すべてをコンピュータの中に

（繋がってしまったデータとその未来）

公開シンポジウム

漢字構造情報のRDF化の試み
（守岡知彦）

国立国会図書館のメタデータ標準 — データを繋げるメタデータ: DC-NDL —
（柴田洋子）

PageRank と学術論文の評価: ノーベル賞の窓を探そう
（藤田裕二）

全国共同利用・共同研究拠点
「人文学諸領域の複合的共同研究国際拠点」
2013.2.16

目次

この冊子の説明

この冊子は，京都大学人文科学研究所共同研究プロジェクト:「情報処理技術は漢字文献からどのような情報を抽出できるか――人文情報学の基礎を築く」主催の公開シンポジウム「**すべてをコンピュータの中に（繋がってしまったデータとその未来）**」（2013年2月16日開催）の予稿集である。

シンポジウムについて

日時と場所

2013年2月16日（金）, 13:00-17:40
京都大学人文科学研究所本館101セミナー室

録画による記録

人文: USTREAM (http://ustream.tv/channel/zinbun/)

プログラム

第1部

趣旨説明
13:00-13:10

漢字構造情報のRDF化の試み
守岡知彦（京都大学人文科学研究所）
13:10-14:00

国立国会図書館のメタデータ標準—データを繋げるメタデータ: DC-NDL—
柴田洋子（国立国会図書館電子情報部）
14:20-15:10

PageRank と学術論文の評価: ノーベル賞の窓を探そう
藤田裕二（(株)ターンストーンリサーチ, 日本大学）
15:30-16:20

第2部

パネルディスカッション
コメントと問題提起: 永崎研宣（人文情報学研究所）
16:40-17:40

漢字構造情報の RDF 化の試み

守岡 知彦

1　はじめに

多くの漢字は偏と旁などの部品の組み合わせによって構成されている。こうした漢字の部品の組合せ構造は形の抽象的表現となるだけでなく、字義や音価にも関係しており、字源に基づく文字構造の分析は「解字」と呼ばれ、そうしたデータは重要な辞書記述のひとつでもある。

こうした漢字の部品の組合せ構造に関する情報のことを「漢字構造情報」と呼ぶことにする。漢字構造情報の機械可読な表現手法には、幾つかの表現形式が提案され利用されてきたが、Ideographic Description Sequence (IDS) 形式が ISO/IEC 10646 [2] の一部として標準化されている。

CHISE project では 2001 年度未踏ソフトウェア創造事業の助成を受けて、当時 UCS に収録されていた統合漢字、同拡張 A, B の約 7 万字の漢字を対象とする IDS 形式に基づく漢字構造情報のデータベース化を行い、「CHISE 漢字構造情報データベース」として公開した。[14][1] 2005 年には WWW ベースの検索サービス「CHISE IDS 漢字検索」[8] を公開した。これは 2005 年の IRG 京都会議のデモを通じ、漢字の符号化作業における有用性も広く知られるようになり、現在では新たに UCS 統合漢字を提案する際に提出しなければならない情報のひとつとなっている。

CHISE では「CHISE 漢字構造情報データベース」は CHISE 文字オントロジー [9][4] の一部を構成するものとなっており、漢字構造情報は文字に関する素性の一種として扱われ、全体として文字に関する意味ネットワークになっている。WWW 上でも、「CHISE IDS 漢字検索」の検索結果の左端をクリックすることで、各文字の情報を得ることができるようになっており、リンクを辿っていくことで文字に関する各種情報を得ることができるようになっている。[11]

一方、WWW の世界では、メタデータやオントロジーを表現するための形式として RDF [7] が標準化されている。CHISE 文字オントロジーのような文字に関する知識を表現した情報資源も他の情報資源との連携を鑑みれば RDF 等の WWW 標準に対応することが重要であるといえる。そこで、2012 年にはこうした情報の RDF/XML 形式 [6] での出力機能を実装した。しかしながら、現状では CHISE 文字オントロジーの構造をそのまま RDF で表現したものとなっており、標準的な語彙の利用や独自語彙の定義の点で問題があるといえる。漢字構造情報に関しても、IDS の構文木をほぼそのまま RDF 化したようなものとなっており、RDF に基づく構造化という点で現状のままで良いかどうか検討の余地があるといえる。

[1] 現在では、拡張漢字 C, D もカバーしている。

そこで、本稿では IDS 形式やその拡張表現に基づいた漢字構造情報の RDF 化に関して複数の手法を挙げて検討し、その問題点や解決法について議論したい。

2 IDS とは

漢字構造情報に基づく符号化手法の試みは 1970 年代に遡るが、漢字符号化の主流とはならず、標準的な記法も確立されて来なかった。その後、ISO/IEC 10646-1:2000 [1] においてはじめて漢字構造情報の標準記法である IDS (Ideographic Description Sequence) とそのためのオペレータ群である IDC (Ideographic Description Characters) （図 1）が定義された。

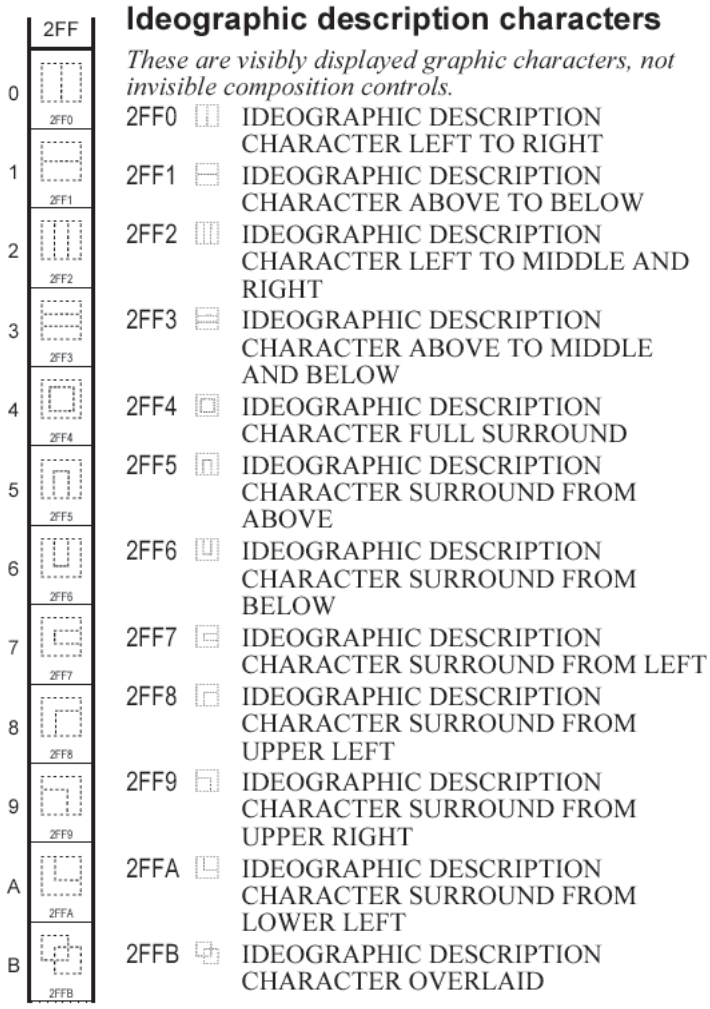


図 1: Ideographic Description Characters

IDS （図 2）は Lisp 言語における S 式と同様な前置記法の一種で、オペレータである IDC の後に 2 個ないしは 3 個の部品をとる。IDC の種類によって後に来る部品の数は決

まっているので、S 式と異なり、括弧は不要である。例えば、「村」は 2 個の部品を左右に並べるオペレータ「⿰」を使って「⿰木寸」と書くことができる。IDS では部品として IDS をとることも可能であり、入れ子状の記述が可能である。

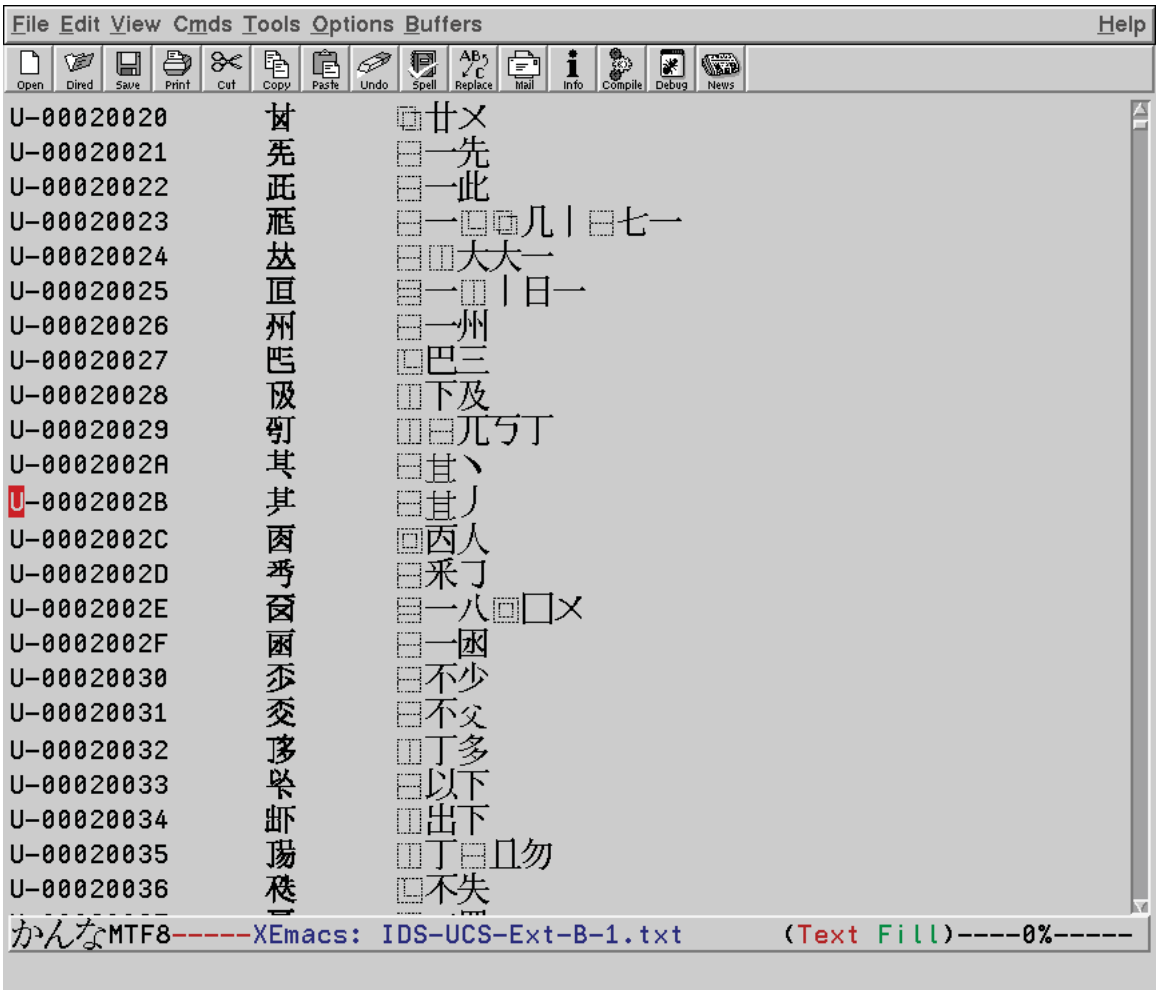


図 2: IDS の例（CHISE 漢字構造情報データベース）

3　S 式による表現

IDS は漢字構造情報を文字列として扱ったり、文字列としてパターンマッチングしたりするのには便利であるが、部品として IDS が現れる場合の処理はそのままでは少し面倒である。よって、より込み入った処理を考えれば、IDS を構文解析した結果の構文木として扱う方が便利であるといえる。そこで、CHISE 文字オントロジーの内部表現としては Lisp 言語における S 式を用いている。

