情報処理学会論文誌 Vol.61 No.2 171–178 (Feb. 2020)

内容アドレッシングを用いた多粒度漢字構造情報表現の試み

守岡 知彦[1,a)]

受付日 2019年5月16日, 採録日 2019年11月7日

概要：複数の包摂粒度を許容する漢字構造記述を内容アドレッシングを用いた分散データモデルの 1 つである IPLD に基づいて実現する試みについて述べる．IPLD は Merkle DAG というデータ構造を用いている．これは暗号学的ハッシュを用いてラベル付けされた不変オブジェクトをノードとする有向非巡回グラフであり，オブジェクト間の関係は片方向しか表現できない．この制約の下で編集可能な文字知識を表現するために，不変性の高い基礎的オブジェクトへのリンクを含む複合的なオブジェクトによって可変性のある文字オブジェクトを表現するための形式を提案する．また，IPLD グラフの逆リンクや場所アドレッシングの実現手法について検討するとともに，既存のグラフストレージでの試験的実装についても述べる．

キーワード：漢字，包摂規準，IDS，内容アドレッシング，IPLD，IPFS

An Attempt of Multiple Granularity Hanzi Structure Representation Using Content Addressing

Tomohiko Morioka[1,a)]

Received: May 16, 2019, Accepted: November 7, 2019

Abstract: This paper describes an attempt to realize content addressed representation of Hanzi structure description to support multiple unification granularity using IPLD which is one of distributed data models. IPLD uses a data structure called Merkle DAG. This is a directed acyclic graph in which every node is labeled with the cryptographic hash of its content. As its name implies, relationships between objects can be expressed only in one direction. In order to realize editable character knowledge under this restriction, we propose a format to represent a variable character object by a complex object including links to basic objects that are highly invariant. We also examine a method to realize reverse links and location addressing of the IPLD graph, and describe an experimental implementation using an existing graph storage.

Keywords: Chinese characters, unification rules, IDS, content addressing, IPLD, IPFS

1. はじめに

CHISE [1] の多粒度漢字構造情報を内容アドレッシングに基づくデータモデルである IPLD を用いて表現する試みについて述べる．

漢字は文字数が多く，また，広い地域において長期間使われてきた文字体系であることから異体字が多く，また，字義や音義が不明であったり解釈が分かれたりする文字も少なくない．このことから，歴史的な文字資料をコンピュータ上で扱う際に文字コード列からなる電子テキストだけでは不十分であることが少なくなく，画像を併用することが少なくない．しかしながら，文字画像は検索や電子テキストとしての再利用性などの点で問題があり，グリフ（文字や記号などの具体的な形状）を国際的な登記簿[*1]に登録する制度を定めた ISO/IEC 10036（JIS X 4165），IVS [2] などの異体字セレクターを用いて漢字の見掛け上の字体や字形をなるべく忠実に表現しようとするアプローチや，文字画像にアノテーションを施すことで検索可能性を高めようとするアプローチ，あるいは，電子テキストもしくは文字画像のいずれかもしくはその両方に XML でマークアップ

[1] 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
Center for Informatics in East Asian Studies, Institute for Research in Humanities, Kyoto University, Kyoto 606–8265, Japan
[a)] tomo@zinbun.kyoto-u.ac.jp

[*1] http://10036ra.org/

を行うアプローチ [3] などが考えられている．このうち，少なくとも ISO/IEC 10036 と IVS はグリフの重複符号化の問題があるが，そもそもグリフの同一性の判断をその形状だけで行うことには無理があるといえ，漢字に関する知識をどのように記述し，それをどのように共有するかが，文字やテキストの安定的な記述という観点において，より本質的な問題といえよう．

CHISE では『Chaon モデル』と呼ぶ方法によって文字を表現するようになっている．これは表現したい文字に関する知識（文字の性質の集合）の機械可読な表現によって文字を表現し操作する方法である．Chaon モデルでは，文字を説明するための要素を『文字素性』と呼ぶ．文字素性としては，部首，画数，部品の組合せ方に関する情報（漢字構造情報），発音，意味，用例，その他文字処理で必要となる各種情報などが考えられる．CHISE では文字素性の集合による文字定義を行い，それに基づいた文字処理の仕組みや文字情報サービスを提供してきた．

CHISE では文字符号などを表現するために ID 素性と呼ばれるオブジェクトの同一性を示すための特殊な素性が用意されているが，現代字への対応が不明な先秦漢字をとりあえず隷定したものや後述する中間部品のように，別の文字オブジェクトとの関係によって表現される対象を扱う場合，その同一性や性質について検討することなくとりあえずデータ化できる方が便利であるため，ID 素性なしに文字オブジェクトを定義することも認めている．しかし，現在の実装では ID 素性がない場合，まったく同じ文字定義を用いたとしても別の文字オブジェクトになるという問題があり，実用上，最低でも 1 つの ID 素性を文字定義に含める必要がある．また，Web の世界で CHISE の文字情報をやりとりするためには文字オブジェクトやその構成要素に IRI を割り振ることが必要といえ，この観点ではすべてのオブジェクトに ID 素性を付与すべきだといえる．しかしながら，恣意的に付与せざるを得ない ID 素性を要することは分散環境において自由に文字を表現し交換するという目標に対して重大な制約を科すものといえる．

