漢字字体情報の安定的な交換について

守岡 知彦
京都大学 人文科学研究所

現在、IVD に登録されている Adobe-Japan1 と汎用電子はデザイン差に当たるような非常に細かい差異を区別している部分があり、そのままでは安定した情報交換に適さないと思われる。この問題を解決するためには、ある程度字形を包摂した『字体』レベルを適切に定義し、IVD に登録されているグリフとの関係を明らかにする必要があると思われる。また、Adobe-Japan1 と汎用電子を共存させる場合、複数の包摂ポリシーを適切に対応付け、両立可能にする必要があるといえる。本稿では、こうした問題について簡単に議論するとともに、包摂ポリシーが異なるグリフ集合を比較したり対応するための手法について検討する。

Toward stable information interchange of Kanji glyphs

MORIOKA, Tomohiko
Institute for Research in Humanities, Kyoto University

Adobe-Japan1 and Han'yo-Denshi collections, which are registered in IVD, may distinguish very small glyph-differences. In the view, each IVS represents a glyph-image, not an abstract glyph. It is bad for stable information interchanges. To resolve it, we need suitable definition of abstract glyphs (*Zitai*) and map them to glyphs registered in IVD. In addition, we need a method to integrate multiple unification policies of glyphs. This paper describes about such kind of problems about Kanji glyphs, and proposes a method to integrate Adobe-Japan1 and Han'yo-Denshi collections.

1 はじめに

IVS (Ideographic Variation Sequence) のように、符号化文字列において漢字グリフを情報交換する場合の問題点とその対策について議論する。

IVS は UCS 統合漢字で包摂された複数の字体を区別するための枠組を与えているが、そこで指示される単位が何であるかは明確に規定していないといえる。むしろ、異なる包摂規準のグリフセットを共存させるための仕組みが IVS といえ、統合・分離の規準は IVD (Ideographic Variation Database) [1] に登録されるコレクションの側によって規定されるといえる。

現在、IVD に登録されているコレクションには Adobe-Japan1 由来のものと「汎用電子情報交換環境整備プログラム」[2]（以下、「汎用電子」と略す）由来のものがある。後者は文字の同定・分離に関して一応の判断規準を設けているが、前者には明示された規定は存在せず、（結果的に）「冴」と「冴」のような細かい差異も区別するようなものとなっている。一方、後者でも「伶」と「伶」のようなデザイン差と思われるものが区別されている。また、「八」と「八」や「交」と「交」のような両者で共通して区別されている細かい差異もある。

両者は JIS X 0208/0212/0213 などいくつかの共通するソースを持っており、また、それ以外にも似た例示字形を持つ文字が存在し、少なからぬ文字が重複していると考えられる。しかしながら、IVS が指し示すもの（範囲）が何であるかについて明確な包摂規準はなく、その同一性について形式的に判断することができない。また、片方での

み区別されている差異があることから、両者の包摂規準が異なるのは明らかである。

こうした細かい差異の存在はしばしば一般的な日本語使用者の直観に反するものになりかねないといえ、実際に IVD を精査しないと何が区別され何が区別されないかは判らないし、また、将来の追加を考慮すれば更に謎めいたものとなってしまう。おそらく、本来必要であったのは、UCS の漢字統合で過剰に包摂される字体レベルの差異[1]であったと思われるが、現状の IVS は字形レベルの何かを指示するものとして扱わざるを得ないといえる。これは書体差やフォントのデザイン差等を考慮すれば細かすぎて安定性に欠くものといえ、UCS の抽象文字よりも細かく、現状の IVS よりも荒い、『字体』に相当するような包摂レベルを設定する必要があると思われる。しかしながら、『字体』の観念が仮にある程度共有されるようなものとして存在していたとしても、相当程度の揺れがあり、用途やポリシー等によって異なる包摂規準を用いざるを得ないことも想像される。よって、『字体レベル』の包摂を導入した場合、どの包摂ポリシーを用いているかを明示するための仕組みや複数の包摂規準間の関係を判断できる仕組みが必要であると考えられる。また、少なくとも、抽象文字と IVD に登録されている既存の IVS コレクション（『字形レベル』の何か）を含めた 3 レベルを関係づける必要はあるといえる。ここでは、こうした『字体レベル』の包摂を導入した場合の情報交換における課題や考えられる方策について議論したい。

