 関係素性名の RDF での表現

ID 素性名の場合と同様に、関係素性名の RDF の表現にも、EgT における関係素性名の URI 表現形式を用いれば良いといえる。即ち、関係素性 ->foo の URI 中での表現は to.foo とする。また、その逆関係素性 <-foo の URI 中での表現は from.foo とする。

また、ID 素性名の場合と同様に、階層的素性名の表現に関しては別の方法を用い、素性名の内のドメイン部を取り除いた部分に関してのみこの手法を用いることとする。

3.3 階層的素性名の RDF での表現

Concord の階層的素性名のセパレーターとして用いている “@”, “/” は RDF/XML における述語の名前で使うことができないために、Concord の階層的素性名を RDF で表現するためには何らかの工夫が必要となる。

単に問題となるセパレーター用の文字を表現することだけを考えれば、これらの文字をエスケープしてやれば良い訳であるが、このようにした場合、同種の素性であるという情報が RDF/XML で十分に表現できないという問題がある。

Concord における素性名に対応するのは RDF の述語であるが、これは URI で表現されるものであり、階層構造を持っている。また、RDF/XML では XML の名前空間を用いることで、URI の先頭部分を述語を表す名前のプレフィックス部に対応させるようになっている。よって、XML の名前空間を用いれば、Concord の階層的素性名を適切に RDF で表現することができると思われる。

しかしながら、XML の名前空間のプレフィックス部は階層化されていないので、何らかの工夫が必要である。選択肢としては、プレフィックスの階層化を行わない方法と行う方法が考えられる。

プレフィックスの階層化を行わない場合、適当に生成したプレフィックスを用いる方法やプレフィックスの重複を容認する方法が考えられる。いずれにしても、xmlns による名前空間の定義によって問題無く表現できる訳であるが、読み易さという点では問題があるといえ、また、後者の場合、プレフィックスの意味が場所によって変わると簡易的な処理がしづらくなると考えられる。

こうした観点から、EgT では XML の名前空間のプレフィックスを階層化することにした。具体的には、Concord の階層的素性名のドメイン部 (“@” より後ろの部分) 中のセパレーターである “/” を “.” に置き換えたものを XML における名前のプレフィックスとする。これにより、例えば、素性名 name@ja/JP は ja.JP:name という名前で表現できる。

実際には、rdf や dc といった良く使われるプレフィックスとの衝突を避けるために、この方法で生成したプレフィックスの前に “est.” という文字列を付加することにする。よって、実際には素性 name@ja/JP は est.ja.JP:name という名前で表現されることとなる。

但し、実際にダブリン・コア [14] を意図する素性名の場合、dc というプレフィックスを用いた方が良いので、名前空間の宣言において用いるプレフィックスを素性の素性 rdf-namespace-prefix で、URI を素性の素性 rdf-namespace-uri を用いて設定できるようにしている。

3.4 メタデータ素性名の RDF での表現

Concord のメタデータ素性名のセパレーターとして用いている “*” は RDF/XML における述語の名前で使うことができないために、Concord のメタデータ素性名を RDF で表現するためには何らかの工夫が必要となる。

単に問題となるセパレーター用の文字を表現することだけを考えれば、これらの文字をエスケープしてやれば良い訳であるが、メタデータ素性名が対象となる素性に関する言及であることを鑑みれば、RDF の文 (triple) に関する文として表現するのが望ましいといえる。

RDF では文に関する文を書くための方法として、文の『具体化』(reification) というものを用意している。これは

文を `rdf:Statement` 型のリソースとして記述し、それを主語とした文を書けるようにするものである。

RDF において、この『具体化』の仕組みを使って、文に関する文を書くための方法は幾つかあるが、EgT では記述のコンパクトさや Concord 形式との親和性の観点から、記述対象となる文に ID を付与し、その ID で示されるリソースを主語とするメタデータ素性対に対応する文を書くという方法を採用することにした。

例えば、ある Concord オブジェクトの URI を `http://foo/bar` とし、そのオブジェクトが素性対

```
(total-strokes . 7)
```