こうした制約が実際に問題となっている対象の 1 つが漢字構造情報の処理である．IDS（Ideographic Description Sequence）[4] を用いた漢字構造記述では IDS を入れ子状に記述することができるが，この文字になっていない中間部品はそのままでは ID 素性を持たないため問題を引き起こすわけである．原理的には，もし文字オブジェクトでも ID 素性の値として文字列やシンボルを用いることができれば GlyphWiki [5] が行っているように IDS に基づく ID を用いることができるといえるが，CHISE では部品として UCS の抽象文字以外のものも利用可能であり，また，多粒度包摂モデルに基づいた部品の包摂粒度を記述する必要があるため，IDS を ID として用いるだけでは解決できないといえる．

しかしながら，もし文字定義そのものを ID のように使えれば ID 素性なしに同じ（構造等価）な文字定義はつねに同じ文字オブジェクトを指すことになると考えられる．たとえば，文字定義に何らかの正規化を施しそのハッシュ値を用いることが考えられる．ここでは P2P ベースの分散型ファイルシステムの 1 つである IPFS とそのデータモデルである IPLD を用いて実際に漢字構造情報を表現することを試みる．

2. IPFS

IPFS（InterPlanetary File System）[6], [7] は Protocol Labs 社が中心となって開発しているオープンソースの P2P 型分散ファイルシステムである．IPFS の開発は Git などで版管理された科学的データを高速に転送するシステムの構築を目的として Juan Benet によって始められたが，後に，分散化され永続化された Web として構想されるようになった．

IPFS では Git と同様にオブジェクトをそのハッシュ値によって参照するという『内容アドレッシング』（content-addressing）を用いている．通常の Web では URL（IRI）という資源の場所に基づくアドレッシング（location-addressing；『場所アドレッシング』）になっており，識別子の指す先が変化する可能性があり，また，オブジェクトに変化がなくてもその識別子が変化してしまうと参照できないという性質を持っている．このことは学術資源の長期保存においては良い性質とはいえず，このため DOI のような永続的識別子の利用が注目されている．この観点から IPFS を見ると，IPFS では DOI のような永続的識別子を任意のオブジェクトに対して機械的に生成する仕組みと見ることができる．IPFS ではオブジェクトのハッシュ値に基づいて内容 ID（CID；Content IDentifier）を生成する仕組みがプロトコルによって定義されており，いつどこで誰が行っても対象とするオブジェクトが同一のバイト列を持っていれば同じ CID が生成されるため，DOI や DNS のように機関を認証して権限を委譲する必要はなく，結果的に，非常に小さな粒度のデータから非常に大きなデータセットの集合といったスケーラビリティで永続的な識別子を生成することができる．

IPFS は**表 1** に示す 4 層で構成されており，下位の 3 層および IPFS が提供するいくつかのアプリケーションに関して，JavaScript と Go 言語による実装が提供されている．

表 1 IPFS のプロトコル階層モデル
Table 1 Model of the IPFS stack.

アプリケーション（IPFS，その他）	
IPNS	名前解決
IPLD	Merkle DAG によるデータモデル
libp2p	ネットワークやルーチング，データ転送

2.1 IPLD

IPLD（InterPlanetary Linked Data）[8], [9] は IPFS のデータモデルを提供する層で，Merkle DAG と呼ばれる暗号学的ハッシュに基づく有向非巡回グラフ（Directed Acyclic Graph; DAG）と，Merkle paths と呼ばれる Merkle DAG を名前付きリンクをたどってトラバースしたものを UNIX 風の階層的ファイル名に対応させる仕組みと，IPLD の正規化されたデータ形式（IPLD Canonical Format）などの仕様により，JSON や XML などの外部形式を IPLD の世界に格納したり IPLD の世界のオブジェクトを指定した外部形式で出力することができる．

IPLD は IPFS のデータモデルであるので，IPFS 上のさまざまなアプリケーションはすべて IPLD の DAG として記述されたデータを持っている．また，CID は複数の版，複数のコーデック（multicodec），複数のハッシュ関数（multihash），複数のバイナリテキスト化方式（multibase）をサポートするが，ここでは IPFS の Go 実装（go-ipfs）の ipfs コマンドの ipfs dag {get|put} や IPFS HTTP API の/api/v0/dag/{get|put} で読み書きできる cidv1-cbor-sha2-256 形式，すなわち，版は CIDv1 でコーデックとして CBOR，ハッシュ関数として SHA2-256 を用いるものとする．また，バイナリのテキスト化方式として Base32 を用いるが，本稿では誌面の都合から以前の版で用いられていた Base58BTC（Bitcoin の Base58）で記述する[*2]．