2 IVS

IVS というのは、ベースとなる文字の後に VS (Variation Selector) [3] と呼ばれる結合文字 (combining character) を置くことで特定の異体字を表現できるようにしたものである。

VS は結合文字の一種であり、IVS は構文論的にはアルファベット系文字等での文字結合と似ているが、結合型アクセント記号と異なり、VS を複数重ねることはできない。VS は全部で 256 個あるが、漢字で使えるものは <VARIATION SELECTOR-17> (U+E0100) 〜 <VARIATION SELECTOR-256> (U+E01EF) の 240 個であり、結局、親字となる統合漢字に 240 個の枝番を振った符号体系と看做すことができる（つまり、Unicode の 21bit を 4bit 拡張し、25 bit にした符号体系と考えることができる）。

3 字体の符号化に関わる問題

3.1 字体を考えるための 2 つの視点

『字体』というものを考える場合、さまざまなアプローチがあると思われるが、ここでは書記言語の構成要素としての『字体』という観点で考える。音声言語に対しては、ある言語において意味を担う最小の単位である「音素」と、それらが実際に発声された「音声」という 2 つの異なる概念（音韻論と音声学という異なる分野）が存在するが、書記言語の構成要素としてのグリフに関しても同様の区別が必要であると思われる。

[2] の 2.5.1 節では『字体』と『字形』を音韻（音素）と音声に対応付けて考えているが、『字体』という抽象的な視覚的記号の単位を設定するためには外形的特徴（類似性）をどこまで保存しどれを無視するかを考える必要があり、これは書記言語の使用者の知識や社会的合意に依存した面があることを考えれば、単純に『字体』を『音素』に対応づけるだけでは不十分であるといえる。よって、（ある書記言語に習熟した（ある書記言語を『母語』とする）者の直観に基づいた）意味弁別の観点から設定されたグリフの単位（そうした視点）と、そうした言語や知識に依存することなく外形的特徴だけから決定可能なグリフの単位（そうした視点）を想定することが望ましいと考える。[2]

『書記言語的直観』や『知識』は実際には揺れやすく、また、時代や地域、コミュニティーや分野、

[1] 新常用漢字のように文字政策が変化した場合に問題となる。

[2] この 2 つの視点は、原理的にはいろんな粒度においても存在し得ると思われる。

テキスト等によって変化しやすいと考えられる。このため、今日の情報交換用符号のあり方を考えれば、情報交換用符号で指示されるような『字体』レベルの単位は、特定の言語や特定の知識になるべく依存しない方が望ましいといえる。しかしながら、こうした『書記言語的直観』や『知識』なしに字形の集合を分節化することはできないということもいえる。よって、テキストアーカイブズ等では『字体』という単位を成り立たしめている前提やどの弁別的特徴を設定し、どの精度まで記述できているかを、併せて記録できることが望ましいと考えられる。

3.2 包含関係を多階層化する場合の問題

IVS のような枝番方式は抽象文字とグリフを木構造の親子関係として表現しているが、このような親子関係の木を仮に抽象文字とグリフの２階層から抽象文字と字体と字形の３階層に拡張することを考える。もし、3 階層（以上に）拡張した場合においても、IVS のような２階層の場合と同様に、木構造で表現できたとしたら、包含関係を簡単に表現でき、操作も容易である。

しかしながら、もし JIS X 0208:1997 のように部品や漢字構造のパターンに対する書き換え規則（あるいは、同等な内包的表現された抽象部品等）として包摂規準を記述しようとした場合、仮に単一の部品が木構造で記述できたとしても、一般に漢字は複数の部品から構成され得るので、もし、複数の部品のそれぞれに字体レベルの異体・変種が存在し、また、その字体レベルの異体・変種に対して複数の字形レベルの異体・変種が存在した場合、字体レベルと字形レベルのそれぞれに各部品が取り得るそれぞれのレベルの異体・変種の順列組合せの異体・変種が存在し得、字体レベルと字形レベルの関係は木構造で表現できなくなる（図 1）。

