KAGE - An Automatic Glyph Generating Engine For Large Character Code Set

上地 宏一

要旨

漢字部品を組み合わせてさまざまな漢字グリフ（フォント）を動的に生成する KAGE システムについて紹介する。KAGE システムは、主に既存の符号化文字集合（文字コード）には含まれない漢字を処理できるコンピュータ環境の実現のためのフォント未実装字の具視化を目的とし、その手段として複数の漢字部品を組み合わせることによる字形生成技術を利用する。漢字部品はその形状的な特徴や、位置構造によって大きさや配置場所を一定のルールにしたがって計算される。また生成されたグリフの品質に問題がある場合や、用意されていない漢字部品を利用したいときなどのために、グリフデザインツールを用意し、グリフの再調整や部品の新規作成が可能で、その成果を共有のデータベースに登録し、再利用することが可能である。本稿ではシステムの現状と今後の展望について説明する。

1 はじめに

現在の文字処理は、すべて符号化文字集合の枠組みで行われている。文字の表現についても、多くは符号化文字集合をもとにしたフォントセットが利用されている。一方で符号化文字集合に採録されていない文字や、字形の異なるグリフを利用したい場合、従来はユーザー定義文字（外字）として自由に定義することができたが、定義したコンピュータのみで利用する手段であり情報交換の障害となるため、ネットワークが発達した今日では適切な手段とはいえない。このような場合、ある程度流通している外字フォントといわれる製品などを利用することになるが、グリフの提供者となるフォントベンダーによって利用できる文字が制限されることになり、ユーザは受動的立場に置かれている。

これらの問題に対して、真の多漢字・多字形字形処理環境を目指すためには、ユーザが任意の漢字グリフを自由に作成（利用）でき、さらに情報交換が保証される処理系の開発が必要である。そしてテキストを具視化するフォントは制限なく利用できることが重要である。このためには、自分の手元で任意の漢字グリフを生成できる仕組みが必要であるが、アウトラインフォント（グリフ）を新たにデザインすることは、手間がかかるため実用的ではない。そこで漢字の特徴にしたがって、複数の漢字部品や筆画を合成して漢字グリフを低コストで自動的に生成する方法が現実的である。

このような観点に立ち、著者は 1998 年から KAGE システムの開発に着手し、2001 年にはエンジン部分を実装した [5]。その後、生成文字品質を向上させるために研究を継続しているが、現在は、生成されたグリフのデザインをユーザが自由に調整できるシステムを開発している。さらに、2002 年から CHISE プロジェクト [1] に参加し、多文字処理技術に KAGE システムを応用するための研究を行っている。

キーワード： Kanji Variants Font Server

2 新たなシステムの構築

KAGE (Kanji-glyph Automatic Generating Engine) システムは、漢字部品を合成して漢字グリフを生成する KAGE エンジンを基本としている。複数の漢字部品を組み合わせて漢字グリフを生成する手法は、19 世紀の中国における活字字母製作でも提案されていた技術であり、近年では「和田研フォント」と称される（当時の）東京大学和田英一研究室で開発されたフォント生成技術 [7, 8] が記憶に新しい。

2.1 漢字グリフ生成技術

通常、漢字部品の合成による漢字グリフの生成技術と言われるものは、コンピュータリソースに制限があるために、少ないデータ量で数千種の漢字フォントを用意する目的で研究されてきた。具体的には次のような手法が用いられた。

漢字部品の任意の大きさへの拡大・縮小

- 部品をいくつかの大きさ毎にデータ化する
- 明朝体の特徴的なウロコや撥ね、払いの形状を部品化 (独立）し、基本部分を拡大・縮小したあとに独立部品を合成して擬似的に処理する [6]
- 漢字字形を芯線（スケルトン）と肉付け方法に分割してデータ化する

漢字部品の大きさ、位置の決定

- 漢字全文字種に、部品の種類、大きさ、位置をパラメータ付加する
- 部品に対して大きさの係数を割り当て、部品同士の相対値で大きさを決定する
- 部品の特徴に応じて、大きさや配置を自動的に計算する