とメタデータ素性対

```
(total-strokes*source daikanwa)
```

を持つ時、EgT はこの 2 つの素性対で表現された文「『`http://foo/bar` の総画数は 7 画である』という情報の出典は大漢和である」を RDF/XML において

```
<rdf:Description rdf:about="http://foo/bar">
  <est:total-strokes
    rdf:ID="est...total-strokes">
    7
  </est:from.wrong>
</rdf:Description>
```

```
<rdf:Description
  rdf:about="#est...total-strokes">
  <dc:source> daikanwa </dc:source>
</rdf:Description>
```

という風に表現する。^{*4}

ここで、述語に付ける ID は述語の名前の “.” を “...” に置き換えたものを使うことにしている。また、メタデータ素性対に対応する文の述語にはメタデータ識別子を用いる。

3.5 文字のエスケープ

Concord 素性名中の XML のタグ名で使えない文字の内、階層的素性名やメタデータ素性名のセパレーターに関しては前述の方法で対処したが、それ以外のものに関しては別途次のようなエスケープ処理を行っている：

- 素性名の先頭に現れる “*” は `meta.` に置き換える。
- それ以外の “*” は `.-.` に置き換える。
- “/” は `...` に置き換える。

4. 課題

4.1 素性の定義情報の表現

3 節で述べたように、Concord と RDF のデータモデルはインスタンスレベルでは同様であるが、その基礎となるデータモデルにおいては、RDF がクラスベースであるの

に対し、Concord はプロトタイプベース（クラスがない）という違いがあり、また、CHISE 文字オントロジーにおける素性の定義が CCS 関連に偏っており、現状、貧弱であるということもあり、Concord における素性の定義の RDF 出力機能は未実装である。

Concord の素性定義情報（素性の素性の集合）を単純に RDF 化するだけなら簡単であるが、RDF スキーマ [12] を用いた定義を出力したい場合はセマンティクスのすり合わせ等の検討を行う必要があると考えられる。

4.2 語彙標準との対応

CHISE 文字オントロジーは、元々、RDF のことを意識せずに設計・開発を行ってきたため、そこで用いられている素性の仕様は、ダブリン・コアや DCMI Metadata Terms [15]、FOAF [16] といった標準的な語彙に沿ったものとなっていない。このため、単純に RDF 化した場合、EgT 独自の語彙を多用することになってしまい、他のメタデータやオントロジーとの連携ということを考えた場合、問題があるといえる。

RDF 化した場合になるべく標準的な語彙を用いるようにするためには、

(1) 従来の素性名や仕様を変えずに、RDF 出力レイヤーで対処する

(2) 標準に沿った形に素性名や素性の定義を変更するという 2 つの方法が考えられる。1 は既存のデータを変更する手間が要らないという利点があるが、RDF のデータを Concord に取り込むことを考えた場合の処理がややこしくなり、ラウンドトリップ変換ができなくなるおそれがあると考えられる。一方、2 は RDF との相互変換が簡単になるものの、既存の膨大なデータを変換する必要があり、この作業を安全に行うためにはチェック作業等を含めて多大な手間がかかる可能性がある。

よって、単に RDF 出力機能を提供するだけであれば 1 の方法が無難であるが、将来的に RDF との相互変換を行うとすれば 2 の方法を採用の方が良いといえる。

4.3 関係素性の表現

現在、Concord の関係素性を RDF/XML で表現する場合、関係素性 `->foo` は `to.foo`、その逆関係素性 `<-foo` は `from.foo` という名前の述語を用いることにしているが、DCMI Metadata Terms [15] では `hasVersion` と `isVersionOf` や `references` と `isReferencedBy` といった名前になっており、現状の EgT の方式はこの流儀に則ったものになっていない。

これは Concord ではある関係素性を定義したらその逆関係になる素性も自動的に定義されるようにしているため、現状の方式なら、接頭辞を変えるだけで逆関係の素性名を作れるのに対して、DCMI Metadata Terms のような