IPLD のオブジェクト（DAG のノードとなるもの）は JSON のオブジェクトと同様な key と value の対の集合である．たとえば，

```
{ "name": "Paul Marie Ghislain Otlet" }
```

は

```
zdpuAwbrw9r5jy1gUoB61EoNLGRossy171TGKyB2AbAvmArmo
```

という CID を持つ IPLD のオブジェクトの JSON 表現である．

IPLD に JSON のオブジェクトを格納する場合，IPFS のようにバイト列そのものが格納されるわけではなく，正規化された CBOR（Concise Binary Object Representation）[10] 形式に変換されて格納される．この際，空白・改行の有無・個数や key-value 対の順番は捨象されるため，これらの差異にかかわらず同じオブジェクトとして扱われる．また，現状，go-ipfs の dag サブコマンドでは JSON 以外の形式がサポートされていないが，原理的には，XML や YAML，RDF といった形式であっても，それが同一の key-value 対の集合を表現しているならば同一のオブジェクトとして扱われることになっている．

IPLD では他のオブジェクトへのリンクは key が "/" で value が CID である link-object と呼ばれる特殊なオブジェクトで表現される．たとえば，

```
{ "/":
    "zdpuAwbrw9r5jy1gUoB61EoNLGRossy171TGKyB2AbAvmArmo"
}
```

は link-object の JSON 表現である．また，

```
{ "name":  "Henri La Fontaine",
  "collaborator":
    { "/":
      "zdpuAwbrw9r5jy1gUoB61EoNLGRossy171TGKyB2AbAvmArmo"
    }
}
```

を JSON 表現として持つ IPLD オブジェクト

```
zdpuAoDrX6sbMNE5N3i7YTrVuEidCL41EoSryCr5y8UhrR4PC
```

のようにある key の value に他のオブジェクトへのリンクやリンクの配列を入れることができ，これにより DAG を構成することができる．

Merkle-paths の規定により，リンクを持つオブジェクトはリンクを値として持つ key を用いて，CID/key のような派生的 CID によってリンク先を参照することができる．たとえば，

```
% ipfs dag get \
zdpuAoDrX6sbMNE5N3i7YTrVuEidCL41EoSryCr5y8UhrR4PC/collaborator
```

は {"name":"Paul Marie Ghislain Otlet"} という出力を返す．もし，リンク先にも同様に名前付きのリンクがある場合，CID/a/b/c/...のように階層的にリンクをたどることができる．

3. IPLD での表現上の注意点

IPLD では構造等価な JSON データは同じ CID になる．CHISE 的にいえば，同じ素性対の集合で記述された（すなわち，同じ文字定義を持つ）2 つの文字オブジェクトは必ず同一（一意に定まる）ということである．いい替えれば，ある文字オブジェクトに文字素性を追加したり素性値を書き換えたら CID が変化し，別の文字オブジェクトになってしまうということである．こうしたことから，CHISE の文字データを IPLD 化する場合，変化しにくい部分と変化しやすい部分に分けて扱う方がよいと考えられる[*3]．文字オブジェクトに含まれる情報の場合，文字符号や字書などでの文字番号に関するもの，すなわち，従来 ID 素性で表現してきたような情報は不変性が高いといえる．しかしながら，複数の ID 素性対を同一視して束ねる場合，どれとどれを束ねるかには恣意性があったり新たな ID 素性が追加されたりする（個々の ID 素性対には不変性があったとしても）ことに注意する必要がある．

また，UCS の符号位置や大漢和辞典の文字番号に対応する文字の情報を知りたい場合のように場所アドレッシング

[*2] CID は自己記述的な形式（self-describing format）であり，版やコーデック，ハッシュ関数，テキスト化方式のそれぞれにどれを用いたかの情報が埋め込まれている．各形式は `ipfs cid base32` *CID* で cidv1-cbor-sha2-256 の Base32 に変換でき，下記の例の Base32 でテキスト化した CID を用いて `ipfs dag get bafyre`（以下略）すると下記の JSON を得ることができる．

[*3] このことはおそらく構造データ一般でも成り立つだろう．

が必要な場合も考えられるが，場所アドレッシングで用いられる情報資源の場所を示すための情報（識別子や名前的な情報）はその性格から不変性が高い情報であるといえ，前述したように不変性の高い IPLD オブジェクトとして，可変性のあるオブジェクトの中にこれらのリンクを含むことにより場所アドレッシングを実現するための手がかりとすることができる．