IDS は Lisp 言語における S 式と同様な前置記法なので、その構文解析は極めて容易であり、オペレータの前とオペレータがとる最後の部品の後に括弧を付ければ S 式となる。

CHISE 文字オントロジーでは Chaon モデルに基づき文字素性の集合として文字を表現しているが、漢字構造情報はこの文字素性の 1 つと看做せ、`ideographic-structure` という素性名が使われている。

例えば、「峠」の ideographic-structure 素性は

```
(ideographic-structure
 ((name . "IDEOGRAPHIC DESCRIPTION CHARACTER LEFT TO RIGHT")
  (=ucs             . #x2FF0)
  )
 ((=ucs . #x5C71) ; 山
  )
 ((ideographic-structure
   ((name . "IDEOGRAPHIC DESCRIPTION CHARACTER ABOVE TO BELOW")
    (=ucs           . #x2FF1)
    )
   ((=ucs . #x4E0A) ; 上
    )
   ((=ucs . #x4E0B) ; 下
    ))
  (=ucs             . #x209D7)
  ))
```

のように表すことができる。ここで、

((素性名 . 値) ...)

は Chaon モデル的に文字を素性の集合で表現した「文字指定 (char-spec)」と呼ばれる形式で、1 文字を表す。文字と文字指定は同一視できるようになっているので、漢字構造における部品として現れる文字も ideographic-structure 素性を持つことで再帰的な表現が可能となる。

CHISE 文字オントロジーでは「文字参照 (char-ref)」と呼ばれる形式がある。これは

(:属性名 値 ... :char 文字オブジェクト)

という形の Lisp でいう所の属性リスト (property list) の一種で、:char 属性で指定した文字に他の属性で指定したメタデータを付加することができる。ideographic-structure 素性では文字オブジェクトの代わりにこの形式のデータを置くことができ、これによって部品やオペレータに出典情報や位置情報などの付加データを付随させることが可能である。但し、文字参照形式で記述した付加データは IDS（や拡張 IDS）では表現することはできないという問題があるので注意が必要である。[2]

4 RDF による表現

4.1 文字列を用いた表現

一番安直な方法は IDS の文字列をそのまま用いて、

[2]付加データを落した IDS にすることは可能である。

```
<rdf:Description
 rdf:about="http://www.chise.org/est/rdf.cgi?character=峠">
  <ids>⿰山⿱上下</ids>
</rdf:Description>
```

のように書いてしまうことである。ただ、この場合、部品や部品間のつながりが構造化されておらず問題である。

4.2 コンテナを用いた表現

RDF には複数のものをひとまとめにして扱うための仕組みとして『コンテナ』というものがある。このコンテナを使うことで、3 節で述べた S 式における表現と同様な（IDS の構文木に相当する）木構造のグラフを表現することができる。

RDF が定義しているコンテナ用の語彙には

rdf:Bag メンバーの順序は重要でないグループ

rdf:Seq メンバーの順序が重要なグループ

rdf:Alt メンバーがある指示対象の代替表現（別名や別表記等）になってる場合

rdf:li グループの要素

がある。漢字構造情報の場合、部品の順番に意味があるので、rdf:Seq を使うのが良いといえる。例えば、「峠」の場合、

```
<rdf:Description
 rdf:about="http://www.chise.org/est/rdf.cgi?character=%E5%B3%A0">
  <est:ideographic-structure
   xmlns:est="http://www.chise.org/est/rdf.cgi?domain=est/">
  <rdf:Seq>
    <rdf:li>
      <rdf:Description
       rdf:about="http://www.chise.org/est/rdf.cgi?character=%E2%BF%B0">
      </rdf:Description>
    </rdf:li>
    <rdf:li>
      <rdf:Description
       rdf:about="http://www.chise.org/est/rdf.cgi?character=%E5%B1%B1">
      </rdf:Description>
    </rdf:li>
    <rdf:li>
      <rdf:Description
       rdf:about="http://www.chise.org/est/rdf.cgi?character=%F0%A0%A7%97">
      </rdf:Description>
    </rdf:li>
  </rdf:Seq>
  </est:ideographic-structure>
</rdf:Description>
```

のように書くことができる。[3]

4.3 コレクションを用いた表現

4.2 節で述べた RDF コンテナの問題点は、記述したもの以外のメンバーが存在しないことを明示できないことである。これは記述困難な部品を持った文字の漢字構造を記述する時に、不完全な漢字構造情報が表現できるという意味では便利かも知れないが、完全な漢字構造情報を表現するのには問題である。

RDF には複数のメンバーからなるグループを表現するための仕組みとして『コレクション』というものもある。RDF コレクションは、RDF コンテナと異なり、閉じたリストを表現するもので、メンバーとして明示されたものだけからなるグループを表現することができる。

RDF コレクションは、`rdf:first` と `rdf:rest` という 2 つの語彙と定義済み資源 `rdf:nil` を使って表現するようになっている。これは S 式におけるリストの表現と同じものであり、`rdf:first` は Lisp における `car` 部、`rdf:rest` は Lisp における `cdr` 部、`rdf:nil` は Lisp における `nil` と同じものである。

これを使って、例えば、「峠」の漢字構造情報は

```
<rdf:Description
  rdf:about="http://www.chise.org/est/rdf.cgi?character=%E5%B3%A0">
   <est:ideographic-structure
    xmlns:est="http://www.chise.org/est/rdf.cgi?domain=est/"
    rdf:nodeID="cons1" />
</rdf:Description>
<rdf:Description rdf:nodeID="cons1">
   <rdf:first
    rdf:resource="http://www.chise.org/est/rdf.cgi?character=%E2%BF%B0"/>
   <rdf:rest rdf:nodeID="cons2"/>
</rdf:Description>
<rdf:Description rdf:nodeID="cons2">
   <rdf:first
    rdf:resource="http://www.chise.org/est/rdf.cgi?character=%E5%B1%B1"/>
   <rdf:rest rdf:nodeID="cons3"/>
</rdf:Description>
<rdf:Description rdf:nodeID="cons3">
   <rdf:first
    rdf:resource="http://www.chise.org/est/rdf.cgi?character=%F0%A0%A7%97"/>
   <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:Description>
```

のように表現することができる。

ただ、こうした表現方法だと、長いリストを書くのが面倒なので、S 式の場合と同様な短縮記法が用意されている。例えば、上記の例の場合、

[3] 2013 年 1 月 26 日現在、EsT[12] (CHISE-wiki) による CHISE 文字オントロジーの RDF/XML 形式での出力に用いているのはこの形式である。

```
<rdf:Description
 rdf:about="http://www.chise.org/est/rdf.cgi?character=%E5%B3%A0">
  <est:ideographic-structure
   xmlns:est="http://www.chise.org/est/rdf.cgi?domain=est/"
   rdf:parseType="Collection">
    <rdf:Description
     rdf:about="http://www.chise.org/est/rdf.cgi?character=%E2%BF%B0"/>
    <rdf:Description
     rdf:about="http://www.chise.org/est/rdf.cgi?character=%E5%B1%B1"/>
    <rdf:Description
     rdf:about="http://www.chise.org/est/rdf.cgi?character=%F0%A0%A7%97"/>
  </est:ideographic-structure>
</rdf:Description>
```

という風に表現できる。

4.4 IDC の述語化

4.3 節で述べたように RDF コレクションを使うことで、S 式の場合と同様な情報を記述することができる訳だが、この方法の場合 IDS の構文木に相当するものを表現することはできているものの、それが意味しているもの、即ち、オペレーターの意味や部品の位置関係といったものは RDF のグラフとしては表現されていないといえる。

この問題を解決するには、例えば、「峠」の場合、『「木」は「峠」の左側である』というような文を RDF で表現すれば良いといえる。つまり、IDS では部品の組合せ方に関する情報はオペレーターである IDC が担っており、4.3 節の方法ではそれを資源として表現していたが、『〜の左側である』のような相対的な位置関係を示す述語を用いる訳である。

安岡孝一氏[4]は IDS の RDF 化を検討した際にこうしたアプローチを採った。以下、安岡氏のメモ「IDS は RDF で書けるのか」から引用する：

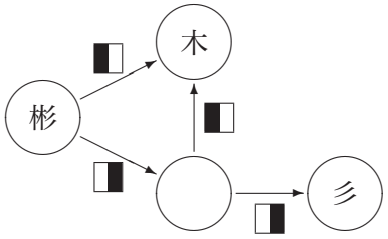
> ナイーブには、RDF の主語と目的語をいずれも「漢字」とし、述語を IDC とすれば、書ける気がする。しかし、そのような形式にした場合、IDC は目的語を複数とるのに対し、RDF の目的語は 1 つ (複数書くことも可能だが、それは単なる並置であって順序関係がない) なので、そのまま直感的に記述することができない。[5]
>
> この問題を解決するには、たとえば U+2FF0 ⿰ を ◧ と ◨ に分けて、それぞれを述語とすることが考えられる。すなわち「杉」に対する「 ⿰ 木彡」という IDS であれば
>
> 杉 ◧ 木 .
> 杉 ◨ 彡 .
>
> と RDF 表現すればいいわけだ。

[4]京都大学人文科学研究所附属東アジア人文情報学研究センター准教授。漢字コード屋さんにしてタイプライター史家

[5][守岡注] 4.2 節や 4.3 節で述べたように、順序関係を持った複数の要素からなるグループを目的語として記述することは可能である。

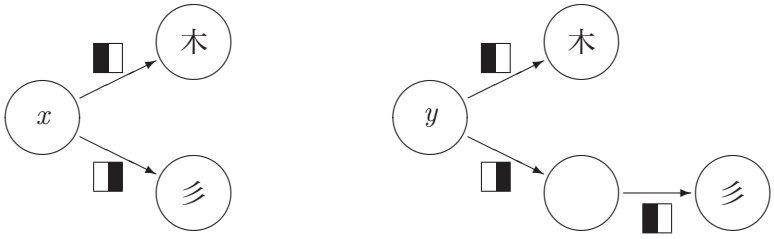
これで漢字構造情報を『直観的』に RDF で記述できるような気がする。しかし、漢字の部品には複数の漢字を部品として組み合わせたものもあり、漢字構造情報には再帰的な構造が存在している。このため、ナイーブにこの方式を適用すると、ループ構造が生じてしまうのである。また、IDC には『横に部品を３つ並べる』ことを意味するものや『縦に部品を３つ並べる』ことを意味する引数が３つのオペレーターが存在するが、安岡氏は『〜の中である』ことを意味する述語を用いる代わりに、２引数のオペレーターを使った形に正規化するアプローチを採った。以下、再び安岡氏のメモ「IDS は RDF で書けるのか」から引用する：