その他

- 漢字を筆画毎に分割し、筆画の輪郭情報（大きさ毎に複数保有する）を組み合わせてフォントを実装する

既存の研究の多くは、閉じられた漢字集合（主にJIS 漢字と呼ばれる JIS X 0208 が対象）をいかに省コストでフォント実装するかに着目していたため、多くはパラメータ付与型の技術となっている。特に、90年代のワードプロセッサの普及において、少ないメモリで複数の書体を搭載した製品が多く開発されたが、ここにもこれらの技術が応用されてきた。しかし、近年のコンピュータは高性能となり、リソースに制限がなくなったため、これらの省メモリ重視のフォントは文字品質の点で劣っていることから、よりアナログ原字に近い、１文字ごとに調整された高品質フォントが普及するに至っている。一部のフォントベンダーでは、自動的に生成されたフォントをベースとして利用し、漢字集合内の同じ部品同士の字形の統一や開発期間の短縮に活用している。

KAGE システムでは、フォント実装されていない（もしくは実装されていてもライセンスなどの制限によって利用できない）漢字グリフを、フォントデザインの専門知識を持たないユーザが簡単に利用する、という目的のため、再度この漢字グリフ生成技術に着目した。特に、KAGE システムのグリフ生成対象は、閉じられていない漢字集合となるため、パラメータ付与型の技術は利用できない。そこで部品の特徴を元に「自動」的に部品の大きさや配置を計算する技術に着目する必要があった。本稿ではこの技術を特に「グリフ自動生成技術」と呼び、パラメータ付与型の「フォント生成技術」と区別する。

2.2　新たな生成エンジンの構築

「和田研フォント」は、当時高価だったアウトラインフォント（ベクトルフォント）を低コストでデザインする目的で研究され、その手段として漢字部品をスケルトン骨格データと肉付けによる書体修飾とに分離したデータ化と、部品の特徴による大きさと配置の半自動計算ルールを利用することで一定の成果を上げた。しかし、アウトラインフォントの普及や生成される文字品質の問題により、現在ではフォントそのものは普及していない。KAGE システムを開発する段階で、和田研フォントの技術は閉じられていない漢字集合のグリフ実装に最適であると考えたが、すでに和田研フォントの研究は終了し、またデータも公開されていなかったため（現在は公開されている）、論文などを参考にして新たにグリフ自動生成エンジンを再実装した。目的の相違から、和田研フォントのグリフ生成エンジンとは異なる特徴を持つに至った。

3　KAGE システムの特徴

3.1　KAGE エンジンの実装

拡張 IDS による漢字記述

KAGE エンジンは、ユーザが必要とする漢字字形を必要になったときに動的に生成する必要がある。ここでフォントを生成するためにフォントエディタを使っていては時間がかかってしまい、実用的ではない。そこで、必要とする漢字が漢字部品の集合から成り立っていると仮定して、その漢字部品とその構造をドキュメントに記述しておくと、エンジンが解釈してグリフをその場で生成し、具視化する仕組みを考える。ここで生成すべき漢字字形を表現するために IDS (Ideographic Description Sequence) [2, 3] の拡張表現を利用している。本来の IDS と異なる点としては、IDC (Ideographic Description Character) のうち、U+2FFB (ideographic description character overlaid) はさまざまな重ね合わせが考えられ、字形が一意に特定できないために廃止した。また漢字部品として任意の漢字が利用できるが、元来 UCS のコードポイントは具体的な字形を特定せず、例えば国や地域によっても想定される字形が異なってくることから、字形を固定するために詳細字形コードを付加して部品の字形を特定することができる。この他、UCS コード以外にも、漢字部品データベース (詳細は後述) に登録されているさまざまな漢字部品を利用することが可能となっている。

このように KAGE エンジンでは IDS 拡張表現によってさまざまな漢字を記述することが可能である。漢字部品そのものは有限の集合であるため、全く未知の漢字に対して必ずしも IDS で表現できるとは限らないが、その場合は漢字部品の追加によって対処することになる。