一方，IPLD は DAG に基づくデータモデルであるため，それ自体では逆リンクが表現できないことに注意する必要がある．

4. 多粒度漢字構造モデル

多くの漢字は偏と旁などの部品の組合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方のことを『漢字構造』と呼ぶことにする [11]．漢字構造の表現法としていくつかの形式が提案され利用されてきたが [12]，現在では，ISO/IEC 10646 [4] の一部として標準化された Ideographic Description Sequence（IDS）という形式が普及している．漢字構造は部品の組合せ方を示す演算子と部品からなる構文木で表現できる．IDS は演算子として IDC（Ideographic Description Characters），部品として UCS に収録された漢字，部品用文字，および IDS を用いたものであるが，部品としてそれ以外のものを用いることも原理的には可能である．

ここで，部品として複数の異なる包摂粒度を持つものを用いれば，複数の部品の組合せで構成される漢字の各部品の包摂範囲を示すことで，その漢字の包摂範囲を示すことができるといえる．これを『多粒度漢字構造モデル』と呼ぶ [1]．多粒度漢字構造モデルにおいて，どのような包摂粒度階層を用いるかは随意であるといえるが，現在，CHISE 文字オントロジーでは，主な階層として，超抽象文字（字種）–抽象文字–抽象字体–抽象字形–字形という 5 階層の粒度を用いている．また，補助的な階層として，抽象文字粒度と抽象字体粒度の間に統合字体粒度，抽象字体粒度と抽象字形粒度の間に詳細字体粒度を置くことを許している．

5. 符号化文字の表現

漢字構造をデータ化する場合，そのノードとなる部品をどのように表現するかが問題となる．すなわち，IPLD の世界において CHISE の文字オブジェクトをどのように表現するかを考える必要があるといえる．

狭義の IDS の場合，部品は UCS に収録されたものに限定されるため，その部品の UCS の符号位置を使って表現すればよい．たとえば，符号化方式（CES）として UTF-8 を用いることにすればそのバイト列は一意に決まる．しかしながら，「漢」を示す U+6F22 は「漢」や「漢」も包摂するため，字体を特定したい場合別の仕組みが必要である．

たとえば，「漢」という字体を示したい場合，U+6F22 U+E0101 という IVS を用いてこれが Adobe-Japan1 の CID+13332 のグリフであることを示したり U+6F22 E0103 という IVS を用いて汎用電子 [13] の JC8705 [*4] のグリフもしくは文字情報基盤 [14] の MJ030268 のグリフであることを示すという方法がある．ただ，この方法の場合，CID+13332 と JC8705 が同じ字体であるということはこれだけでは分からない．また，汎用電子と文字情報基盤は IVS を共有しているため，同じ IVS にもかかわらずグリフが大きく異なっている場合にそのどちらを指しているかは IVS だけでは決定できない．また，IVS の場合，ある包摂された統合漢字をどのように区別するか，すなわち，どういう包摂粒度を用いるかは IVD の定義に基づき，利用者が自由に設定することはできない．また，IVD は IVS と例示されるグリフの対応関係を定義するだけであり，グリフの分離・統合の基準は明示されない．よって，現実的には字体を指示するための仕組みというよりは特定用途におけるグリフセット（大規模外字集合）との相互変換を実現するための仕組みと考えた方がよい．

人文系資料などのための仕組みとしては，どういうものを区別しどういうものを統合するかについて，字体の差異を捨象した字種レベルの記述から細かな字形差に着目した記述までを許容するようなスケーラビリティがあることが望ましく，また，どういった文字でも表現できるような自由度があることが望ましい．しかしながら，インターネットでの情報共有が基本となる今日，いたずらに外字を使用することは望ましくなく，なるべく外字の使用は控え，将来的に UCS に収録された際に容易に置き換え可能な仕組みを用いることが望ましい．また，外字であってもその文字が捜し出せるような仕組みが必要であるといえる．

UCS や JIS X 0213 といった文字符号の標準，Adobe-Japan1 や文字情報基盤のような標準的なグリフセットは文字に関する情報共有のための基盤として有用であるが，こうした符号化文字やグリフの集合におけるあるコードポイントで文字を表現するとき，それが実際にはどういう文字概念を指示しているかを示すことができた方がよいといえる．また，これはさまざまな詳細さにより記述が可能であると同時によく使われるものに効率的にアクセスできることが望ましいといえる．

こうした観点から，CHISE では

(1) 包摂粒度を示すための指示子
(2) ドメイン
(3) 等価性の定義

という 3 種類の仕組みで符号化文字を表現している．IPLD 化においてもこの仕組みを踏襲することにする．

[*4] JIS X 0213 の第 1 面 87 区 05 点の例示字形に対応するグリフであることを示している．

5.1 符号位置オブジェクト

符号位置オブジェクトは

{ 文字符号名：符号位置 }

のような文字符号名と符号位置の対で表現する．ただし，符号位置は整数で表現する[*5]．なお，ISO/IEC 2022 の図形文字集合の場合，94 × 94 文字集合に関しては GL 表現を用い，94 文字集合および 96 文字集合に関しては ISO 646 系に関しては GL 表現，それ以外（ISO 646 系と組み合わせることを想定したもの）に関しては GR 表現を用いるものとする．たとえば，U+5167（内）は