> ただし、このやり方では「彬」を ◧ と ◨ で表現する場合に、ひと工夫必要となる。一意性を保っておかないと、ヤヤコシイことになりかねないからだ。そのために、たとえば「左から切っていく」というアプローチを、ここでは仮に採用しておくことにしよう。すなわち
>
> 彬 ◧ 木 .
> 彬 ◨ 杉 .
>
> という RDF 表現だけを許す、というやり方だ。少し厳密な言い方をすれば、述語 ◧ の目的語は、さらに述語 ◧ を取り得るようなものであってはならない、という制限をかけることになる。
>
> ちなみに U+2FF2 ⿲ で書かれた IDS に対しても、 ◧ と ◨ の組合せで RDF を書くことにすれば、大丈夫な気がする。たとえば「彬」に対する「⿲木木彡」という IDS は、匿名ノードを使って
>
> 彬 ◧ 木 .
> 彬 ◨ [◧ 木], [◨ 彡] .
>
> と RDF 表現すれば、たぶん問題ないと思う。



こうして、木構造であるはずの IDS がループ構造を持つグラフに化けてしまった訳である。こうなってしまうと検索も面倒なことになってしまう。以下、また安岡氏のメモ「IDS は RDF で書けるのか」を引用する：[6]

> ただ、このやり方を採った場合、たとえば「杉」を部品に含む漢字の検索は、多少ヒネリが必要となる。「杉」に到達可能な有向 IDS-RDF グラフが答となるのは当然だが、それに加えて



[6]最初の引用からこの引用を合わせたものがメモの全文である（守岡が付けた注は除く）。

を満たす x や y (いずれも匿名ノードの可能性がある) に到達可能な有向 IDS-RDF グラフも答となりうるのだ。この点に注意して処理系を組む必要があるだろう。

直観的に表現しようとしたはずなのに、なんでこんなややこしいことになってしまったのだろう？

4.5 『文字概念』と『文字の出現』

4.4 節で述べた安岡氏の方法がおかしな結果になってしまったのは、結論からいえば、抽象概念としての文字と相対的な位置情報を持った（いわば、『文字概念』のインスタンスとしての）文字を区別せずに、『文字概念』に直接リンクを張ってしまったことに起因するといえる。もう一度、3 節の S 式による表現と、4.2 節と 4.3 節の RDF による表現を注意深く見直してみれば、文字と部品が直接リンクしておらず、入れ子になる場合も `est:ideographic-structure` という述語（`ideographic-structure` 素性）やコンテナ等を介して間接的につながっていることに気づくかも知れない。

判り易くするために、4.4 節で安岡氏が挙げた「彬」の例を 4.2 節で述べた RDF コンテナを用いる方法で記述した場合のグラフを図 3 に示す。

4.4 節の「彬」のグラフと同様にループ構造は生じているが、`ideographic-structure` 素性の値となる `rdf:Seq` コンテナがその型情報と共に要素の位置情報を保持しており、それらの位置情報で示されるコンテナ中の場所の中身として文字オブジェクトが参照されている。こうした参照関係をコンテナの中の子箱にオブジェクトが入っているように看做せば、この RDF は本質的に木構造を表現しているという風に解釈することができる（図 4）。[7]

もう少し概念的な問題として考えてみれば、例えば、抽象的な概念としての『木』と紙に書かれた文字や印刷された文字（ここにある「木」）はその存在論的な位置が異なっていることに気づくと思う。UCS のコードポイント U+6728 や JIS X 0208 の区点 44-58 が指し示しているのは、かつてこの世に存在し、また、これから紙に書かれたり、印刷されたり、ディスプレイに映し出されたり、その他各種デバイスで表示されたり、将来的には脳内に直接伝達されたりされなかったりするかも知れない、すべての宇宙、過去と未来のすべての『木』という漢字なのである！

こうしたものを文字符号の世界では『抽象文字』と呼ぶ。抽象文字は具体的な文字の形を持たず、ある範囲の字体を包摂する。これに対し、紙に書かれた文字や印刷された文字、あるいは、ディスプレイに映し出されたり、その他各種デバイスで表示された文字は『字形』と呼ばれる。字形は具体的な形を持つ抽象文字のインスタンスの一種といえる。

ただ、IVS (Ideographic Variation Sequence)[3] に関する議論に見られるように、ある程度具体的な形を指示しつつ、紙やディスプレイといった文字を表現するデバイス上での具体的な形を持たないもの、いわば、『字形レベルの文字概念』とでもいうものも文字を考える場合に考えなければならない概念のひとつである。ただ、ややこしいことに、これもまた、しばしば、「字形」と呼ばれたりもする。[8]

[7] (a (a b)) のようなリストをループ構造ではなく木構造だと解釈するのと同様である。
[8] グリフと呼ばれることもあるが、こちらもいろんな使われ方をしがちである。

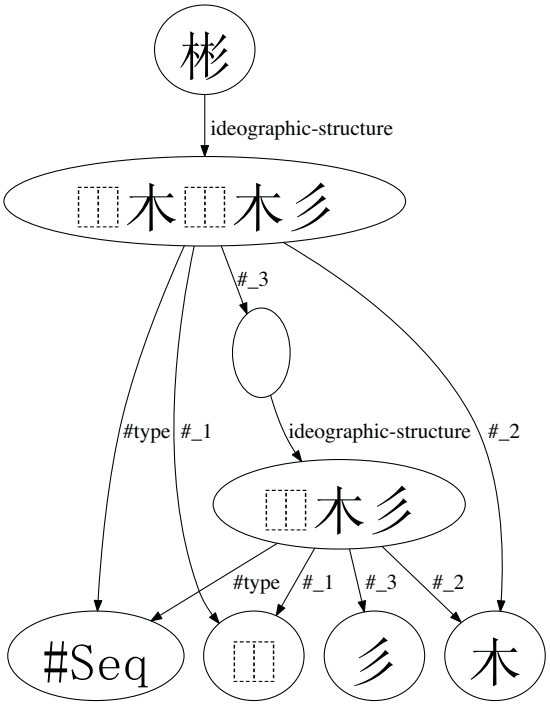


図 3: RDF コンテナを使って「彬」を書いた場合

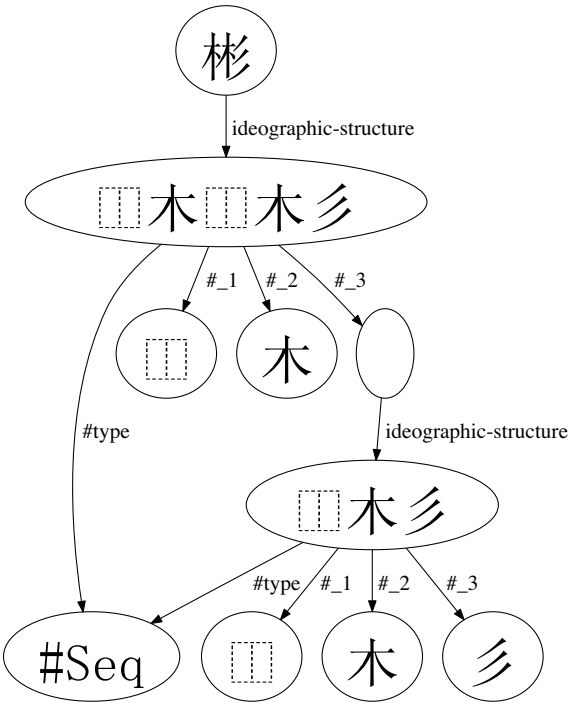


図 4: 図 3 のグラフが表現する木構造

具体物上の『字形』は具体的な形を持つだけでなく、その具体物上の具体的な位置を占めている。これに対し、ある IVS が表現するものは実際に紙やディスプレイといった物理的なデバイス上に書かれたり、印刷されたり、表示されたりしない限り、具体物上の具体的な位置を持たない。その意味では、抽象文字と同様である。もし、URI のような ID を付与するとすれば、[5] 前者は具体物に振られた ID とその中での相対的な位置、例えば、本の場合、本が存在する場所とページ番号とカラム番号と行番号と行中の何文字かというような情報によって一意に指示することができるだろう。一方、後者はそういう訳にはいかない。物理的な実体ではないからである。もちろん、印刷物の場合、同じ版面から印刷されたものはそれぞれの具体物中の同じ相対位置にほぼ同じ形の文字が書かれているはずである。とはいえ、紙の繊維が微妙に違ってたり、汚れたり折り目が付いてたりして、同じ場所をスキャンしても異なった画像になってしまうかも知れない。つまり、ある種の『理想化』を行わないと、『字形レベルの文字概念』にならないといえる。[9]

漢字構造情報は、その全体としては、『字形レベルの文字概念』と同様に、具体物に紐づけられていない（物理世界の時空上の座標位置を持たない）抽象的な存在である。しかしながら、その中の部品は漢字構造中の相対的な位置関係を持つといわざるを得ない。即ち、漢字構造情報は『字形レベルの文字概念』や抽象文字と（それらが指し示す形状の具体度＝包摂範囲の差を無視すれば）存在論的に同種のものだといえ、その属性の中に位置情報を含まないものといえる。しかし、漢字の中の部品に関する情報にはその属性のひとつとして位置情報が必要なのである。よって、仮に形状が一緒だとしても（そして、同じ漢字概念を示していたとしても）、位置情報が異なれば違うオブジェクトだと考えなければならない。

このことは「『杉』の左側の部品と『松』の左側の部品は同じ形をしているが、『木』とは形が異なる。これは偏化変形の一種である」という文が書けるかという問題を考えれば判り易いかも知れない。4.4 節の表現の場合、『「彬」の左側の部品』と『「彬」の真中の部品』の区別がなく、全て『抽象文字の「木」』になっている。このため、「「彬」の左側の部品と「彬」の真中の部品は形が似ている」という文が書けない訳である。

こうした性質の差異を表現するために、具体物に紐づけられていない（物理世界の時空上の座標位置を持たない）抽象的な存在としての文字を『文字概念』と呼ぶことにする。これは [10] における『グリフ』と同様に、それが指し示す形状の具体度＝包摂範囲の差に関係なく、抽象文字や字体、『字形レベルの文字概念』といったものは全て『文字概念』の一種と考えることにする。一方、何らかの位置情報で示される場所に現れた『文字概念』のインスタンスのようなものを『文字の出現』(character occurrence) と呼ぶことにする。[10] このモデルに従えば、字形や漢字の中の部品は『文字の出現』の一種になる。

---

[9] 例えば、『大漢和番号 12345 の字形』は、多分、この世に存在する全ての大漢和辭典の大漢和番号 12345 の字形の平均画像を意味しないと思われる。なぜなら、ある本には折り目が付いてるかも知れないし、ある本には繊維にゴミが付いてるかも知れないし、ある本には落書や汚れが付いてるかも知れない。ここで想定されているものは、究極的には、『理想的な紙に理想的に印刷され理想的に保存された大漢和番号 12345 の字形』というこの世には実際に存在しないものだといえる。