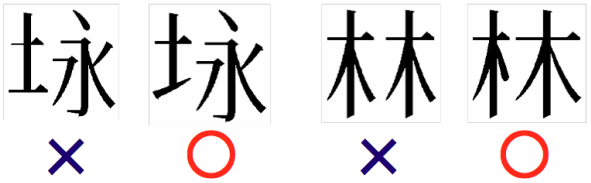

図 1: 左右結合の左部品の変形

IDS 記述の前処理

IDS で漢字を記述する際には、いくつか問題点がある。そのひとつに部品変形がある（図 1）。図のように「土」や「木」は、左右結合の左側に部品が位置する場合、変形現象が起こる。しかし変形部品は通常の UCS には含まれず、また、あえて変形しない部品を用いたい、というケースは少ないため、ユーザが変形部品を IDS に記述することは繁雑である。そこで KAGE エンジンでは、入力された IDS 記述を解析して、これらの変形処理を自動処理する前処理を行う。具体的には変形ルールに適合し、かつ変形された部品が漢字部品として用意されている場合は変形部品へ置換する。

字形の骨格表現と肉付けによる書体表現

IDS で記述された漢字の構造に沿って、KAGE エンジンが漢字部品を任意の大きさに拡大・縮小して配置することになるが、ここで部品データを輪郭形式で保持している場合、拡大・縮小処理によって筆画の太さがバラバラになってしまう。そこで KAGE エンジンでは和田研フォントと同様に、部品をスケルトン形式で保有する。和田研フォントでは 16 種類のストロークデータがあるが、KAGE システムでは、さらに共通化して直線と曲線の 2 種類の基本ストロークを元に、5 種類の応用ストロークを用意している（図 2）。各ストロークは始点、終点および 1,2 点の制御点からなる 2〜4 点の座標で表現する。各ストロークの頭部と尾部の形状を指定することで「撥ね」や「払い」、「ウロコ」を表現する。

また、箱型の縦棒のトメ（下に突き出す部分）は、自動的に付加されるため、例えば「口」という文字は 4 つの座標からなる 4 線分の組み合わせで表現することで、左下と右下のトメの突き出しは自動的に再現される（図 3）。

曲線は、TrueType フォントと親和性を高めるために 2 次 B スプラインを用いている。

いわゆる書体を表現するための肉付け方式は、ゴシック体と、明朝体の 2 種類を用意している。ゴシック体は、骨格に対して単純に等幅の線を付加しているだけである。明朝体は、通常の明朝体よりも簡略化した装飾を施している（図 4）。一般の商業印刷物で用いるようなフォントのデザインでは、ゴシック体、明朝体ともに複雑なデザインが必要であり、さらに、錯視などを考慮した細かい調整が必要とされているが、KAGE システムで想定しているのは、300〜600dpi の 11 ポイント本文における印刷用途であるため、解像度としては 100 ドット程度となり、それほど複雑な肉付けを施しても結果的には意味をなさないと考えている。そのためエンジンから出力されるグリフは 200 × 200 メッシュの解像度で表現されている。ただし、エンジン自体は任意の解像度に対応しているため、後述する輪郭ベクトル形式での出力では 1000 × 1000 メッシュの高解像度による出力を行っている。