^{*4} ここでは、簡単にするために、名前空間の宣言を省略している。

名前では、関係を示すベースとなる名前が名詞の場合と動詞の場合で違う変化をさせないといけないことや、接頭辞と接尾辞の両方を使わなければならないといったことから、名前を機械的に生成・変換するのが面倒であるといえる。

DCMI Metadata Terms の場合、新たな名前を作る必要がないので、これで問題がないといえるが、Concord の場合、そういう訳には行かないので、DCMI Metadata Terms のような名前をそのまま採用するのは難しいと思われる。しかしながら、DCMI Metadata Terms との親和性は重要であると考えられるので、何らかの対策は必要であろう。例えば、

- (1) 少なくとも relation の一種として定義する
 - (2) 関係素性の素性として(人手で) RDF における名前を付け、RDF 化レイヤーで変換する
 - (3) 関係素性に対して、RDF 用の別名を付ける
- といった対策が考えられるが、少なくとも DCMI Metadata Terms の relation 関連の述語に対応するものだけでも対策する必要があると思われる。

4.4 素性値中のオブジェクトの表現

関係素性や構造素性は値の要素としてオブジェクトをとるが、こうした素性を RDF で表現する場合、現在は、値中の要素は、中身が空の `rdf:Description` タグを使って表現するようにしている。例えば、

```
((=id . B46S03070-11000002)
 (title . "先秦")
 (<-subarticle
  (:genre . book@ruimoku)
  (=id . B46S03070)
  (title . "中國史學入門(上)"))))
(->creator
  (:genre . creator@ruimoku)
  (=id . B46S03070-11000002/0)
 (->creator/name
  (:genre . person-name@ruimoku)
  (=name . "貝塚 茂樹"))))
```

このようなオブジェクト仕様は

```
<rdf:Description
  rdf:about="http://www.chise.org/est/rdf.cgi
?article@ruimoku=rep.id:B46S03070-11000002">
  <est:rep.id> B46S03070-11000002 </est:rep.id>
  <dc:title> 先秦 </dc:title>
  <est:from.subarticle>
    <rdf:Description
      rdf:about="http://www.chise.org/est/rdf.cgi
?book@ruimoku=rep.id:B46S03070">
      </rdf:Description>
    </est:from.subarticle>
  <est:to.creator>
    <rdf:Description
      rdf:about="http://www.chise.org/est/rdf.cgi
```

```
?creator@ruimoku=rep.id:B46S03070-11000002/0">
  </rdf:Description>
</est:to.creator>
</rdf:Description>
```

のように表現されるが、これでは参照されるオブジェクトの ID しか分からず、その属性を知るには該当オブジェクトの定義を参照しなければならず、読みづらい。そこで、参照されるオブジェクトを示す `rdf:Description` タグの中に幾つかの重要な属性を記述できるようにするための仕組みを設けるのが望ましいといえる。

4.5 URL の形式

現在、EgT の RDF 出力機能は CGI で実装されており、`http://www.chise.org/est/rdf.cgi?genre=feature:value` という URL でアクセスすることになっているが、これは `http://est.chise.org/rdf/genre/feature=value` というような静的な Web 頁的な形式にした方が良いかもしれない。

5. おわりに

「CHISE 文字オントロジー」や「東洋学文献類目データベース Ver.7」で用いている有向グラフ型データベースシステム Concord 上のデータの RDF 化を試みた。これは Concord 用の表示・編集用 Web サービスである EgT に RDF/XML 形式による出力機能を追加することで実現している。

Concord 上のデータを RDF/XML 形式する上で問題となるのは、Concord オブジェクトでは CHISE 文字オントロジーで導入された『階層的素性名方式』に基づく構造を持った素性名を XML の世界においてどのように表現するかということである。ここでは素性名におけるドメインを XML の名前空間に対応させるとともに、“.” をセパレーターとする階層的な命名規則を導入することでこの問題を解決した。また、メタデータ素性に関しては、文の『具体化』(reification) の仕組みを用い、メタデータ素性の言及対象となる素性対に対応する述語に ID を付与し、その ID で示されるリソースを主語とするメタデータ素性対に対応する文を書くという方法によって実現した。こうした手法によって、Concord オブジェクトを RDF/XML 形式で表現することができた。