[10] [15] では 7.2 節「もう一つのインスタンス問題：表現のオントロジー」において、文字の問題に関しても議論している。ここでは文字を文章や絵画、小説、音楽などの「表現」の一種とする立場を採っており、文字の場合、「実体」である二次元図形と、「表現」の一種である『（書かれた）文字図形』、「内容」としての『文字の典型形状の仕様』、および、「表現物」としての紙に書かれた文字を区別している。ここでの議論はやや文字の視覚的記号としての側面に偏っているようにも思えるが、概ね、『（書かれた）文字図形』は『文字の出現』、『文字の典型形状の仕様』は『文字概念』、『紙に書かれた文字』は字形に相当するものと考えられる。

4.6 部品概念を採り入れた IDC の述語化

4.4 節と 4.5 節での議論を踏まえれば、漢字、漢字構造情報、『文字の出現』の一種としての漢字の中の部品を別のオブジェクトとして扱いつつ、『IDC の述語化』を行った方法を採るのが望ましいと考えられる。

4.6.1 『IDC の述語化』の改良

『IDC の述語化』には、4.4 節で紹介した安岡氏の方法のように、3 引数オペレーターを 2 引数のものに分解・正規化する方法も考えられるが、ここでは IDS が表現する情報を保存できるように、分解・正規化を行わずに述語化することにする。

ここでは、IDC の『(文字の) 名前』 (character name) を手がかりに関係素性名を決めることにする。

例えば、U+2FF0 の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER LEFT TO RIGHT>

であるが、この内の用字系 (script) を示す接頭辞 IDEOGRAPHIC DESCRIPTION CHARACTER を除いた部分は LEFT TO RIGHT であり、ここから、関係素性

->connect-left, ->connect-right

を生成することができる。

同様に、U+2FF1 の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER ABOVE TO BELOW>

であるので、関係素性

->connect-above, ->connect-below

が生成できる。

U+2FF2 の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER LEFT TO MIDDLE AND RIGHT>

であるので、関係素性

->connect-left, ->connect-middle, ->connect-right

が生成できる。

同様に、U+2FF3 の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER ABOVE TO MIDDLE AND BELOW>

であるので、関係素性

->connect-above, ->connect-middle, ->connect-below

が生成できる。
　ただ、この場合、横に並べた場合の真中と縦に並べた時の真中が区別できないのは問題かも知れない。そこで、前者を

　　->connect-middle@horizontal,

後者を

　　->connect-middle@vertical

とすることにする。
　U+2FF4 の場合、その名前は

　　<IDEOGRAPHIC DESCRIPTION CHARACTER FULL SURROUND>

であるが、前述の方法のままでは適用できないので、

　　囲む部分を示す関係素性を ->enclosure@full

　　囲まれる部分を示す関係素性を ->surrounded@full

とすることにする。[11]
　U+2FF5 の場合、その名前は

　　<IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM ABOVE>

であるので、前述の方法に準じて、

　　囲む部分を示す関係素性を ->enclosure@from-above

　　囲まれる部分を示す関係素性を ->surrounded@from-above

とすることにする。
　同様に、U+2FF6 の場合、その名前は

　　<IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM BELOW>

であるので、

　　囲む部分を示す関係素性 ->enclosure@from-below

　　囲まれる部分を示す関係素性 ->surrounded@from-below

が生成できる。
　同様に、U+2FF7 の場合、その名前は

　　<IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LEFT>

であるので、

[11] GlyphWiki の偏化変形修飾子では囲い系部品を示す修飾子の分類が IDC のと異なっているため、GlyphWiki との対応を考慮して、囲み系の関係素性名にドメインを用いている。[13]

囲む部分を示す関係素性 ->enclosure@from-left

囲まれる部分を示す関係素性 ->surrounded@from-left

が生成できる。

同様に、U+2FF8 の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM UPPER LEFT>

であるので、

囲む部分を示す関係素性 ->enclosure@from-upper-left

囲まれる部分を示す関係素性 ->surrounded@from-upper-left

が生成できる。

同様に、U+2FF9 の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM UPPER RIGHT>

であるので、

囲む部分を示す関係素性 ->enclosure@from-upper-right

囲まれる部分を示す関係素性 ->surrounded@from-upper-right

が生成できる。

同様に、U+2FFA の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LOWER LEFT>

であるので、

囲む部分を示す関係素性 ->enclosure@from-lower-left

囲まれる部分を示す関係素性 ->surrounded@from-lower-left

が生成できる。

U+2FFB の場合、その名前は

<IDEOGRAPHIC DESCRIPTION CHARACTER OVERLAID>

であるが、便宜的に、

上書きされる（主要な）部分を示す関係素性 ->underlying

上書きする（付加的）部分を示す関係素性 ->overlaying

を用いることにする。

4.6.2 IDC のための型の導入

IDC を型としても表現するために、IDC の名前の接頭辞を除いた部分から機械的に作成した語彙を用いることにする。例えば、U+2FF0 の場合は chise.ids:left-to-right となり、U+2FFA の場合は chise.ids:surround-from-lower-left となる。

4.6.3 『文字の出現』の RDF 化

『文字の出現』を『文字概念』(= 文字オブジェクト) と同格なものとして扱うためには、『文字概念』と同様な情報資源として表現すれば良い。また、『文字の出現』と『文字概念』の間の関係を示す述語として chise.ids:to.content (->content 素性) を導入する。

これらを使って、例えば、「彬」の漢字構造情報は

```
<rdf:Description
 rdf:about="http://www.chise.org/est/rdf.cgi?character=彬">
  <est:ideographic-structure
   xmlns:est="http://www.chise.org/est/rdf.cgi?domain=est/">
    <chise.ids:left-to-right
     xmlns:chise.ids="http://www.chise.org/est/rdf.cgi?domain=chise.ids/">
      <chise.ids:to.connect-left>
        <rdf:Description
         rdf:about="http://www.chise.org/est/rdf.cgi?character-occurrence=彬/left">
          <chise.ids:to.content>
            <rdf:Description
             rdf:about="http://www.chise.org/est/rdf.cgi?character=木">
            </rdf:Description>
          </chise.ids:to.content>
        </rdf:Description>
      </chise.ids:to.connect-left>
      <chise.ids:to.connect-right>
        <rdf:Description
         rdf:about="http://www.chise.org/est/rdf.cgi?character-occurrence=彬/right">
          <chise.ids:to.content>
            <rdf:Description
             rdf:about="http://www.chise.org/est/rdf.cgi?character=杉">
              <est:ideographic-structure
               xmlns:est="http://www.chise.org/est/rdf.cgi?domain=est/">
                <chise.ids:left-to-right>
                  <chise.ids:to.connect-left>
                    <rdf:Description
                     rdf:about="http://www.chise.org/est/rdf.cgi?character-occurrence=杉
/left">
                      <chise.ids:to.content>
                        <rdf:Description
                         rdf:about="http://www.chise.org/est/rdf.cgi?character=木">
                        </rdf:Description>
                      </chise.ids:to.content>
                    </rdf:Description>
                  </chise.ids:to.connect-left>
                  <chise.ids:to.connect-right>
                    <rdf:Description
                     rdf:about="http://www.chise.org/est/rdf.cgi?character-occurrence=杉
/right">
                      <chise.ids:to.content>
                        <rdf:Description
                         rdf:about="http://www.chise.org/est/rdf.cgi?character=彡">
                        </rdf:Description>
                      </chise.ids:to.content>
                    </rdf:Description>
                  </chise.ids:to.connect-right>
                </chise.ids:left-to-right>
```

```
            </est:ideographic-structure>
          </rdf:Description>
        </chise.ids:to.content>
      </rdf:Description>
    </chise.ids:to.connect-right>
  </chise.ids:left-to-right>
</est:ideographic-structure>
</rdf:Description>
```

のように書くことができる。図 5 にこの RDF グラフを示す。

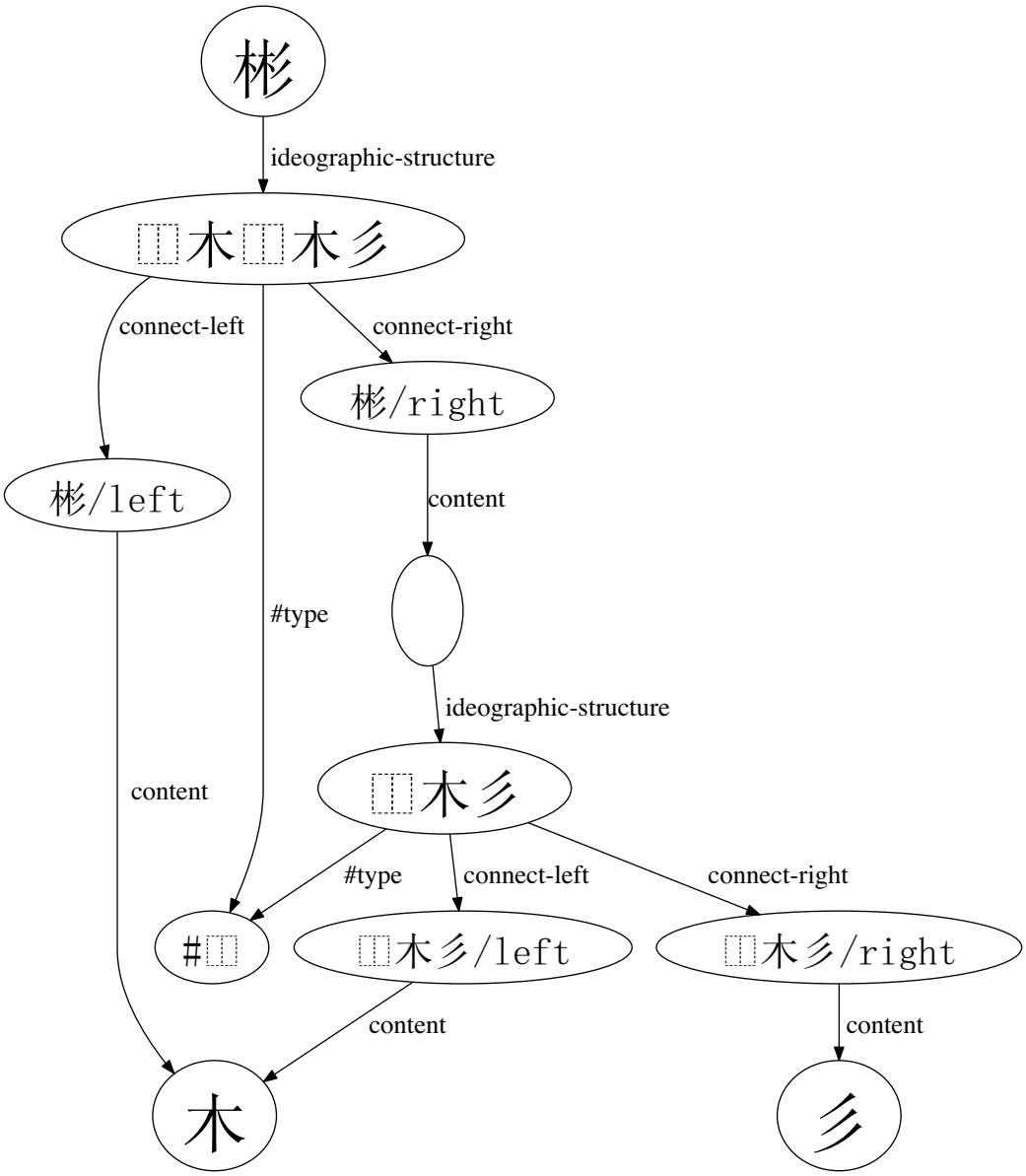


図 5: 『文字の出現』オブジェクトを導入した場合の「彬」のグラフ

4.6.4 IDS 用コンテナを使った簡略化

4.6.3 節の例を見れば判るように、『文字の出現』をオブジェクト化した結果、漢字構造情報の RDF はかなり複雑なものとなってしまった。

しかしながら、4.5 節で述べたように、4.2 節で述べた RDF コンテナを使った方法であっても、コンテナが要素の位置情報を保持しており、実質的に『文字の出現』をオブジェクト化した場合と同様の情報を表現できていた訳である。