{ "ucs"：20839 }

という JSON で表現でき，その CID は

`zdpuAxm2netKNpcrd29ToLk9hsvrvEXpAgLDEAGsW6AU4zaD1`

となる．

5.2 包摂情報付き符号位置オブジェクト

包摂情報付き符号位置オブジェクトは符号位置オブジェクトと包摂粒度を示す述語を用いて，

```
{
  包摂粒度を示す述語：
    { "code-point"：{ "/"：符号位置オブジェクトの CID } }
}
```

のように表現する．ただし，包摂粒度を示す述語は**表 2** の「述語名」欄に示すものを用いる．また，JSON 形式では CID は文字列で表現するものとする．

たとえば，JIS X 0208 の 0x4662（38 区 66 点）の抽象文字（規格上の包摂範囲）に対応する包摂情報付き符号位置オブジェクトは符号位置オブジェクト

{ "jis-x0208"：18018 }

の CID

`zdpuAovC2r7BJj5hZVuMKmWZh5U8xnHtgArgZHoRfXqW2vUnz`

を用いて，

```
{
  "abstract-character-of"：
    { "code-point"：
      { "/"：
"zdpuAovC2r7BJj5hZVuMKmWZh5U8xnHtgArgZHoRfXqW2vUnz"
      }
    }
}
```

という JSON で表現でき，その CID は

`zdpuAtw9b6Qbmna3DkzvkXYuJWQ6LqSuQTCL9RiySFd4Ajorc`

となる．

表 2 包摂粒度の表現
Table 2 Representation of unification granularity.

包摂粒度名	述語名	接頭辞
超抽象文字	super-abstract-character-of	`==>`
抽象文字	abstract-character-of	`=>`
統合字体	unified-glyph-of	`=+>`
抽象字体	abstract-glyph-of	`=`
詳細字体	detailed-glyph-of	`=>>`
抽象字形	abstract-glyph-form-of	`==`
字形	glyph-image-of	`===`

[*5] 10 進数で記述する．

5.3 ドメイン情報の付与

UCS 統合漢字では各符号位置に対してそのソースとなった中国，香港，マカオ，台湾，日本，韓国，北朝鮮，ベトナム，そのほかの例示字形が多欄表示されている．よって，UCS 統合漢字の例示字形を示すには，符号位置（と字形という包摂粒度）だけではなく，どのソースのものであるかを示す必要がある．また，JIS X 0208 では版によって例示字形が異なるものがある．この場合，どの版のものかを示さないと例示字形を決定できない．あるいは，複数の異なる包摂ポリシが存在する場合，ある抽象文字の包摂範囲はそのポリシを明記しないと決定できないといえる．

CHISE ではこの種の情報を『ドメイン』として扱っているが，IPLD ではドメインおよび包摂情報付き符号位置オブジェクトを

```
{
  包摂粒度を示す述語：
    { "code-point"：{ "/"：符号位置オブジェクトの CID } },
    { "context"    ：{ "/"：ドメインオブジェクトの CID } }
}
```

のように表現する．ここで，ドメインオブジェクトは

{ "CHISE-domain"：ドメイン名 }

のように表現する．

5.4 符号化文字オブジェクトの IPLD での表現

符号化文字オブジェクトは同じ文字を指し示す包摂情報付き符号位置オブジェクトの CID の集合 $\{C_1, C_2, C_3, \dots\}$ を用いて

{ "unify"：{ f_1：{ "/"：C_1 }, f_2：{ "/"：C_2 },
f_3：{ "/"：C_3 }, ... } }

のように表現する．

ただし，f_n は文字符号名に包摂粒度を示す接頭辞（表 2 の「接頭辞」欄にこの一覧を示す）を付けたものである．たとえば，JIS X 0208 の抽象字体の場合，`=jis-x0208` となる．また，ドメイン情報が付与されている場合は，この接頭辞の後に文字符号名をつなげたものの後に `@` を置き，その後にドメイン名をつなげる．たとえば，JIS X 0208 の 1990 年版の場合，`=jis-x0208@1990` となる．なお，JSON では文字列で表現するものとする（たとえば，`=jis-x0208` は `"=jis-x0208"` という文字列で表現される）．

6. 漢字構造情報の IPLD での表現

CHISE 文字オントロジーでは漢字構造情報は IDS 形式

をパースした結果の構文木を S 式にしたものとして表現しており，その文字素性として ideographic-structure を用いている．CHISE の漢字構造情報を IPLD 化する場合，その ideographic-structure 素性をそのまま IPLD の JSON に直訳することは可能であるが，RDF との親和性を考慮し，CHISE の RDF 化 [15] で行ったのと同様な『IDC の述語化』や IDS 用コンテナを使ったモデル [16] を用いることにした．