もし包摂規準間に優先順位を付けることができれば、木構造に落し込ことは可能である (図 2) が、この場合、優先順位の低い包摂規準に対する親子関係が複数の木に分割されてしまうという問題点がある。また、木構造が３階層に収まらなくなるという問題もある（図 3)。また、便宜的に付けた優先順位が不自然なものだった場合、包含関係の木構造は直観に反したものとなる恐れがある。

また、もし、『字体レベル』の包摂規準の集合 (包摂ポリシー）を一意に決めることができず、複数の包摂ポリシーを使い分ける必要があった場合、さらに問題は複雑となる。この場合、字体レベルに相当する単位を複数用意する必要があり、また、前述のような理由から仮に２つの包摂ポリシー A, B 間の各包摂規準 a_i, b_j が常に単純な包含関係になっていたとしても、両者に属するオブジェクト間の関係は単純な木構造にならないといえるが、一般には、各包摂規準も単純な包含関係にならないので、より複雑な事態となる得る。

3.3 包摂範囲の衝突

歴史的事情から、UCS の統合漢字には、その包摂規準に従えば、本来、単一の抽象文字で包摂されるべきものの幾つかに異なるコードポイントを振っている場合がある。これには『原規格分離』と呼ばれる意図された例外の他に、統合漢字拡張 B までの手作業によるチェック洩れによる重複と思われるケースがある。いずれにしても、統合漢字は、理念としては、抽象文字がある範囲の字体や字形の集合を包摂するとしていたものの、当初は実際にその包摂範囲を明確に定義することができておらず、例示字形から類推するしかなかった。また、統合漢字における包摂は、実際の用例に対してではなく、ソースとした文字表（符号化文字集合の規格や辞書等）とのマッピングの形で行われており、計算機における文字処理においても、こうしたマッピングを適切に扱うことが求められ、古典文献の電子化といった例外的な用途を除けば、さまざまな字体や字形が実際に包摂し得るかどうかを機械的に判断するといったことはあまり必要なかったといえる。