そこで、IDC に相当する RDF コンテナを導入し、『文字の出現』における位置の情報をコンテナ側に持たせることで、簡略化することを考える。

例えば、「彬」の漢字構造情報は

```
<rdf:Description
 rdf:about="http://www.chise.org/est/rdf.cgi?character=彬">
  <est:ideographic-structure
   xmlns:est="http://www.chise.org/est/rdf.cgi?domain=est/">
    <chise.ids:left-to-right
     xmlns:chise.ids="http://www.chise.org/est/rdf.cgi?domain=chise.ids/">
      <chise.ids:to.connect-left>
        <rdf:Description
         rdf:about="http://www.chise.org/est/rdf.cgi?character=木">
        </rdf:Description>
      </chise.ids:to.connect-left>
      <chise.ids:to.connect-right>
        <rdf:Description
         rdf:about="http://www.chise.org/est/rdf.cgi?character=杉">
          <est:ideographic-structure
           xmlns:est="http://www.chise.org/est/rdf.cgi?domain=est/">
            <chise.ids:left-to-right>
              <chise.ids:to.connect-left>
                <rdf:Description
                 rdf:about="http://www.chise.org/est/rdf.cgi?character=木">
                </rdf:Description>
              </chise.ids:to.connect-left>
              <chise.ids:to.connect-right>
                <rdf:Description
                 rdf:about="http://www.chise.org/est/rdf.cgi?character=彡">
                </rdf:Description>
              </chise.ids:to.connect-right>
            </chise.ids:left-to-right>
          </est:ideographic-structure>
        </rdf:Description>
      </chise.ids:to.connect-right>
    </chise.ids:left-to-right>
  </est:ideographic-structure>
</rdf:Description>
```

のように表現する訳である。図 6 にこの RDF グラフを示す。

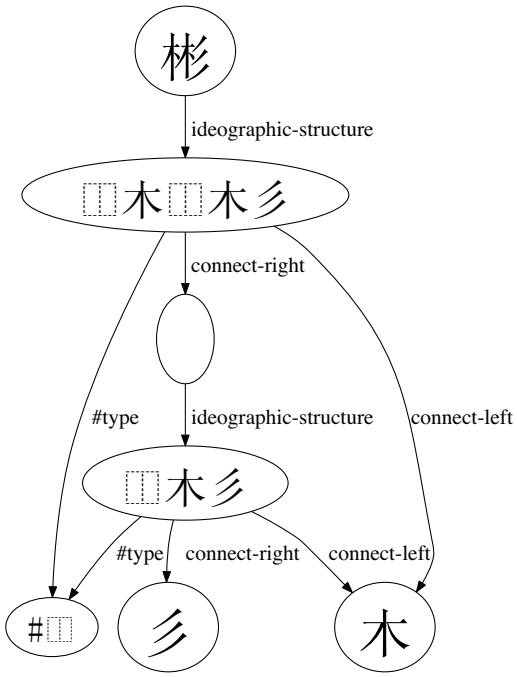


図 6: IDC コンテナを使った簡略化を行った場合の「彬」のグラフ

4.7　包摂範囲の異なる文字概念間の関係

これまで挙げた例では、簡単化のために、漢字構造情報中の部品は全て UCS の抽象文字相当のものとして扱ってきたが、字体レベルのものや字形レベルのものといったより包摂範囲の狭い（より具体的な形状を示す）『文字概念』オブジェクトを部品としても良い。実際、CHISE 文字オントロジー中では包摂範囲の異なるものを部品として用いた漢字構造情報が用いられており、部品の包摂範囲を使って『文字概念』が示す包摂範囲を示すことができる。

5　おわりに

漢字構造情報を RDF を用いた表現法について議論し、

- 文字列を用いた表現
- RDF コンテナを用いた表現
- RDF コレクションを用いた表現
- 安岡メモでの表現
- 『文字の出現』オブジェクトを用いた表現
- IDC コンテナを用いた簡略化

という 6 種類の方法を実例を挙げて比較検討を行った。

漢字構造情報を RDF を用いて表現する場合、IDS の構文木に相当する情報が十分に保持されていることと、それが意味するものが RDF のグラフとしてきちんと表現されていることが望ましい。そのためには、

- IDC の述語化・型の導入
- 抽象文字や字形、グリフといったものを、包摂範囲の差に基づく軸と位置情報の有無に関する軸で整理
  - 後者の軸に基づき、

    『文字概念』位置情報を持たないもの（デバイスに紐づけられていないもの；表現としての文字の『内容』; 概念としての文字の『設計図』のようなもの）

    『文字の出現』位置情報を持つもの（文字が置かれた場に紐づけられている；文字の『表現』）

    という概念を導入
- 『文字の出現』を通常の資源、もしくは、IDC コンテナで表現

といった手法を採るのが良いと考えられる。

本稿では漢字構造情報の RDF に関わる問題に関してなるべく網羅的に取り上げることを試みたが、『多粒度漢字構造情報』の RDF 化等の幾つかの問題については、残念ながら今回は取り上げることができなかった。こうした問題に関してはまた別稿で取り上げたい。

最後に、本稿を書くきっかけを与えて頂いた、山崎直樹氏と安岡孝一氏に感謝する。特に、安岡孝一氏には氏のメモ「IDS は RDF で書けるのか」をご教示頂いたばかりか、その LaTeX のソースコードを提供して頂き、その利用をお許し頂いたことは、そこで提案された『IDC の述語化』というアイディアとともに、本稿を書く上での大きな助けとなった。

参考文献

[1] International Organization for Standardization (ISO). *Information technology — Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane (BMP)*, 2000 年 3 月. ISO/IEC 10646-1:2000.

[2] International Organization for Standardization (ISO). *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*, 2012 年 6 月. ISO/IEC 10646:2012.

[3] Ken Lunde, Richard Cook, and John H. Jenkins (ed.). Unicode Ideographic Variation Database. Unicode Technical Standard #37, 2011 年 11 月. Revision 8.

[4] Tomohiko Morioka. CHISE: Character processing based on character ontology. In *Large-scale Knowledge Resources (LKR2008)*, No. 4938 in LNAI, pp. 148–162, 2008 年 3 月.

[5] Henry S. Thompson. Web Architecture and Naming for Knowledge Resources. In *Large-scale Knowledge Resources (LKR2008)*, No. 4938 in LNAI, pp. 334–343, 2008 年 3 月.

[6] World Wide Web Consortium, http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/. *RDF/XML Syntax Specification (Revised)*, 2004 年 2 月.

[7] World Wide Web Consortium, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/. *Resource Description Framework (RDF): Concepts and Abstract Syntax*, 2004 年 2 月.

[8] 守岡知彦. CHISE IDS 漢字検索. `http://www.chise.org/ids-find`.

[9] 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 年 10 月. 2006-CH-72.

[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, 第 2009 巻, pp. 9–14. 情報処理学会, 情報処理学会, 2009 年.

[11] 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 年 7 月.

[12] 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 —人文工学の可能性〜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, 第 2010 巻, pp. 33–40. 情報処理学会, 情報処理学会, 2010 年 12 月.

[13] 守岡知彦. 漢字字形データベースと文字オントロジーの データ統合の可能性について. 情処研報, Vol. 2012-CH-94, No. 8, pp. 1–8, 2012 年 5 月.

[14] 守岡知彦, クリスティアン・ウィッテルン. 文字データベースに基づく文字オブジェクト技術の構築. 情報処理振興事業協会 平成 13 年度 成果報告集. 情報処理振興事業協会, 2002 年. `http://www.ipa.go.jp/NBP/13nendo/reports/explorat/charadb/charadb.pdf`.

[15] 溝口理一郎. オントロジー工学. 知の科学. オーム社, 2005 年 1 月.

国立国会図書館のメタデータ標準

－データを繋げるメタデータ：DC-NDL－

柴田 洋子（国立国会図書館）
yshibata@ndl.go.jp

要旨：知識や情報と利用者を的確に結びつけることは、図書館の重要な役割の一つである。国立国会図書館では、膨大な情報資源の中から利用者が適切な情報を発見・識別できる手段として、また、図書館をはじめ様々な機関が所有するデータを繋げ、その利用可能性を向上させる枠組みとして、メタデータ標準 DC-NDL を定めている。本稿では、セマンティックウェブに対応した DC-NDL のコンセプトや策定経緯、スキーマ設計等について紹介する。
キーワード：メタデータ，セマンティックウェブ，Dublin Core

1. はじめに

知識や情報と利用者を的確に結びつけることは、図書館の重要な役割の一つである。国立国会図書館（NDL）は、国の唯一の納本図書館・保存図書館として、国内で刊行される出版物を広く収集し、貴重な文化的資産として保存するとともに、知識・文化の基盤として誰もが利用できる環境・手段を提供することを使命としている。これらは、図書や雑誌といった印刷出版物にとどまらず、インターネット上で流通する様々な情報資源も含まれる。膨大な情報資源の中から利用者が適切な情報を発見できる手段として、その情報資源に関するデータ、すなわちメタデータが重要である。そのため、NDL では、媒体に拠らず多様な情報資源を適切に整理（組織化）し、利用に供するためのメタデータの枠組みとして「国立国会図書館ダブリンコアメタデータ記述（DC-NDL）」[1]を定め、これに準拠したメタデータを作成・提供している。

本稿では、2 章で DC-NDL の概要、コンセプト、3 章で Dublin Core と NDL の動向からみた DC-NDL の変遷、4 章で概念モデルやフォーマット等の仕組みを解説し、5 章で今後の取組みについて述べる。

[1] 以下、特に版表示等がない場合、DC-NDL は最新版の DC-NDL 2011 年 12 月版を表わす。

2. 枠組み

2.1. DC-NDL とは

DC-NDL は、NDL が採用するメタデータの記述語彙と記述規則の総称である。国内の図書館や関連機関等におけるデータの交換・共有にも利用できるメタデータ標準としてだけでなく、その名が示すように、国際的なメタデータ標準である Dublin Core に基づき、国際的なメタデータの流通に寄与することも志向している。そのため、Dublin Core 等の国際的なメタデータ標準で定義された語彙に加え、日本語の読みへの対応等、NDL が組織化を行う際に必要な語彙を独自で定義して採用している。また、各語彙について、NDL における標準的なメタデータの記述方法も定めており、以下の三部で構成されている[2]。

