漢字情報サービスの分権的構成のための要件について

守岡知彦（国文学研究資料館）

@じんもんこん 2024（2024年12月8日）

本日のトピック

- CHISE 文字オントロジーの分権化 (decentraisation)
  - 分権化とは何か?
    - 内容 ID
    - 配布の分権化と編集の分権化
  - 漢字情報の分権化が難しい理由と対策
    - CL-Concord / CL-CHISE での実装

CHISE：文字オントロジーに基づく文字処理
CHaracter Information Service Environment

- 文字に関する性質（漢字構造、部首、画数、音価、字義、用例情報、etc)に関する機械可読記述によって文字を表現
  - 文字符号非依存性、 → 文字符号に対するメタシステム
  - 編集可能性
    - わかっていることだけをとりあえず記録可能、いつでも修正可能
  - 多粒度漢字構造モデルに基づく文字オントロジー
    - 部分的な既知情報を突き合わせて正体を探るための手助け
    - 部品や構造レベルでの異体情報や歴史的字体用例の提供
    - 字種-(抽象文字)-字体-字形等のさまざまな包摂粒度でリンク可能な構造

CHISE-wiki (E_ST)：文字の詳細情報の表示

時 時

Abstract Glyph (IWDS-1)：<□・□> 日 <⿰・□><<土/扌>/<土/士>>寸

結合：

- + U+E0100：時

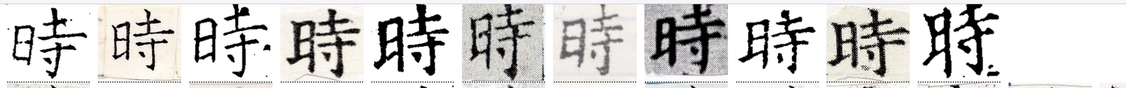
部首：日部 (R072)
画数：6
漢字構造：⿰ 日寺
反切：(反切)市之
Web韻圖 (廣韻)：/時/
総画数：10
= UCS：U+6642 (26178) [-][+]
= Big5：0xAEC9 (44745) [-][+]
← 字種：時

→ [HDIC] 篆隸萬象名義：  是之反。是也，善也，中也。

→ [HDIC] 宋本玉篇：  (反切)市之切。春夏秋冬四時也。

→ [HDIC] 新撰字鏡：  (晡時)　  市之之喜二反。四時也，是也，此也。　  之喜反。在日部。

→ [HNG] 中国写本： 

→ **[HNG] 中国版本**： 

→ **[HNG] 日本写本**： 

→ **[HNG] 日本版本**： 

→ **[HNG] 韓国資料**： 

→ **[HNG] その他**： 

→甲骨文字： 

→ 說文小篆： 

→ 古字： 

→ 古字(說文)：

→ 包摂：時 時

古典中国語形態素用例：微時 時晦 舊時 昔時 時論 時政 時忌 時望 相時 逢時 歲時 時氣 卽時 平時 或時 少時 異時 異時 日時 時人 時宜

東洋学文献類目用例（題名・キーワード等）：

- 李寅生著「論宋元時期的中日文化交流及相互影響」 巴蜀書社 (@成都), 2007年3月
- 守屋 美都雄編「四時纂要—中國古農書古歲時記の新資料」, 1961年11月
- 許雪姬訪問「日治時期在「滿洲」的台灣人」中央研究院近代史研究所口述歷史叢書;79, 中央研究院近代史研究所 (@臺北), 20
- 汪伯岩(au)「第二次國內革命戰爭時期的農村革命根嚴地」, 1955年2月
- 傅衣凌(au)「明清時代商人及商業資本」, 1956年7月

https://www.chise.org/est/view/character/時

CHISE IDS 漢字検索：部品で漢字を探す

https://www.chise.org/ids-find

https://www.chise.org
/ipns/ids-find.chise.org/index.ja.html
(Qme94UnoCa5fXGUNFnVByyf9sWdbSpbUQDhVsjHLvds4DA)

どうして分権化したいの？

- 一意な URL を永続的に維持することの困難さ
- 安定した Web サービスを行うための技術的・経済的問題
  - ドメイン名やサーバーの長期にわたる維持
    - データを持ってる研究者個人が長期間・安定的に（急激なアクセス数増加等やセキュリティリスクに対処しつつ）サービスし続けるのはあまり簡単ではない
- プラットフォームは便利だが、本当に信じて良いのか問題

➡検閲が容易なアーキテクチャーに学問の自由の根幹を任せて大丈夫?