文献 [15] と異なるのは，IDC を `"operator"` というメンバに格納するということと，包摂粒度を示す述語の場合と同様，IDC に対応する述語も接頭辞はとるということである．**表 3** に IDC に対応する述語の一覧を示す．なお，`SLR`（Surround from Lower Right）と `SRT`（Surround from RighT）は CHISE が拡張した IDC で，それぞれ，引数としてとる部品に対する右下からの囲みと右からの囲みを示す．また，2 つの IDC を「・」でつないだものは 2 つの IDC の両方のパターンを包摂した抽象 IDC を示す[*6]．

たとえば，「字」の漢字構造は

```
{
  ”operator” : ”\u2FF1”,
  ”above” :
    { ”/” :
      "zdpuArp21drHhyzxKZDXSsJ4G1QvXNsGgpUK6evUq7aqatNJv"
    },
  ”below” :
    { ”/” :
      "zdpuArnniMgYWGyhxnQNhwzDi4FR4148CT8prAoyvnkwKwwzS"
    }
```

表 3 IDC 述語一覧
Table 3 IDC predicates.

IDC	述語 1	述語 2	述語 3
⿰	left	right	————
⿱	above	below	————
⿲	left	middle	right
⿳	above	middle	below
⿴	surround	filling	————
⿵	surround	filling	————
⿶	surround	filling	————
⿷	surround	filling	————
⿸	surround	filling	————
⿹	surround	filling	————
⿺	surround	filling	————
⿻	underlying	overlaying	————
SLR（⿺）	surround	filling	————
SRT（⿹）	surround	filling	————
⿰・SLR	left	right	————
⿰・⿺	left	right	————
⿱・⿹	above	below	————
⿱・⿵	above	below	————

[*6] ⿰・⿺は IWDS-1 [17] の 307 番，⿱・⿹は同 305 番の包摂ルールに対応する．

}
という JSON で表現でき，その CID は
`zdpuB2d2yHpE3EdgBjgyF371mZktmknqXmyqJYMNMYBMmjXgL`
となる．

また，U+5B57 の抽象文字を示す包摂情報付き符号位置オブジェクトの漢字構造情報は

```
{
  ”operator” : ”\u2FF1”,
  ”above” :
    { ”/” :
      "zdpuB2qk4Bna1D3QE3Mb8WMf7yRdZ3dcSNkxN2Te9QJBZzWCV"
    },
  ”below” :
    { ”/” :
      "zdpuAxHGREWVdV6Skj8DnHP6z5aH85ajLAufqAhgYLukT1f5N"
    }
}
```

という JSON で表現でき，その CID は
`zdpuAmm22JYYkdPxBsdzBku2zCBWkZrkPZ2hLTbxQFPNzaTdX`
となる．

7. 可変オブジェクトの参照と名前解決

3 章で述べたように，不変性が高い情報（それはしばしば場所アドレッシングにおける識別子の役割を果たしている）を起点にそれらをリンクとして含む形で可変性のあるオブジェクトを記述する場合，場所アドレッシングや編集可能なオブジェクトを実現するという問題は不変性が高いオブジェクトにリンクされたオブジェクトをどのようにして探すかという逆リンクの探索問題に帰着するといえる．

もし，IPLD のグラフ全体を高速に探索可能であるならば，そうした検索サービスを利用して指定したオブジェクトを起点に指定したグラフパターンでつながったオブジェクトを検索すればよいわけであるが，この種のサービスは Web の検索サービスと同様にいくつかの起点からロボットを放ち，IPLD のグラフをたどりインデックスを作ることによって実現されており，タイムラグもあり完全ではない．結局，場所アドレッシングしたり逆リンクをたどったりしたいあるまとまった単位のデータセット（たとえば，UCS 統合漢字や CHISE 文字オントロジー，あるいは，字書や研究対象となった資料など）に収録されるオブジェクト間のリンクのすべてに対し，その逆リンクの情報（リンク先のオブジェクトとそのリンクに付けられた key を指定するとリンク元のオブジェクトが得られるインデックス）を設ける必要があるといえる．

このインデックスを IPLD オブジェクトで実現した場合，この IPLD オブジェクトはインデックスに対応するデータセット全体を表現したものと考えることができる．そして，データセットのどこかが書き換えられるとインデックスの IPLD オブジェクトの CID も変わるので，変更後の

新しい CID をなんらかの方法で受けわたせばよいといえる．このための手段としては，IPFS の PubSub 機能や後述する IPNS の利用が考えられる．

IPFS には IPNS という名前解決のための仕組みが存在する．これは公開鍵暗号技術を用いて IPFS ネットワークのノードに Peer ID というものを割り当て，電子署名技術を用いて Peer ID に IPFS の CID を対応付けるものである．これを用いれば，IPNS からの相対パスによる場所アドレッシングを実現することができるので，データセット名に対応するいずれかの場所にインデックスを置くことができる．このインデックスは前述のように IPLD オブジェクト（の CID）でもよいしより効率的なキーバリューストアやデータベースのファイルでもよい[*7]．ただし，IPNS という仕組みはノードに依存したものといえ，その上に置かれたインデックス（ファイル）はその全体を複数のノードで共有できるとはいえ，それ自体は分権化（decentralize）されていない．

きちんと分権化された仕組みに基づいてインデックスを実現することが重要であるといえるが，当面のアドホックな解決法として既存の Web 技術を併用することで場所アドレッシングの実現や逆リンクや付加情報の管理を行うことを試みた[*8]．これは具体的には IPLD のオブジェクトに対応する Concord [18] オブジェクトを作成し，EgT [19]（CHISE-wiki）を用いて参照するという方法である [20].