「第一部　NDL Metadata Terms」

NDL が独自に定義した語彙集。

例）

- タイトルや作成者に対する読み "dcndl:transcription"
- 図書館資料の特性に応じた語彙（博士論文の学位分野名 "dcndl:degreeName"、報告番号 "dcndl:dissertationNumber"等）
- 標準的な統制語彙・分類体系（国際標準図書番号"dcndl:ISBN"、国立国会図書館件名標目表"dcndl:NDLSH"等）

「第二部　Application Profile」

NDL がメタデータを作成する際に用いる語彙の標準的な記述方法について定めたもの。既存の語彙及び独自で定義した語彙の NDL における使用法、値の記述形式、表現例等を説明。

「第三部 RDF スキーマ」

NDL が独自に定義した語彙をセマンティックウェブにおける標準的なメタデータの表現方法である RDF（Resource Description Framework）形式で記述したファイル。「第一部 NDL Metadata Terms」の機械可読版に相当。

2.2. セマンティックウェブ志向

セマンティックウェブは、インターネット上の情報資源に明確な意味を持つデータを付与し、コンピュータがその意味を機械的に処理できる次世代ウェブの構想及びその実現に向けた取組みである。セマンティックウェブ技術の活用により、利用者は、インターネット上の膨大な情報資源の中から自身の求める情報に的確にたどり着くことができるようになる。

そのため、DC-NDL のメタデータスキーマは RDF を用いている。しかし、DC-NDL

[2] http://www.ndl.go.jp/jp/aboutus/standards/meta.html（参照 2013-01-11）

を採用する機関やシステムで必ずしもRDF形式のメタデータに対応できるとは限らないため、RDF形式で表現できない場合でも使用できる語彙もあわせて定義している。

加えて、メタデータの語彙もRDF形式で定義している。語彙を定義する際、データ項目に独自の名称をつけただけでは、コンピュータはその意味を理解することができない。そのため、その語彙の持つ意味を機械処理できる形式で表現する必要がある。そこで、RDF を用いて語彙を定義し、語彙の記述対象となる情報資源（リソース）の範囲を指定する定義域、語彙の値が取り得る範囲を指定する値域をそれぞれ設定するとともに、Dublin Core で定義された主要な語彙と関連づけ、これらをRDFスキーマとして定義・公開している。

2.3. 相互運用性の向上

NDLでは、自館のみならず、国内の図書館や関連機関等においてもDC-NDLの利用を推進している。そこで、メタデータの相互運用性を向上するための基礎的なモデルであるアプリケーション・プロファイル[3]の考え方を取り入れ、語彙の定義（セマンティクス）と語彙の記述方法・記述形式（シンタックス）を分けて定義している。アプリケーション・プロファイルは、Dublin Coreが提唱している枠組みであり、後者のシンタックスに該当し、メタデータを記述する際にどのような語彙を使用し、どのような形式で記述するかを定義した記述規則ともいえる。この中では、使用する語彙を必ずしもDublin Coreで定義されているものに限定する必要はなく、その他のメタデータスキーマで定義されている既存の語彙を流用することも、新たに独自で定義した語彙を使用することもできる。できるだけ既存の標準的な語彙を再利用することで、国際的な語彙の共有を促進することができ、相互運用性を高めることができる。

DC-NDL では、NDL Metadata Terms がセマンティクスに相当し、Application Profile がシンタックスに相当する。既存の語彙では表現できない要素についてのみNDL Metadata Terms として定義しており、日本語の特性への配慮や国内で普及している統制語彙や分類体系への対応を行っている。後者の Application Profile は NDLにおける標準的な記述規則であり、特定のシステムの実装に基づくものではないが、これを公開することで、他の図書館や関連機関等が自機関のアプリケーション・プロファイルを検討する際の参考となることも企図している。

2.4. Dublin Core への準拠

NDL では、国内にとどまらず国外の機関とのメタデータの交換・共有を促進するため、国際的に普及しているDublin Coreへの準拠に努めている。これにより、言語の壁を超えて世界と繋がり、より幅広くデータが活用されることが期待される。DC-NDL

[3] DC-NDLの「第二部 Application Profile」と区別するため、Dublin CoreにおけるApplication Profileは「アプリケーション・プロファイル」と以下表記する。

に関しては、実際は独自に拡張した要素を少なからず含むため、シンプルとは言い難いとの指摘もあるが、Dublin Core の語彙自体はそのままの形で利用し、その基本的な考え方や枠組みを受け継ぐ姿勢を明示するため、「国立国会図書館ダブリンコアメタデータ記述（DC-NDL）」と呼称している。

3. 歩み

DC-NDL の祖は、2001 年 3 月に策定した「国立国会図書館メタデータ記述要素（NDL メタデータ）」に遡る。以降、国内外のメタデータを取り巻く状況を適宜反映しながら、3 度の改訂を経て現在の DC-NDL に至る。その変遷について、NDL における電子図書館サービスの進展と Dublin Core の動向に沿って述べる。

3.1. 国立国会図書館メタデータ記述要素（NDL メタデータ）（2001 年 3 月）

NDL メタデータは、今後の NDL における電子図書館構築のあり方をまとめた「国立国会図書館電子図書館構想」[4]（1998 年 5 月）に基づき、NDL がウェブサイトや電子雑誌等のネットワーク系電子出版物を組織化する際に用いるメタデータ（書誌情報）の基準として策定された。記述要素には、次の理由から Dublin Core Metadata Element Set（DCMES）[5]と呼ばれる Dublin Core の基本 15 要素を採用した。

- 国際的な標準化を推進しており、普及しているメタデータである
- リンク情報や権利関係のための記述要素があり、インターネット上の情報資源の記述に利点がある
- 簡便であり、作成者や出版者自身がコンテンツ（一次情報）そのものにメタデータを付与できる

基本 15 要素に加え、各要素の意味内容を補完し、より詳細にメタデータを組織化するため、Dublin Core で推奨された限定子の一部と、NDL が独自で設定した限定子を使用した。

許諾を得たウェブサイトや電子雑誌を収集・保存する実験プロジェクト「国立国会図書館インターネット資源選択的蓄積実験事業（WARP）」（2002 年）やインターネット上のデータベースを組織化したリンク集「国立国会図書館データベース・ナビゲーション・サービス（Dnavi）」[6]（同年）等、主に NDL の新たな電子図書館サービスで採用された。

3.2. 国立国会図書館ダブリンコアメタデータ記述要素（DC-NDL 2007 年版）

NDL メタデータ策定後、前述の WARP が「国立国会図書館インターネット情報選択的蓄積事業」として 2006 年から本格事業化した。また、博物館・美術館等の文化機関

[4] http://www.ndl.go.jp/jp/aboutus/elib_plan.html（参照 2013-01-11）
[5] http://dublincore.org/documents/dces/（参照 2013-01-11）
[6] http://dnavi.da.ndl.go.jp/（参照 2013-01-11）

や学術機関等が提供しているデジタルコンテンツを一元的に検索できる「国立国会図書館デジタルアーカイブポータル（PORTA）」が 2007 年に公開された。これにより、NDL におけるデジタルアーカイブの構築が進展し、組織化の対象として扱うべき情報資源の種類や特性もさらに多様化した。

一方、Dublin Core についても新たな動きが見られた。まず、2003 年に DCMES が国際規格化（ISO 15836）され、名実ともに国際的なメタデータ標準となった。さらに、2005 年には国内標準（JIS X 0836）としても定められた。

また、新たな技術への対応として、特にセマンティックウェブへの取組みが進み、Dublin Core の概念的な構造を示した DCMI Abstract Model（DCMI 抽象モデル）[7]が提示され（2005 年）、その後、改訂したモデルが公開された（2007 年）[8]。

これらを踏まえ、NDL では、国内機関におけるメタデータ作成の実践を視野に入れ、2006 年度から NDL メタデータの改訂作業に着手し、2007 年 5 月に「国立国会図書館ダブリンコアメタデータ記述要素（DC-NDL 2007 年版）」を公開した。Dublin Core に準拠する基本方針は変わらず、日本語の情報資源に対応するため、タイトルや作成者の読みや、国内における統制語彙や識別子等の標準体系に関する語彙を独自に拡張した。そして、NDL 内外のシステムとデータを連携する際、PORTA のメタデータとして実際に利用することで、国内の図書館や関連機関のデジタルアーカイブへの浸透を図った。

3.3. 国立国会図書館ダブリンコアメタデータ記述（DC-NDL 2010 年 6 月版）

Dublin Core は相互運用性を考慮し、語彙として基本 15 要素を定めたが、その記述方法についての制約は設けず、非常にシンプルかつ自由度の高いメタデータが記述できることから広く普及した。しかし、セマンティックウェブ環境でのコンピュータによる自動処理には、より厳密で精緻なメタデータが必要とされる等の理由から、Dublin Core において徐々に見直しが図られ、2008 年 1 月に DCMI Metadata Terms[9]が公表された。DCMI Metadata Terms では、RDF を参考にセマンティックウェブの実現に向け、記述要素の拡充や意味定義の見直し、使用範囲の明確化や概念関係の整理が行われた。

一方、NDL におけるメタデータを取り巻く状況もさらに進展した。2009 年 7 月に国立国会図書館法が改正され、日本国内の公的機関等が一般に公開しているインターネット情報については許諾を得ずに、収集・保存ができるようになったことを受け、WARP[10]におけるウェブサイトの収集範囲・頻度が拡充した。また、大規模デジタル化事業の成果であるデジタル化画像のメタデータ作成や、紙・電子媒体を問わず一括して情報資源

---

7 http://dublincore.org/documents/abstract-model/（参照 2013-01-11）
8 改訂案は 2007 年 4 月に提示され、6 月に確定した。一方、DC-NDL2007 年版は同年 5 月に公開したため、最終的なモデルの反映は次回改訂への課題とした。
9 http://dublincore.org/documents/dcmi-terms/（参照 2013-01-11）
10 正式名称は「インターネット資料収集保存事業（ウェブサイト別）」に変更。

及びメタデータを検索できる新サービス「国立国会図書館サーチ」[11]の開発版に対応するため、2009 年 9 月から大幅な改訂作業に着手した。

この際、DC-NDL の特徴の一つでもあるセマンティックウェブへの対応を強化した。DCMI Metadata Terms のセマンティックウェブ志向と歩調をあわせ、DC-NDL で独自に定義した語彙もコンピュータで解釈できるように NDL Metadata Terms の RDF スキーマを用意した。また、「2.3 相互運用性の向上」で述べた通り、語彙の意味の定義（セマンティクス）と語彙の使用方法（シンタックス）が一体となっていた従来の構成を改め、NDL で独自に定義した語彙集 NDL Metadata Terms と、Dublin Core 等の標準的な語彙及び NDL Metadata Terms 双方の記述方法を定めた Application Profile に分離した。こうした構成変更を反映し、記述要素以外も含むことから、従来の日本語名称「国立国会図書館ダブリンコアメタデータ記述要素」を「国立国会図書館ダブリンコアメタデータ記述」とした。