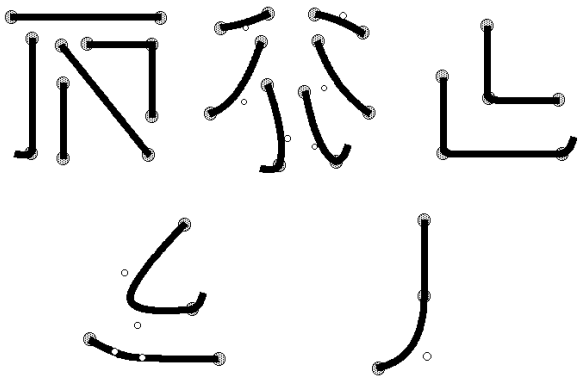

図 2: 応用ストローク 5 種（左上から直線、曲線、かぎ、複雑曲線、縦払い

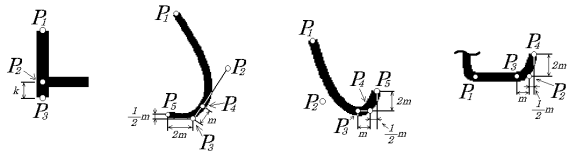

図 3: 筆末の自動処理（縦棒のトメ、左撥ね、右撥ね、かぎ撥ね）

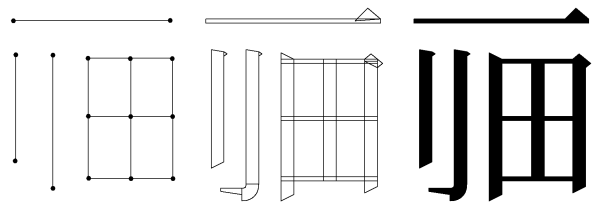

図 4: 明朝体風の肉付け

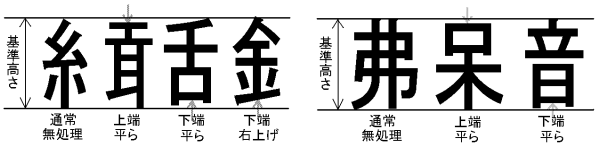

図 5: 部品調整処理（左部品、右部品）

グリフ生成のルール

漢字部品の配置や大きさの決定は、エンジンが自動的に行っている。ユーザの要求を動的に処理し、瞬時にグリフを配信する必要があるため、ルールは非常に単純なものとしている。具体的には、部品を X 軸（左右結合の場合）、Y 軸（上下結合の場合）それぞれの方向にスキャンし、筆画の線分をまたぐ回数を数える。ここで部品ごとの最大本数を部品複雑係数とし、係数比を基準に部品ごとの大きさを計算する。例えば左右 2 部品結合では以下の計算式を用いている。

$$ratio = \frac{(x_l \times 0.7^{b_{lL}+b_{lR}})^{0.6}}{(x_r \times 0.7^{b_{rL}+b_{rR}})^{0.6} \times 1.05}$$

（l:左部品, r:右部品, L:左端, R:右端, x:部品複雑係数（X 軸）, b:平ら（箱型）ならば 1, もしくは 0）

この計算式によってある程度の自動化が可能となっているが、多用される部品に関しては、複雑係数を手動で付与することで、より品質の高い出力を目指している。具体的には常用漢字の集合をサンプルに、生成されたグリフの中で違和感の大きい字に関して手作業による調整を施している。計算された数値に対して、さらに数種類の例外ルールを適用して大きさを調整し、その後部品の形状に応じて位置を決定してグリフが生成される（図 5）。

グリフのネットワーク配信

KAGE システムの目的は、集合に含まれない文字を利用することであり、さらに用いた文字の情報交換が保証される必要があったが、情報交換の保証の手段として、ネットワークを利用してグリフを共有することとし、サーバ＆クライアント型のシステムを採用する。そこでシステム実現のための第一段階として、ネットワーク上で文字を共有するためにグリフ配信サーバを実装した。サーバは拡張 IDS による文字の記述、書体、データ形式が入力されると、動的にグリフを生成し、クライアントに配信する。通信プロトコルは HTTP を利用する。

本来の IDS 表記は UCS のコードポイントを利用するが、その符号化は UTF-8 や UTF-16 など複数存在する。また HTTP プロトコルを利用するときに URL エンコーディングで表記することもあり表記法が一定


泳
=⿰氵永
=u2ff0.u6c35.u6c38

図 6: 上段：元となる字 中段：本来の IDS 下段：KAGE 拡張 IDS

図 7: ブラウザでの呼び出し

しない。そこで KAGE システムでは UCS のコードポイントを `uHHHH`（+verb+H+は 16 進数の数値）の形式で表現し、ASCII コード内に収まるように変換して記述する（図 6）。そして部品と部品を “.” で区切る。この方式をとることにより、IDS 表記に UCS コード以外の部品も記述できるようになる。

配信するグリフは、画像データとベクトル輪郭データの 2 種類の形式の出力が可能である。画像データ形式では、PNG (Portable Network Graphics) ファイルとして配信され、例えば HTML 文書内の `img` タグの `href` 属性からサーバに IDS 記述を送信し、配信された画像を文字として利用することが可能となっている。つまり、グリフは 1 つの画像ファイルとして返信されることになる（図 7）。ベクトル輪郭データ形式では、SVG (Scalable Vector Graphics) ファイルおよび EPS (Encapsulated PostScript) ファイルとして配信される。現状ではベクトル輪郭データ形式の具体的な利用方法は想定されていないが、将来的には、動的外字切替によるアプリケーション内でのコード外漢字の利用システムへのグリフ配信を考えている。

サーバプログラムは、当初 HTTP サーバ機能ごと実装していたが、エラー処理やスケーラビリティに問題が生じたため、現在は一般の HTTP サーバ（例えば Apache HTTP Server）の実行型 CGI プログラムとしてサービスを行う。つまり、HTTP サーバから呼び出されると、IDS 記述が埋め込まれた環境変数を読み込み、グリフを生成して、標準出力にデータを