7.1 符号位置オブジェクト

符号位置オブジェクトに対しては `code-point` ジャンルの Concord オブジェクトを生成する．

Concord における符号位置オブジェクトはその ID として IPLD の CID を用い，ID 素性 `=ipld` にも CID を格納する．

また，IPLD のオブジェクトのメンバはその名前の前に `=` を付けた ID 素性を用いて表現する．ただし，この `=` は Concord において ID 素性にするために付けた接頭辞であり，包摂粒度を示すものではない．

7.2 包摂情報付き符号位置オブジェクト

包摂情報付き符号位置オブジェクトに対しては `coded-character` ジャンルの Concord オブジェクトを生成する．

Concord における包摂情報付き符号位置オブジェクトはその ID として IPLD の CID を用い，ID 素性 `=ipld` にも CID を格納する．

包摂粒度を示す述語に対応する素性は IPLD のメンバ名の接尾辞 `-of` をとり，*-of* を示す関係素性の接頭辞 `<-` を付けたものを用いる．これにより対応する符号位置オブジェクトに逆関係素性が自動的に付与される．

また，ドメイン付きの情報の場合，ドメイン付きの階層的素性名を用い，空白ノードに相当するオブジェクトは作らない．

また，利便性のために，対応する CHISE の文字オブジェクトを `character` 素性に格納している．

7.3 符号化文字オブジェクト

符号化文字オブジェクトに対しても，包摂情報付き符号位置オブジェクトと同様に，`coded-character` ジャンルの Concord オブジェクトを生成する．

包摂情報付き符号位置オブジェクトと同様に，Concord における符号化文字オブジェクトはその ID として IPLD の CID を用い，ID 素性 `=ipld` にも CID を格納する．

IPLD におけるメンバ `unify` は関係素性 `->unify` で表現する．これによりその素性値に含まれる各包摂情報付き符号位置オブジェクトに符号化文字オブジェクトへの逆関係素性 `->unify` が自動的に付与される．もし，符号化文字オブジェクトが再定義されれば，新たな符号化文字オブジェクトへの逆関係素性が蓄積されることになる．

また，利便性のために，包摂情報付き符号位置オブジェクトと同様に，対応する CHISE の文字オブジェクトを `character` 素性に格納している．

7.4 漢字構造オブジェクト

漢字構造情報を表現する IPLD オブジェクトに対しては `glyph` ジャンルの Concord オブジェクト（漢字構造オブジェクト）を生成する．

実用上，`coded-character` ジャンルを用いてもよいが，漢字構造記述には漢字の抽象形状を示すことを目的にしたものと文字がどういう音符・意符・形符の組合せから成り立っているかを示すことを目的にしたもの（解字）があり，両者を区別するために，現状の漢字構造情報は `glyph` ジャンルに入れ，将来的に解字情報を `coded-character` ジャンルに入れることを検討している．ただ，この問題に関しては本稿では詳述しない．

Concord における漢字構造オブジェクトはその ID として IPLD の CID を用い，ID 素性 `=ipld` にも CID を格納

[*7] 後者の場合，その形式のインデックスを扱うためのシステムが必要であり，前者の場合でも IPNS は IPLD よりも上位層のプロトコルであるため，IPLD だけで完結することはできない．なお，いずれの場合でもプログラムを JavaScript のような想定される環境で実行可能な形式で記述してプログラム自体を IPFS に載せ，IPNS 相対パスで実行するような環境を整えれば，インデックスに対応した版の処理系を含むシステム全体を IPFS で完結した形で配布することが可能である．もしそのシステム全体の CID に 1 つ以上の PIN が打たれた状態を維持できればその版の永続的保存が可能である．

[*8] インデックスを IPLD で実現した場合，大量（100 万個のオーダ）の小さなオブジェクトを扱う必要が生じるが，現在の go-ipfs 実装のローカルストレージはこうしたケースが苦手であり，現状ではキーバリューストアや RDF ストア，RDBMS などを使った方が現実的であると思われる．現状では，インデックス以外の IPLD オブジェクトに対してもこうした方法でキャッシュした方が現実的だと思われる．