3.4. 国立国会図書館ダブリンコアメタデータ記述（DC-NDL 2011 年 12 月版）

DC-NDL 2010 年 6 月版の公開後、同年 10 月に DCMI Metadata Terms の小規模な変更[12]が行われた。小規模とはいえ、タイトルの表現方法の変更等少なからず DC-NDL にも影響を及ぼす内容であった。また、「国立国会図書館サーチ」においても、開発版の成果を踏まえ、メタデータの提供等に関するさらなるサービス充実化のための機能拡張への対応も必要となった。そのため、Dublin Core に準拠する基本的な方針や構成は変更せず、新たな語彙の追加や記述方法の修正等のみを実施し、DC-NDL 2011 年 12 月版として公開した。

表 1 は、Dublin Core の動向及び NDL の電子図書館サービスの進展からみた DC-NDL の変遷をまとめたものである。

11 http://iss.ndl.go.jp/（参照 2013-01-11）
12 http://dublincore.org/usage/decisions/2010/dcterms-changes/（参照 2013-01-11）

表 1　DC-NDL の変遷（2013 年 1 月現在）

年	DC-NDL と NDL の動向	Dublin Core の動向
1997		基本 15 要素の基本的合意
1998	「国立国会図書館電子図書館構想」策定	
2000	「電子図書館サービス実施基本計画」[13]「ネットワーク系電子出版物に関する指針」[14]策定	
2001	NDL メタデータ公開	
2002	WARP・Dnavi 等の電子図書館サービス公開 *ウェブサイト、データベースの組織化開始	
2003		国際規格 (ISO 15836)
2004	「国立国会図書館電子図書館中期計画 2004」[15]策定	
2005		DCMI 抽象モデル（推奨）公開
2006	WARP 本格事業化	DCMES　1.1 公開
2007	DC-NDL2007 年版公開 PORTA 公開 *多様な媒体・種類の情報資源に関するメタデータ標準化の必要性	DCMI 抽象モデル改訂 シンガポールフレームワーク提示
2008		DCMI Metadata Terms 公開 「アプリケーション・プロファイルのためのシンガポールフレームワーク」[16]公開
2009	著作権法改正（2010 年 1 月施行） *原資料保存のためのデジタル化が可能に 大規模デジタル化事業の実施（2 か年） 国立国会図書館法改正（2010 年 4 月施行） *国内公的機関等のウェブサイト・電子雑誌等を許諾なく収集・保存可能に	「ダブリンコア・アプリケーション・プロファイルのためのガイドライン」[17]公開
2010	DC-NDL2010 年 6 月版公開 「インターネット資料収集保存事業」拡充 「国立国会図書館サーチ（開発版）」公開	DCMI Metadata Terms 小規模改訂
2011	DC-NDL2011 年 12 月版公開 「国立国会図書館デジタル化資料」[18]公開 *デジタル化コンテンツのメタデータの標準化が課題	
2012	「国立国会図書館サーチ」正式公開 *API による DC-NDL 形式のメタデータ提供開始	

13 http://www.ndl.go.jp/jp/aboutus/elib_standardproject.html（参照 2013-01-11）
14 http://www.ndl.go.jp/jp/aboutus/elib_nw.html（参照 2013-01-11）
15 http://www.ndl.go.jp/jp/aboutus/elib_plan2004.html（参照 2013-01-11）
16 http://dublincore.org/documents/singapore-framework/（参照 2013-01-11）
17 http://dublincore.org/documents/profile-guidelines/（参照 2013-01-11）
18 http://dl.ndl.go.jp/（参照 2013-01-11）

4. 仕組み

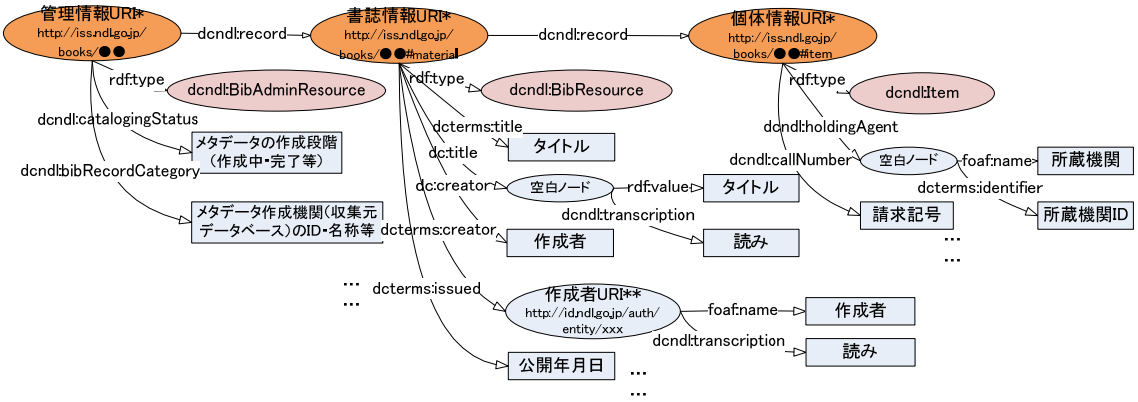
本章では、DC-NDL のスキーマ構造について説明する。

4.1. 概念モデル

DC-NDL の語彙は、3 つのクラス「管理情報（dcndl:BibAdminResource）」「書誌情報（dcndl:BibResource）」及び「個体情報（dcndl:Item）」に分かれる。「管理情報」はメタデータ自体に関する情報（メタデータの作成日や更新日等）、「書誌情報」は記述対象の情報資源に関する情報（タイトルや作成者等）、「個体情報」は記述対象を所蔵する各機関のローカル情報（請求記号等）に該当する。各クラス間は関連性を表わす語彙（dcndl:record）で関連づけられている。

RDF は、ものごとの関係を主語、述語及び目的語の 3 つの要素（トリプル）で記述し、トリプルを組み合わせることで、複雑な構造のデータも表現できる。また、インターネット上のリソースに限らず、概念、人物、事象等あらゆるものに一意の識別子である URI を付与することで、これらの関係を自由に記述していくことができる。

図 1 は RDF で表現した DC-NDL の概念モデル例である。数多のトリプルを繋げることで、各データ間の関係性が記述できる仕組みとなっている。



*国立国会図書館サーチ<http://iss.ndl.go.jp/>に拠る
**国立国会図書館典拠データ検索・提供サービス（Web NDL Authorities）<http://id.ndl.go.jp/auth/>に拠る

図 1　DC-NDL 概念モデル（イメージ）

4.2. 採用語彙

「2.3 相互運用性の向上」で述べた通り、既存の語彙をできる限り利用するアプリケーション・プロファイルの考え方に準じ、表 2 の語彙を採用している。

表 2　DC-NDL で採用する語彙

語彙	名前空間	接頭辞
Dublin Core Metadata Element Set, Version 1.1	http://purl.org/dc/elements/1.1/	dc
DCMI Metadata Terms	http://purl.org/dc/terms/	dcterms
Dublin Core Type Vocabulary	http://purl.org/dc/dcmitype/	dcmitype
FOAF Vocabulary	http://xmlns.com/foaf/0.1/	foaf
RDF Vocabulary	http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf
RDF Schema Vocabulary	http://www.w3.org/2000/01/rdf-schema#	rdfs
OWL Web Ontology Language	http://www.w3.org/2002/07/owl#	owl
NDL Metadata Terms	http://ndl.go.jp/dcndl/terms/	dcndl
NDL Type Vocabulary	http://ndl.go.jp/ndltype/	ndltype

4.3.　メタデータフォーマット

システムごとに内部で保持するデータ形式が異なっていても、用途が内部処理のためであれば、そのシステムに適した形式で実装していても支障はない。しかし、他のシステムとデータを交換・共有する場合、相互に変換可能な語彙やデータモデルが必要になる。連携先のシステムごとにカスタマイズすることも可能だが、多数のシステムと効率的にやりとりするためには標準的なデータ形式で入出力を行うことが望ましい。そのため、図書館や関連機関等がメタデータを交換・共有するための標準的なフォーマットとして、DC-NDL(RDF)及び DC-NDL(Simple)の 2 種類を定義している。

DC-NDL(RDF)は、構造化された RDF/XML 形式で入出力でき、リッチなメタデータを作成・利用できる。一方、DC-NDL(Simple)は、利用頻度の高い要素のみを XML 形式で入出力できるフォーマットである。前者と比較して情報量は限られているが、複雑な階層構造を持たず文字通りシンプルなフォーマットであるため、RDF に対応できないシステムでも実装することができる。

これらのフォーマットは、機関間のデータ交換だけでなく、利用者が自由にデータを繋げ、新たなサービスとして利活用することもできる。

5.　取組み

5.1.　現在の取組み—NDL での採用状況—

国内の関連機関の所有する多様な情報資源を統合的に検索できる「国立国会図書館サーチ」では、連携機関とのメタデータの交換用フォーマットとして DC-NDL(RDF)及び DC-NDL(Simple)を実装しており、各形式のメタデータを取得できる API（Application Program Interface）も提供している[19]。

[19] http://iss.ndl.go.jp/information/metadata/（参照 2013-01-11）

図 2 は、国立国会図書館サーチから出力した DC-NDL(RDF)形式のメタデータ例である。「4.1 概念モデル」で述べた 3 クラスから成り、構造化された形で表現されている。一方、DC-NDL(Simple)は、RDF を用いないフラットな XML で表現されているため、管理情報及び個体情報の記述は含まれていない[20]。

図 2　DC-NDL(RDF)形式のメタデータ例

上記以外に「国立国会図書館デジタル化資料」で提供する所蔵資料のデジタル化コンテンツ（画像、音源等）や WARP で収集・保存しているウェブサイトのメタデータスキーマとして使用しているほか、「国立国会図書館典拠データ検索・提供サービス（Web NDL Authorities）」[21]の典拠データを記述するメタデータにも一部 NDL Metadata Terms を採用している。また、「国立国会図書館東日本大震災アーカイブ構築プロジェ

[20] DC-NDL(Simple)の例は、下記参照。
http://iss.ndl.go.jp/api/oaipmh?verb=GetRecord&metadataPrefix=dcndl_simple&identifier=oai:iss.ndl.go.jp:R100000002-I000011254088-00

[21] http://id.ndl.go.jp/auth/ndla（参照 2013-01-11）

クト」[22]では、国内外の各機関等が収集・保存している東日本大震災の記録・教訓等の一元的な検索を可能とするため、文書、写真、音声及び動画等の多様な震災に関する情報資源のためのメタデータスキーマを策定しており、そのベースに DC-NDL を取り入れている。

5.2. 今後の取組み (1) —オンライン資料への対応—

2012 年 6 月、国立国会図書館法の一部が改正され、民間の出版するオンライン資料を NDL が収集し、保存することが可能となった[23]。オンライン資料とは、いわゆる電子書籍や電子雑誌に相当し、2013 年 7 月から無償かつ DRM（技術的制限手段）のないオンライン資料の収集が開始される。具体的には、私人がインターネット上で公開した電子書籍・電子雑誌のうち、特定のコード（ISBN、ISSN、DOI）が付与されたもの又は特定のフォーマット（PDF、PDF/A、EPUB、DAISY）で作成されたもののいずれかに該当するものであり、年鑑、要覧、機関誌、調査報告書、事業報告書、学術論文、紀要、技報、ニュースレター、小説、実用書、児童書等が挙げられる。