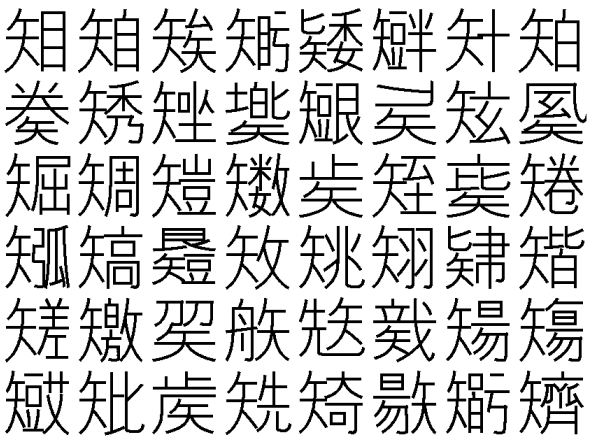

図 8: 生成されたグリフの例

書き出すと、HTTP サーバがクライアントに送信するという非常に単純な仕組みを利用している。

3.2 グリフエディタ

デザインの調整と部品の新規作成

2001 年にエンジン実装の第一段階が完了したが、生成した漢字グリフの文字品質は、ただちに利用できるような実用的レベルまでは達しなかった。部品が小さくなったときに生じるひずみの調整や、部品の大きさ・配置を決定するルールに改善の余地が見られる（図 8）。

漢字グリフ生成エンジンを実用的なものとするためには、印刷・出版向けの高解像度フォントや、市販されている一般的なフォントの品質までは必要ないが、少なくとも一般のユーザが見たときに問題とならない文字品質水準に達する必要がある。そこで、KAGE システムでは、グリフ生成ルールの見直しだけでなく、生成グリフのデザインをユーザが任意に調整できるエディタツールを用意することにした。

このエディタは市販されているフォントエディタのように輪郭を編集するのではなく、部品の大きさ、位置や、部品同士が接触する場合の末端形状の変更や部品内の筆画の調整を行うもので、フォントデザインの経験がないユーザでも容易に作業できるものを目指している。現在エディタは実装中であるが、テキスト形式で表現されている部品データを直接編集することも可能であり、また GUI を利用してマウス操作やボタンクリックでデザインすることも可能である。

エディタは Web ブラウザ上で利用することを想定していて、JavaScript 版（図 9）と Macromedia Flash を利用した Flash 版を実装中である。

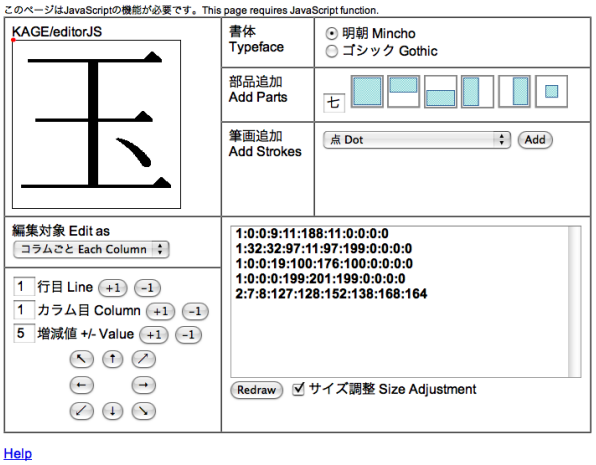

図 9: グリフエディタ (JavaScript 版)

漢字部品データベースへの登録

ユーザによって編集されたグリフは、後述する漢字部品データベースに登録することができる。IDS 表現から生成された自動生成グリフを調整した場合は、IDS 表現を部品の名前とするレコードとして登録されるため、今後同じ IDS 表現のグリフが要求された場合は、部品として人の手によって調整されたグリフが出力される。また、新しい部品をデザインして、データベースに登録することにより、その部品を IDS 記述内で呼び出して利用できるようになる。このようにエディタを活用することにより、より高品質で多種類の漢字グリフが生成可能となる。