なお、この機能は既に「CHISE IDS 漢字検索」[17] や「東洋学文献類目検索 Ver.7」の検索結果の詳細画面表示 (EgT) で試験的に公開しており、EgT の表示画面上の「RDF」ボタンを押すことで、該当オブジェクトの RDF/XML 形式の出力を得ることができる。

この EgT に実装した RDF/XML 出力機能は現在の所、一般のオブジェクトの定義情報に限られており、素性の定義情報を RDF スキーマや OWL 等を用いて出力すること

はできていない。また、素性の定義や「CHISE 文字オントロジー」等の記述に用いている語彙が独自のものになっており、ダブリンコア等のメタデータ標準と調和していないのをどうするかも今後行うべき重要な課題であろう。

参考文献

- [1] : CHISE Project, <http://www.chise.org>.
- [2] 守岡知彦: 文字オントロジーに基づく文字処理について, 情処研報, Vol. 2006, No. 112, pp. 25-32 (2006). 2006-CH-72.
- [3] Morioka, T.: CHISE: Character Processing based on Character Ontology, *Large-scale Knowledge Resources (LKR2008)*, LNAI, No. 4938, pp. 148-162 (2008).
- [4] 守岡知彦: Concord: プロトタイプ方式のオブジェクト指向データベースの試み, Linux Conference 抄録集, Vol. 4 (2006).
- [5] 守岡知彦: 東洋学文献類目のセマンティック Wiki 化の試み, 情処研報, Vol. 2011-CH-91, No. 6, pp. 1-8 (2011).
- [6] : 東洋学文献類目データベース第7版, <http://ruimoku.zinbun.kyoto-u.ac.jp/ruimoku7/>.
- [7] World Wide Web Consortium: *Resource Description Framework (RDF): Concepts and Abstract Syntax*, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/> (2004).
- [8] World Wide Web Consortium: *RDF/XML Syntax Specification (Revised)*, <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/> (2004).
- [9] 守岡知彦: Wiki 的手法に基づく構造化データの編集について, 人文科学とコンピュータシンポジウム論文集—人工工学の可能性—異分野融合による「実質化」の方法—, 情報処理学会シンポジウムシリーズ, Vol. 2010, No. 15, 情報処理学会, 情報処理学会, pp. 33-40 (2010).
- [10] 守岡知彦: CHISE に基づくグリフ・オントロジーの試み, 人文科学とコンピュータシンポジウム論文集—デジタル・ヒューマニティーズの可能性—, 情報処理学会シンポジウムシリーズ, Vol. 2009, No. 16, 情報処理学会, 情報処理学会, pp. 9-14 (2009).
- [11] 守岡知彦: 漢字字体情報の安定的な交換について, 人文科学とコンピュータシンポジウム論文集:「デジタル・アーカイブ」再考—いま改めて問う記録・保存・活用の技術—, 情報処理学会シンポジウムシリーズ, Vol. 2011, No. 8, 情報処理学会, 情報処理学会, pp. 353-360 (2011).
- [12] World Wide Web Consortium: *RDF Vocabulary Description Language 1.0: RDF Schema*, <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/> (2004).
- [13] 守岡知彦: CHISE のセマンティック Wiki 化の試み, 情処研報, Vol. 2010-CH-87, No. 8, pp. 1-8 (2010).
- [14] Dublin Core Metadata Initiative: *Dublin Core Metadata Element Set, Version 1.1*, <http://dublincore.org/documents/2012/06/14/dces/> (2012).
- [15] DCMI Usage Board: *DCMI Metadata Terms*, <http://dublincore.org/documents/2012/06/14/dcmi-terms/> (2012).
- [16] Brickley, D. and Miller, L.: *FOAF Vocabulary Specification 0.98*, http://xmlns.com/foaf/spec/20100809_marco_polo_edition (2010). Namespace Document 9.
- [17] 守岡知彦: CHISE IDS 漢字検索, <http://www.chise.org/ids-find>.