IPFS とは

- InterPlanetary File System（惑星間ファイルシステム）の略
  - P2P 型分散ファイルシステム
  - Web3 （分権的アーキテクチャに基づく次世代 Web）
- Git と同様な内容に基づく参照 (content-addressing)

内容に基づく参照

- 内容アドレッシング (content addressing)
  - ID の指示対象はデータの内容（内容 ID; Content ID (CID)）
    - 例：Git（オブジェクトのハッシュ値）、(DOI)
      - 永続性のある ID を考える地獄と苦悩から解放される！
- 場所アドレッシング (location-addressing)
  - ID の指示対象はデータを置く場所
    - 例：IRI, 連番、住所、各種 ID 等

IPFS のプロトコルスタック

IPFS (UnixFS（以下「狭義の」）), …	アプリケーション
IPNS	名前解決
IPLD	データモデル（Merkle DAG に基づく）
libp2p (Bitswap)	ネットワークやルーチング、データ転送

IPLD

- Inter Planetary Linked Data の略
  - IPFS プロトコルスタックにおけるデータモデル層
- JSON と同様な構造データを扱う
- データの暗号学的ハッシュ（通常は SHA-256）値を用いて生成した内容 ID (CID) を用いてデータを参照する(Merkle Directed Acyclic Graphs (DAG; 有向非巡回グラフ))
- Web における Linked Data と同様なものを IPFS 上で実現するための基盤としてさまざまな試みが行われている

そろそろ実用的な分権化を

- （私の）IPFS/IPLD 化としては 4 度目の試み
  - 古典中国語 UD の IPLD 化 (CH-118)
  - 漢字構造情報のIPLD 化（oricom 30, 情処論文誌 Vol.61 No.2）

そろそろ実用化したい

- IPLD は魅力的だが、Web ブラウザでの対応を考えれば（狭義の）IPFS の方が現状無難?（だったのだが）

IPFS/IPLD を巡る最近の状況

- Bluesky の急成長で AT Protocol への注目が集まる
  - AT Protocol ではコンテンツを IPLD の dag-cbor 形式で表現
    - 現状、Bluesky は中央集権的だが、分権的構成の実現や別システムに移行可能な（ベンダーロックイン耐性の高い）プロトコルを目指している
  - DASL (Data-Addressed Structures & Links) の登場
  - IPLD のサブセット（形式を限定、P2P 不要；脱 IPFS 化）

DASL (Data-Addressed Structures & Links)

- IPLD の美味しいところだけを使う
  - dCBOR42 (RAW, CIDv1, dag-cbor, sha-256(or BLAKE3)) only
    - IPLD におけるいろんな選択肢が不要なので実装が簡単
  - CBOR (Concise Binary Object Representation): JSON と同様な構造データをバイナリーで表現する機械可読専用表現 (RFC 8949; STD 94)
    - 冗長性が少なく、データ量が少なく、パースが簡単になる
  - dCBOR (Deterministic CBOR)：CBOR を正規化 (draft-bormann-cbor-det-03)
  - dCBOR42：dCBOR + tag 42 (IPLD のリンク表現)

分権化のレベル

- 配布の分権化 (Lv1)
  - すでに出来上がったデータを IPFS 等で分権的に配信・配布・共有
  - （データ自体は中央集権的に編集しても良い）
- 編集の分権化 (Lv2)
  - 別個に作成されたデータを統合可能にする
    - 統合可能な部分は自動的に統合
    - 矛盾が生じてる部分（衝突箇所）を検出し、統合支援を行う