しかしながら、IVS が登場したこと（そして、複数のコレクションが登録されるようになったこと）により、基底文字となる統合漢字と IVS に

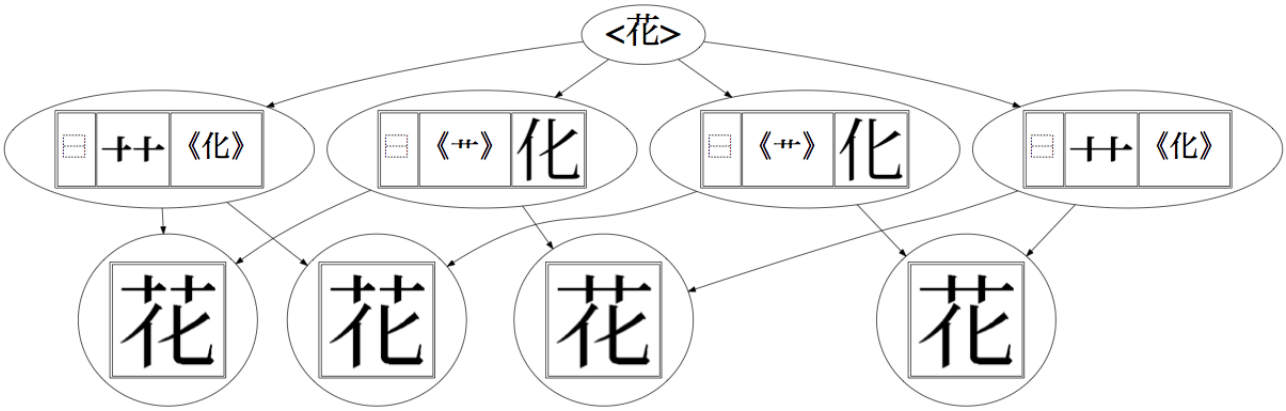

図 1: 複数の部品に変種がある場合

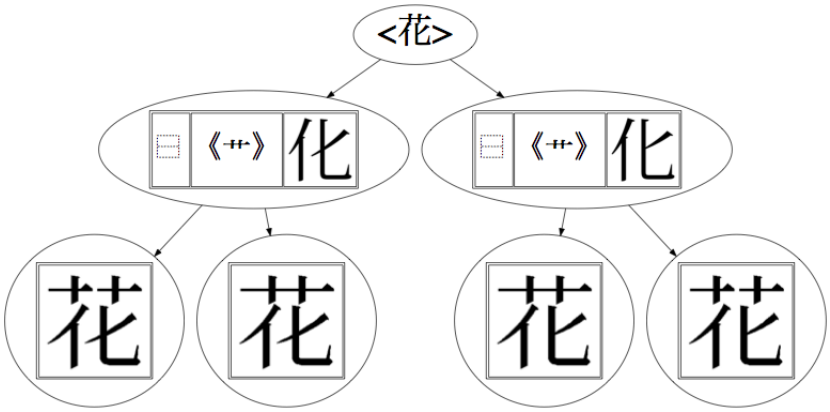

図 2: 《化》の差異を優先した場合

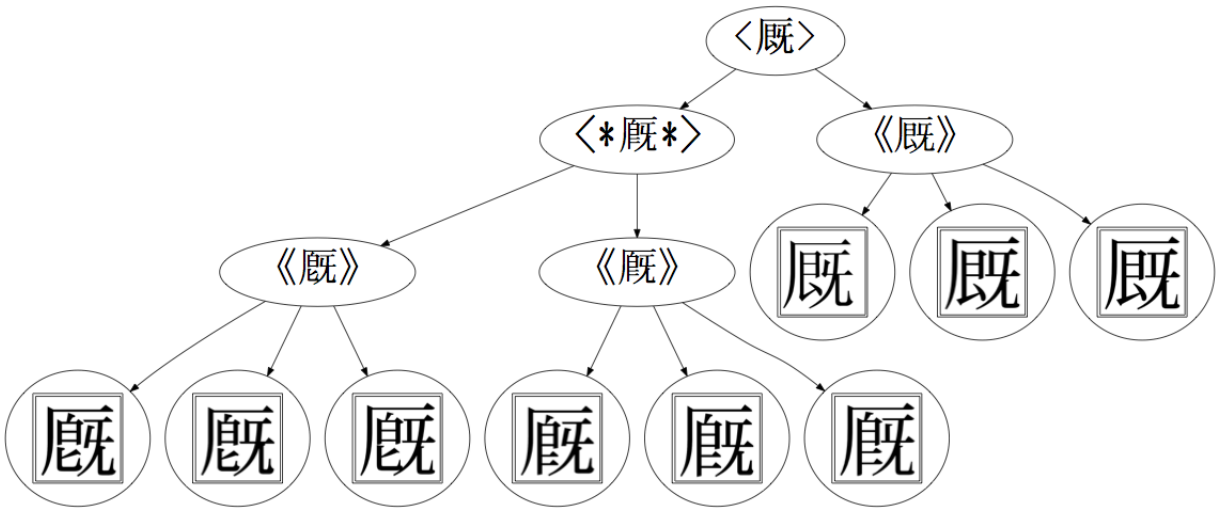

図 3: U+53A9 を木構造化した例

よって指示されるグリフとの関係を把握する必要ができた。IVS によって指示されるグリフはその基底文字となる統合漢字で包摂される必要がある。[3] また、IVD にグリフを登録したい場合、あるグリフを包摂し得る統合漢字が複数ある場合に、どの統合漢字を基底文字とするかの恣意性が生じることとなる。こうした場合、複数の統合漢字を基底文字とする複数の IVS からそのグリフを指示するようにするという選択もあり得、実際、Adobe-Japan1 はこうしたことを行っている。

こうした UCS 固有の問題を別にしても、実際にさまざまなテキストに現れるさまざまな字体・字形を符号化する場合、明示された規準に基づく包摂範囲を拡張したり、文脈や知識等に基づき外形的特徴だけでは包摂し得ないような字体・字形を抽象文字・字体に対応づける必要がある場合がある。こうした対応を JIS X 0208:1997 のような字体の書き換え規則として表現した場合、これを機械的に適用すると意図しない包摂関係が生じるおそれがある（それを避けようとする場合、多数の包摂除外を設けなければならないかも知れない)。例えば、木偏を手偏のように表現するからといって、この両者を単純に包摂するような規準を設けるのはまずいということは容易に想像できるだろう。