する．

IDC を示すメンバ operator は同名の素性 operator で表現する．

IDC に対応する述語を示すメンバ名は接頭辞 -> を付けた関係素性で表現する．これにより部品オブジェクトに逆関係素性が自動的に付与される．

また，対応する coded-character ジャンルのオブジェクトに対して漢字構造オブジェクトへのリンクとなる関係素性 ->ideographic-structure を付与する．これにより漢字構造オブジェクトから coded-character ジャンルのオブジェクトへの逆関係素性が自動的に付与される．この結果，もし，複数のオブジェクトが同じ漢字構造を持つ場合，その集合が漢字構造オブジェクトの素性 <-ideographic-structure に集積することになる．

8. おわりに

CHISE の漢字構造情報の IPLD 化について述べた．漢字構造情報は有向グラフで表現される文字オブジェクトを部品として入れ子状に表現されるという複雑な構造をとるため，CHISE の RDF 化において一番難しい対象の 1 つであった．このため，IPLD 化においても難易度が高いことが予想されたが，本稿で述べた手法により IPLD 上で漢字構造情報を表現することができ，また，IPLD に基づく表現を用いることによって文字ではなく IDS で記述された部品の同一性を保証することができた．

IPLD ではデータの書き換えによってオブジェクトの内容 ID が変化してしまうため，変化しないオブジェクトを手がかりに変化したオブジェクトを探すための仕組みが必要である．また，逆リンクの実現にも同様の仕組みが必要であるといえる．本稿では Concord をこのための仕組みとして用いたが，IPFS の名前解決のための枠組である IPNS を用いる方法を検討することも重要な課題といえる．また，検索や照合，異体字処理といった文字処理を実現するためには関係素性や形態素情報といった漢字構造情報以外の素性も必要であり，文字情報サービスでは用例情報の利用も重要である．よって，CHISE 全体を IPLD 化することも今後行うべき重要な課題といえる．

参考文献

[1] Morioka, T.: Multiple-policy Character Annotation based on CHISE, *Journal of the Japanese Association for Digital Humanities*, Vol.1, No.1, pp.86–106 (2015).
[2] Lunde, K., Cook, R. and Jenkins, J.H. (Eds.): Unicode Ideographic Variation Database, Unicode Technical Standard #37, Revision 8 (2011).
[3] 守岡知彦：文字画像のマークアップの試み，東洋学へのコンピューター利用第 14 回研究セミナー，pp.21–30 (2003).
[4] International Organization for Standardization (ISO): *Information technology — Universal Coded Character Set* (*UCS*), ISO/IEC 10646:2014 (2014).
[5] 上地宏一：GlyphWiki, 入手先 ⟨http://glyphwiki.org/⟩.
[6] Benet, J.: IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3), arXiv preprint arXiv:1407.3561 (2014).
[7] Protocol Labs: IPFS is the Distributed Web, available from ⟨https://ipfs.io/⟩.
[8] Protocol Labs: IPLD, available from ⟨https://ipld.io/⟩.
[9] Dias, D.: IPLD—The “thin-waist” merkle dag format, available from ⟨https://github.com/ipld/specs/blob/master/IPLD.md⟩.
[10] Bormann, C. and Hoffman, P.: *Concise Binary Object Representation* (*CBOR*), Internet Engineering Task Force (IETF), RFC 7049 (2013).
[11] 守岡知彦：CHISE 漢字構造情報データベース，東洋学へのコンピューター利用第 17 回研究セミナー，pp.93–103 (2006).
[12] 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), 入手先 ⟨http://www.ipa.go.jp/NBP/13nendo/reports/explorat/charadb/charadb.pdf⟩.
[13] 日本規格協会，国立国語研究所，情報処理学会：平成 20 年度汎用電子情報交換環境整備プログラム成果報告書 (2009).
[14] 文字情報基盤整備事業，入手先 ⟨http://mojikiban.ipa.go.jp/⟩.
[15] 守岡知彦：CHISE の RDF 化の試み，情処研報，Vol.2017-CH-114, No.1, pp.1–6 (2017).
[16] 守岡知彦（著），山崎直樹（編）：漢字構造情報の RDF 化の試み，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，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，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」，pp.3–22 (2013).
[17] IRG Working Document Series, available from ⟨http://appsrv.cse.cuhk.edu.hk/~irg/irgwds.html⟩.
[18] 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).
[19] 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，じんもんこん 2010 論文集，情報処理学会シンポジウムシリーズ，Vol.2010, No.15, pp.33–40, 情報処理学会 (2010).
[20] 守岡知彦：古典中国語 UD コーパスの IPFS を用いた表現の試み，情処研報，Vol.2018-CH-118, No.6, pp.1–7 (2018).

守岡 知彦 （正会員）

1969 年生．1999 年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了，博士（情報科学）．1999 年電子技術総合研究所 COE 特別研究員．2000 年京都大学人文科学研究所附属漢字情報研究センター助手．2009 年同所附属東アジア人文情報学研究センター助教．漢字文献を中心とした人文情報学の研究に従事．2007 年度山下記念研究賞受賞．2016 年度日本語学会春季大会発表賞受賞．