（Git がやってるようなことを（漢字情報専用で）もう少し賢くやる）

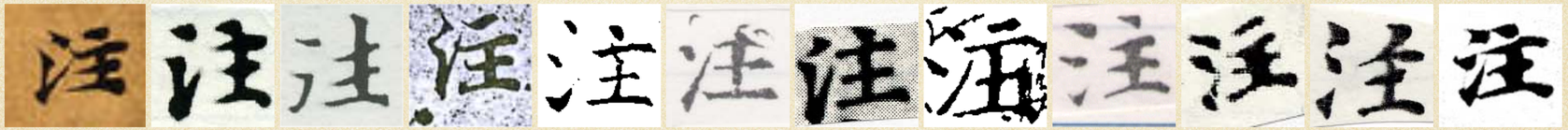
漢字情報の分権化の問題点

- CHISE の文字オブジェクトは素性の集合で表現されるので IPLD/DASL 化するのは容易
  - しかしながら、漢字は
    - 漢字構造記述：多くの漢字は部品の組み合わせで表現される
    - 異体字・類字関係

    で多数の文字が繋がっていて、一箇所書き換えると影響が波及する
    - 一つの文字・部品の CID が変わると多数のオブジェクトの CID が変わる
      - データの再利用性や統合が難しくなる

字形と字体

- 字形：紙に書かれたり印刷されたり石に彫られたり画面に表示された文字。その場所にのみ存在し、２つとして同じものはない



- 字体：社会的に共有された文字の抽象形状（書体依存）

字形と字体と字体記述

- 字体は字形の集合と見做すことができる
  - 外延的記述：字形を列挙して字体を示す（例：HNG）

    注 = {  , …}

  - 内包的記述：満たされるべき性質で字体を示す

    （例：CHISE の Chaon モデル）

    注 = <部首: 水部> ∧ <部首内画数: 5> ∧ <画数: 8> ∧ <漢字構造: ⿰氵主> ∧ …

    誰が書いても似た感じになりそう（分権化 (Lv2) しやすそう?）

字体記述の分権化を阻むもの

- 字体の外延的記述は物や場所、固有名依存性があって分権化しづらい（中央集権的に ID を決めて共有するしかない?）
- 字体の内包的記述は述語の論理積で表現でき分権化(Lv2)しやすい
  - 実際には、内包的記述の中に固有名が紛れ込みがち
  - 素性名や素性値等の形式の標準化は必要
  - 多数の関係が入り組んだグラフだと内包的記述といえども大変（IPLD/DASL で解決可能だが、CID の安定性に配慮が必要）

衝突を減らすための方策

- 字体の包摂ポリシーの記述（粒度情報付きID）をその他の記述から分離し、前者の形式的な記述の変更の影響範囲を減らす
- 字体の包摂ポリシーの記述における衝突可能性を減らす
  - 名目上の包摂粒度（例：〜の字体）と実際の包摂範囲の分離
  - 包摂範囲の関係（包含、交差、分離）を集合演算で判定
    - 生のCCS素性対の集合＋包摂粒度ラベルによる表現（分離型包摂情報付きCCS素性対方式）へ

今回の実装方針

- Common Lisp 版 Concord/CHISE (CL-Concord/CL-CHISE) を利用
- UCS 抽象文字以外の文字オブジェクトID を分離型包摂情報付きCCS素性対の IPLD/DASL CID で表現
- 素性値は従来方式（Valkey/Redis）で表現
- P2P を用いた分散化（分権化 Lv1）のことは今回は気にしない

実装

- CL-Concord での文字オブジェクト生成部を IPLD/DASL 化
  - Kubo (Go 版 IPFS 実装）の ipfs コマンドを毎回呼び出す実装だと IPLD/DASL 化前に比べて CL-CHISE ビルド（文字定義ファイルの読み込み＋IDS ファイルの読み込み＋漢字構造変換処理）にかかる時間が 46 倍に
  - Kubo RPC API を利用する実装だと 1.5 倍に抑えられた

今後の計画

- 文字オブジェクトIDの計算のCL-IPLD化
- 文字定義情報のIPLD/DASL化（分権化Lv1の実現）
- 別個に定義した文字定義情報を統合するための仕組み（GitHubにおけるpull requestみたいなもの）の検討

まとめ

- 分権化にもいろいろある

  配布の分権化(LV1) / 編集の分権化(LV2)
- 漢字情報の分権化(Lv2) が難しいのは字体の包摂範囲の記述と異体字記述や漢字構造記述が相互依存するから
  - 字体の包摂範囲の記述（粒度情報付きID）をその他の記述から分離
- Common Lisp 版 CHISE で実装してみた（ビルド時間が1.5倍になるのは許容できるか？）