『原規格分離』のようなケースにせよ、異形字を対応させるために包摂規準を拡張する場合にせよ、結果として、複数の抽象文字（あるいは、字体）の包摂範囲が重なってしまった場合、なんらかのルールを設けて包摂範囲が重ならないように工夫するか、包摂範囲の重なりを許容するか、何らかの方策が必要であるといえる。

4　グリフ情報をどう管理するか

4.1　多粒度漢字構造情報

複数の部品から構成される漢字の抽象形状は、使われる部品とその組み合わせ方の組みで表現できる。このための記法としては IDS (Ideographic Description Sequence) 形式 [3] が広く用いられているが、ここで使われる部品と組み合わせ方を指定するオペレーターのどちらか、あるいは、その双方に対して、UCS にある抽象文字ではなく、範囲指定を許すようなオブジェクトに置き換えたものを考えることができる。[4] もし、部品に対して予め、抽象文字 -『字体』-『字形レベル』等の各レベルの変種を含むオブジェクト間の関係を予め定義しておけば、それを組み合わせることによって、さまざまな包摂範囲を表現できるといえる。[5]

例えば、3画の草冠「艹」と4画の草冠「艹」がある時、この両者を包摂するオブジェクトを《艹》とする。また、付き出ていない形の「化」と付き出てる形の「化」がある時に、この両者を包摂するオブジェクトを《化》とする。この時、「花」は「艹 化」、「花」は「艹化」、この両者を包摂する3画の草冠を持つ『花』の集合は「艹《化》」と書くことができる（図 1)。

4.2　階層化できない場合

一般に3階層以上の包含関係が木構造にならないとしても、もし、各オブジェクトが CHISE の Chaon モデル [6] のように素性の集合で表現されている場合、集合演算を用いることでオブジェクト間の関係を調べることができる。また、その素性として、4.1 節で述べた『多粒度漢字構造情報』を用いることで、各オブジェクトの包摂範囲を表現することもできる。

しかしながら、交差している場合の処理をどのように行うかを考える必要があるといえる。また、素性の集合演算は階層構造を利用した処理に比べてコストが高いので、ファイル入出力のような高速性を要求される処理には向かないといえる。よって、包摂ポリシー毎に抽象文字 -『字体』-『字形』間の関係を木構造化して表現したインデックスを用意するのが望ましいと考えられる。このためには、包摂ポリシーを定義したら包含関係の木構造が生成されるような仕組みがあれば良いと考えら

[3]現実には、包摂されないと思われるものも登録されているが…

れる。

4.3 階層化する場合

一般に 3 階層以上の包含関係が木構造にならないとしても、その大部分が木構造の範囲に収まるか木構造で無理なく近似することができるとすれば、抽象文字 -『字体』-『字形レベル』等のオブジェクト間の関係を記述したグリフオントロジー [5] の記述量を削減するために、木構造で記述可能な部分は木構造で記述するのが望ましいといえる。

実際、IVS を持つ統合漢字の総数に対して、なんらかの問題が起こり得る可能性がある統合漢字の数は 1 % 程度であり、大部分を木構造で書くことは問題がないといえる。

ただ、Adobe-Japan1 と汎用電子の包摂ポリシー（個々のコードポイントの包摂範囲）は一致しない部分があるので、この両者の各々から類推される包摂範囲・ポリシーをもって単一の『字体レベル』を設定することはできず、場合によっては、『字体レベル』に相当する階層を複数設ける必要があるといえる。

5 『字体レベル』の包摂規準

5.1 汎用電子の包摂ポリシー

汎用電子では、文字の同定・分離に関して [2] の 4.2.3.3 節で

1. 部分字形の構成の違いがある場合は異体字とする
2. 画数の違いがある場合は異体字とする
3. 運筆の方向違いがある場合はデザイン差とはみない
4. 部分字形の配置の違いがある場合は異体字とする
5. 新字体・旧字体の差がある場合は異体字とする
6. 一見して部分形状の差が顕著である場合はデザイン差とはみない

という判断基準を設けている。

一方、[2] の 2.5.1 節で述べているように、ソースとなる文字セットに対して 1 対 1 対応させることを基本としているため、住基統一文字や戸籍統一文字で別のコードポイントが振られているものに対しては、この判断規準では同じ字体として統合されるべきケースであっても、分離することになっている。