オンライン資料の制度的収集の開始により、これまで以上に、紙媒体と電子媒体、PDF と EPUB 等の同一著作[24]で異なる媒体・フォーマットの情報資源が増加すると見込まれる。これらの情報資源を一元的なフレームワーク[25]において表現できるようにするとともに、情報資源間の関連性をメタデータとして記述することが重要となる。NDL では、DC-NDL を用いて、異なる媒体やフォーマットの情報資源を同一著作のもとに集めるような仕組みをまだ実現できていない。そのため、利用者が求める情報資源を多様な媒体・フォーマットの中から効率的に識別・選択できるような同一著作の集中機能の導入を検討する等、後述する国際的な動向を見ながら今後の対応を考える必要がある。

また、オンライン資料では、資料の作成者・公開者自身が既にメタデータを付与している場合や、本文のテキスト情報をもち、全文検索が可能な場合も多い。そのため、情報資源の種別を問わずに一元的に検索・利用でき、かつオンライン資料の特性に応じた組織化が可能なメタデータを検討するにあたり、その作成単位や記述の粒度について留意する必要があろう。

こうした課題の解決に取組むにあたり、相互運用性の向上を実現するために設計された DC-NDL の基本的な枠組みや仕組みは有効であると期待される。

5.3. 今後の取組み (2) —国際的な動向への対応—

図書館コミュニティでは、メタデータに相当する書誌データの機械可読形式として

---

[22] http://www.ndl.go.jp/jp/311earthquake/disaster_archives/index.html（参照 2013-01-11）
[23] http://www.ndl.go.jp/jp/aboutus/online_data.html（参照 2013-01-11）
[24] 本稿における「著作」は、書誌記述の概念モデル FRBR における「著作（Work）」（知的・芸術的創造物の単位）に相当する。
[25] メタデータの記述、流通及び交換等のための容れ物。枠組み。

MARC（MAchine Readable Cataloging）が広く普及している。しかし、MARC は 1960 年代のデータ管理技術に基づくため、技術の発展や情報資源の記述方法の変化に対応し、より現在及び今後のウェブ環境に即したデータ作成・交換環境へ移行する必要性が徐々に認識され始めた。そこで、XML や RDF、URI 等のセマンティックウェブを実現するための技術を共通基盤とし、ウェブ上における図書館等が所有するデータの開放性・再利用性を高めるため、新たなフレームワークを構築する動きが起きている。海外では、米国議会図書館等が中心となり、MARC 替わる新しい書誌フレームワーク[26]の開発が進んでいる。そのため、こうした動向等に注意を払いつつ、日本の言語環境等に適したフレームワークを検討するにあたり、DC-NDL の適用可能性を考える必要もある。

5.4. 今後の取組み (3) —さらに繋がるデータへの対応—

ウェブ全体では、セマンティックウェブや Linked Data といった技術・概念はこの数年で浸透し、その利活用を促進するような様々な取組みが行われている。2011 年には W3C の図書館 Linked Data インキュベータグループが Linked Data を図書館データに適用することの利点やその方法等を提言する最終報告書を刊行した[27]。また、国内においても、博物館・美術館のデータや学術情報を Linked Data として公開する LODAC Project[28]等の取組みも見られるようになった。

しかし、国内の図書館等における実際の適用状況を見てみると、Linked Data を活用してデータが繋がりあっているというよりは、OAI-PMH 等のウェブ API を中心としたメタデータ連携が進展している段階である。今後、セマンティックウェブや Linked Data がさらに普及し、各機関が RDF や URI 等の標準に基づき、データの公開・共有を進めることができれば、データ間の関係性をさらに構築でき、一歩進んだデータの繋がりを実現できる。これにより、各機関が所有する情報資源のウェブ上における可視性や利用可能性がさらに向上することも期待される。

こうしたデータの繋がりを実現する「ハブ」の役割を果たせるメタデータ標準として、今後も DC-NDL の維持管理に努めていきたい。

---

[26] http://www.loc.gov/marc/transition/（参照 2013-01-11）
[27] http://www.w3.org/2005/Incubator/lld/XGR-lld-20111025/（参照 2013-01-11）
日本語訳 : http://www.ndl.go.jp/jp/aboutus/standards/translation/XGR-lld-20111025.html（参照 2013-01-11）
[28] http://lod.ac/（参照 2013-01-11）

参考文献

[1] “国立国会図書館ダブリンコアメタデータ記述（DC-NDL）”.
http://www.ndl.go.jp/jp/aboutus/standards/meta.html, (参照 2013-01-11).
[2] “国立国会図書館ダブリンコアメタデータ記述（DC-NDL）解説”.
http://www.ndl.go.jp/jp/aboutus/standards/meta/about_dcndl.html,
(参照 2013-01-11).
[3] “国立国会図書館ダブリンコアメタデータ記述（DC-NDL）実例集”.
http://www.ndl.go.jp/jp/aboutus/standards/meta/dcndl_examples.html,
(参照 2013-01-11).

PageRank と学術論文の評価: ノーベル賞の窓を探そう

藤田裕二（(株)ターンストーンリサーチ，日本大学）
yuji@turnstone.jp

1. はじめに

近年，情報通信技術の発展に伴って，利用可能なデータが急速に増大したことにより，データ処理の新たな手法が続々と考案されつつある．その多くは，確率論を応用したものである．これらの手法は古典的な意味での「アルゴリズム」とは違って，常に正しい(あるいは望ましい)結果をもたらすことは必ずしも保証されていないが，古典的なアルゴリズムを用いると現実的なコストでは解決不可能な課題について，ほぼ望ましい動作をする．

そのようなアルゴリズムのなかに，大手検索サービス Google search で利用されてきた PageRank がある．PageRank はウェブサイトのコンテンツの重要度を，リンク関係に基づいて統計的に評価するアルゴリズムであり，その的確な動作と頑健性は初期の Google 社の発展の有力な原動力となったと言われている．

2. PageRank アルゴリズム紹介

PageRank アルゴリズム自体の解説は馬場肇氏による下記のサイトが直観的かつ正確である．

http://homepage2.nifty.com/baba_hajime/wais/pagerank.html

馬場氏によると PageRank は、 「多くの良質なページからリンクされているページは、やはり良質なページである」 という再帰的な関係をもとに、全てのページの重要度を判定したものである。

このアイデアをウェブサイトにせよ他のコンテンツにせよ，実際に使おうとするならばいくつか問題がある．

最初の問題は，ある文書から引用される別の文書が，他の文書を引用し… という関係が出発点の文書まで巡っている場合は，このループ内部のコンテンツをどうやって評価すればいいのか? ということである．ある文書の重要度が変化すれば，その効果が巡り巡って自分の重要度を変化させてしまい，それが再び伝播して… などとのんびりやっていたのでは永遠に決着しない．重要度の定義が再帰的な構成となっているの

である．さらに言えば，ループ内部の文書のスコアが決まらないことには，これが引用している他の文書のスコアも決まらない．

じつはこの問題は物理学のなかで有限状態マルコフ過程の定常分布として既に解決されており，理論的な回答が存在する．

引用関係にもとづいて，文書 i から j に読者が移動する確率 p_ji を j 行 i 列に収容した行列 M の最大固有値に対応する固有ベクトルが，各文書に読者が滞在する最終的な確率を与える．すなわちベクトルの値が各文書の重要さである．

しかし，その回答をそのまま文書間引用ネットワークに適用するにはいくつか障害があり，これをシンプルかつ実用的な手法で解決した事が，Page，Larry の，ネットワーク科学の理論における主要な貢献の一つである．

3. 学術論文評価への応用と画期的業績の特定

PageRank の発想は，他のテーマにも自然に応用できる．web contents におけるリンク関係は，そもそもは学術論文における引用関係にその発想の源泉をだどることができるので，たとえば重要な学術論文のなかで引用された業績は重要である，としてこのアルゴリズムを応用する事は自然である． 現状，被引用数あるいはそれに基づくインパクトファクタなどで研究業績を評価する事が主流であるが，web 検索の歴史を学術論文の評価がなぞるとすれば，PageRank はより頑健で信頼できる評価であると期待されるので，この発想に基づく先行研究が複数存在する．

これらの研究によると，被引用数に比較して高い PageRank をもつ業績は，数多くの重要な論文の生みの親という役割をもっていることを示す指標であり，

そのいくつかは実際にノーベル賞の受賞などにつながり，画期的なものとして評価されている．

4. 実装上の問題

多数の node 間の遷移確率を保持する巨大な行列を，そのままメモリに格納することは不可能であり，非効率的でもある．ただし行列の大部分の値は 0 であるから，そのような疎行列を圧縮して処理するためのアルゴリズムが複数存在し，いくつかはライブラリとして利用可能である．

PageRank アルゴリズムはいくつかの変数をとって，その振る舞いを変化させるが，特にこれを学術論文の評価に用いる場合は，遷移確率の減衰率に加えて発表年度の補正として適用する陳腐化率とでもいうべきパラメタが新たに加わる．これら二つのパラメタの最適値を求める作業は，PageRank 自体の計算のコストが小さくないだけに，

相当の労力が必要である．また，この陳腐化率はテーマや分野によって異なるので，実用上の見地から適切な値を算出する方法の考案が待たれるところである．

5. スパム問題とアルゴリズムの限界

PageRank アルゴリズムはかなり頑健であり，データの間に存在する意味内容から生じる構造をたくみに捕捉することができることがこれまでの経験で知られているが，悪用しようとすれば弱点が無いわけではない．中古車市場の研究にもあるように，安物を高く売りたいのは普遍的な欲求である．たとえば通信販売のサイトを持っていたとして，手間暇をかけずに自分のサイトのスコアを向上させることができれば，低いコストで高い宣伝効果を得ることができる．そのために必要なのは，他サイトからの多数の(無意味な)リンクである．そのようなサイトは，迷惑メール "スパム" の類似物として "ウェブスパム" と呼ばれており，大手検索サービスでは様々な対策を迫られている．同様の問題は学術誌においても存在し，掲載に際して系列雑誌の，しかも必ずしも本質的でない論文を引用することを求められる事例が知られている．これは PageRank アルゴリズムへの攻撃を意図したものではないが，やはり PageRank の集計結果に悪影響を及ぼす．

また，文献 3 がその最後で指摘しているとおり，これらの指標は，人気のあるものがさらに人気を集める，という側面がある事に常に注意を払って利用する必要がある．結局のところ，コンテンツの価値はそれが担う意味内容に存在し，たった一つの数値が学術的内容の精査の完全な代替となることは，到底ありえないからだ．

参考文献

1. Page, L., Brin, S., "The PageRank Citation Ranking: Bringing Order to the Web"
2. Maslov, S., Redner, S. "Promise and Pitfalls of Extending Google's PageRank Algorithm to Citation Networks"
3. Chen, P, Xie, H., Maslov, S., Redner, S., "Finding Scientific Gems with Google"
4. 馬場肇 "Google の秘密 - PageRank 徹底解説"
   (http://homepage2.nifty.com/baba_hajime/wais/pagerank.html)

すべてをコンピュータの中に（繋がってしまったデータとその未来）

発行日: 2013年2月16日
発行者: 全国共同利用・共同研究拠点「人文学諸領域の複合的共同研究国際拠点」
住所: 〒606-8501 京都大学人文科学研究所
印刷:


Institute for Research in Humanities
KYOTO UNIVERSITY