また、この判断規準では同じ字体に統合されるべき場合であっても、個々の文字を検討した結果、デザイン差としないことになったケースもある。例えば、[2] の 4.2.3.3 節の「部分字形の構成の違い」に関する解説で「立」の 1 画目が縦か横かという違いをデザイン差（異体字とはいえない）としているのに、同 4.2.4 節「検討結果の一例」では、「音」の 1 画目の縦横の差をデザイン差としないとしている。

こうしたことから、汎用電子の実際の包摂ポリシーはその個々の文字を見て帰納的に考えるしかないようであるが、「八」と「八」、「月」と「月」、「音」と「音」、「豕」と「豕」、「牙」と「牙」、「旡」と「旡」、「巨」と「巨」、「成」と「成」、「弍」と「弍」の「三」（変形の有無）といった部品は概ね区別されているようである。

5.2 Adobe-Japan1 の包摂ポリシー

Adobe-Japan1 には明示された包摂規準はないが、汎用電子と同様に筆押えの有無を区別している例がある。また、JIS X 0208 の変更点に関わる箇所に関しては細かい差異を区別しているように思われる。一方、汎用電子のように画数の違いがある場合は区別するというような基準はないようで、3 画の草冠「艹」と 4 画の草冠「艹」を包摂していると考えられる。

5.3 従来の CHISE における『字体レベル』

CHISE 文字オントロジーでは、従来、抽象文字、『字体』、『抽象字形』、例示字形という包摂レベルを設けていた。これらは、抽象オブジェクトと具象オブジェクトの継承関係として扱われており、そのための関係素性 `->subsumptive` と `->denotational` が用意されている。前者は差異が比較的小さな場合に用いられるもので、原則として、字形レベル以下の包摂関係に用いることになっている。一方、後者は差異が比較的大きな場合に用いるもので、原則として、字体レベル以上の包摂関係に用いることになっている。

CHISE 文字オントロジーではこの２つの素性の使い分けによって表現されるものの他にも、派生的なグリフ ID 素性名の命名規則 [5] を用いた記述もなされている。あるグリフ・リソース *foo* に対し、例示字形の素性名を *=foo*, 『抽象字形』レベルの素性名を *=>>>foo*,『字体』レベルの素性名を *=>>foo*, 抽象文字（字体の包摂レベル）の素性名を *=>foo*, 『超抽象文字』（抽象文字を包摂したレベル）の素性名を *==>foo* とするもので、これらの素性名を用いることで、どの包摂レベルを表現しているかを明示することができる。

一方、従来の CHISE における『字体』レベルの包摂規準は、汎用電子 [2] の 4.2.3.3 節で述べられている「デザイン観点からの同定基準」のうち、1., 3., 4., 5. は概ね共通するが、2. に関しては CHISE 文字オントロジーでは字体レベルの差異と看做しているものと看做していないものがある。例えば、「辶」と「辶」や「艹」と「⺾」等は字体レベルの差異として扱っているが、「韋」と「韋」等は字形デザイン差の一種として扱っている。

5.4 包摂ポリシーの統合

汎用電子にせよ Adobe-Japan1 にせよ、あるいは、CHISE 文字オントロジーにせよ、実際に収録されているグリフの包摂範囲は必ずしも一貫しているとはいえず、多かれ少なかれ、例外や揺れを含んでいるといえる。よって、例外的に細かく分離している部分にのみ着目することは妥当とはいえず、包摂の粒度が一番荒い部分（これを『包摂レベルの上限』と呼ぶことにする）と一番細かい部分（これを『包摂レベルの下限』と呼ぶことにする）という幅を持った包摂レベルとして捉える方が妥当であると思われる。[4]

こうした観点でいえば、汎用電子の包摂レベルの上限は [2] の 4.2.3.3 節で述べられている「デザイン観点からの同定基準」に相当するものといえ、下限は「伶」と「伶」や「巻」と「巻」を区別するような『字形レベル』に相当するものといえる。

一方、Adobe-Japan1 の包摂レベルの上限は汎用電子の包摂レベルの上限より高く、３画の草冠「艹」と４画の草冠「⺾」を包摂するようなものと看做すことができる。また、下限は汎用電子の包摂レベルの下限より低く、「丈」と「丈」のような筆押えの有無も区別するような『字形レベル』に相当するものといえる。

このような包摂レベルの上限と下限を対応させた場合、概ね

> Adobe-Japan1 の上限 > CHISE の『字体』レベル ≒ 汎用電子の上限 > 汎用電子の下限 ≒ Adobe-Japan1 の下限

という風に看做すことができる。

そこで、CHISE の『字体』レベルを２つのレベルに分離し、包摂度が高い方を『統合字体』レベル、包摂度の低い方を『字体レベル』とすることにした。そして、この新しい『字体レベル』を汎用電子の包摂レベルの上限になるべく一致させることにし、汎用電子の包摂レベルの上限よりも高い部分を『統合字体』レベルに移すことにする。また、Adobe-Japan1 の包摂レベルの上限は『統合字体』レベルとする。一方、Adobe-Japan1 と汎用電子の下限を『抽象字形』レベルとする。この修正に基づき、派生的なグリフ ID 素性名の命名規則に『統合字体』レベルを表す *=+>foo* を追加する。

この３つの包摂レベルに基づき、IVD に登録されている各グリフに対して、例示字形オブジェク

[4] この時、包摂レベルの上限と下限の差の小ささによってグリフ集合（の包摂規準）の品質を評価することができる。

ト、『抽象字形』オブジェクト、『字体』オブジェクト、『統合字体』オブジェクトを対応させ、CHISE 文字オントロジーに配置する。Adobe-Japan1 と汎用電子の各グリフは各レベルの包摂ポリシーに基づき適切に統合するが、それぞれのコレクションで使い分けがなく『抽象字形』レベルで対応すると思われるものの例示字形に無視できない差異があるものに対しては、例示字形オブジェクトは統合せず、別オブジェクトとする。

このように配置した各レベルのオブジェクトは、包摂レベルを指定したグリフ ID 素性名によって参照することができる。例えば、Adobe-Japan1 の包摂レベルの上限は `=+>adobe-japan1`,『字体レベル』は `=>>adobe-japan1` で参照できる。また、各オブジェクトはオブジェクト間の継承関係によって、各上限・下限に対応するオブジェクトや汎用電子との関係を知ることも可能である。

6 おわりに

本稿では漢字の字体レベルの情報交換や複数の包摂ポリシーを共存させる場合の問題について簡単に議論した。

抽象文字、『字体』、『字形』といった 3 階層以上の包摂レベルを扱おうとした場合、各包摂レベル間のオブジェクトの抽象・具象関係は一般には木構造の範囲に収まらず、記述量や計算コスト、判りやすさ等の点で問題が生じる。これは包摂規準に優先順位を付けることである程度解決できるといえるが、便宜上の中間ノードが必要となり包摂レベルが 3 階層に収まらず、また、包含関係が判りにくいものとなる恐れがある。

ただ、現実には 3 階層程度の比較的綺麗な木構造で表現できる例も少なくなく、部品の異体・変種の組合せ爆発を起こす場合にのみ例外的に対応することで現実的な解決が可能であるといえる。このためには、Adobe-Japan1 と汎用電子を包含するグリフオントロジーを実際に構築することが望ましいといえ、現在、この作業を行っている。また、包摂ポリシーの記述は包摂規準のセットとその優先順位（および、例外処理の仕様）が必要であると考えられるが、その記述量を削減するためには、よく使われる代表的な包摂ポリシーを幾つか定義することが重要であろう。

参考文献

[1] : Ideographic Variation Database, `http://unicode.org/ivd/`.

[2] 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，`http://www.meti.go.jp/information/downloadfiles/c100806a04j.pdf` (2008).

[3] International Organization for Standardization (ISO): *Information technology — Universal Multiple-Octet Coded Character Set (UCS)* (2003). ISO/IEC 10646:2003.

[4] 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).

[5] 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).

[6] Morioka, T.: CHISE: Character Processing based on Character Ontology, *Large-scale Knowledge Resources (LKR2008)*, LNAI, No. 4938, pp. 148–162 (2008).