3.3 漢字部品データベース

初期の KAGE エンジンでは漢字部品を内部に保有していたが、これをデータベースファイルとしてエンジンから独立させた。これにより、データの更新や追加が随時可能となった。データベースファイルへの操作は、Wiki[4] と呼ばれるコラボレーションツールを改造した Web フロントエンド上で行う。データベースの中には、以下のデータが格納されている。

単純部品データ 「土」、「言」、「人」などの漢字部品データで、IDS 記述に用いる部品で、基本的に UCS の部分集合となる。明朝体とゴシック体を別々に定義することができ、それぞれの書体で異なる骨組みの文字を利用できる（例：サンズイの 3 画目）。反対に明朝体とゴシック体が同じ骨組みの文字部品の場合は、部品データを共有することが可能である（例：口）。

```
1:0:0:0:48:201:48:0:0:0:0
2:0:7:114:0:115:167:4:198:0:0
2:7:0:53:67:83:170:192:194:0:0
```

筆種	形状	座標 1	座標 2	座標 3	座標 4
1	0,0	0,48	201,48	0,0	0,0
2	0,7	114,0	115,167	4,198	0,0
2	7,0	53,67	83,170	192,194	0,0

表 1: 「丈」の部品データ

特殊部品データ UCS コードに存在しない部品データで、IDS 記述に用いることが可能である。例えば「gt-12345」（日本学術振興会が配布している GT フォントの 12345 番の字形)、「pd-0001」（ユーザー私用定義部品 0001 番）のように、予約コードを除いたあらゆる名前の部品を定義することができる柔軟性を持っている。

リンクデータ 例えば「吉」という部品を定義するのではなく、「吉＝士＋口」という組合せデータを保持することでデータ量を減らすことが可能になる。このように、部品が定義されていなくてもリンクデータに対象となる漢字の IDS 記述が用意されていれば、IDS からグリフを生成して部品として利用するために定義されている。このデータは、著者オリジナルのデータと CHISE プロジェクトの成果物である漢字構造情報データベースをマージしたものである。

複合部品データ 先述したグリフエディタで調整したグリフは、IDS 記述を部品名としてデータベースに登録される。KAGE エンジンは IDS 記述からグリフを生成する際に、まず部品名としてデータベースに登録されているかをチェックする。

部品情報 現状では活用されていないが、部品データの検索用に付与するメタデータ用に確保されている。

各データは、全てテキスト形式で記述されている（図 10 および表 1,2）ため、専用のツールを利用する必要がなく、扱いが容易となっている。また、データベースが動的に更新されるという特徴から、グリフ生成エンジンから出力されるグリフが時間と共に変化することになり、場合によってはデメリットとなる。そこで、将来的に時間を指定することによってその時間に出力されるはずであるグリフを出力できる機能を追加するために、過去の全てのデータを保持している。

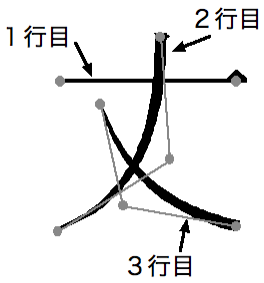

図 10: 「丈」のグリフ

筆種	内容
1	直線（2 座標:始点, 終点）
2	曲線（3 座標:始点, 制御点, 終点）
3	曲げ（3 座標:始点, 経由点, 終点）
6	4 点曲線（4 座標:始点, 制御点 1 , 2 , 終点）
7	縦払い（3 座標:始点, 経由点, 制御点, 終点）

形状	内容
0	開放
2	他筆画への接続
4	左撥ね上げ
5	右撥ね上げ
7	細入り、細終わり
8	止め
12	箱型左上角
13	箱型左下角
22	箱型右上角
23	箱型右下角

表 2: 部品データの詳細（一部）

4 オープン指向の開発

4.1 CHISE プロジェクトとの連携

現在の KAGE システムでは、グリフをネットワークで配信する形態をとっているが、クライアントの OS やアプリケーションそのものを拡張するわけではないため、現状では HTML 文書内での利用のみを想定していて、あまり実用的ではない。

このため、2002 年の秋から CHISE プロジェクトの文字合成サブプロジェクトへ参加している。ここでは、KAGE エンジンが CHISE モデルの任意の文字オブジェクトに付加された IDS 情報をもとにグリフを生成して具視化する機能を提供することを目指している。

現在 CHISE プロジェクトで用いている IDS 表記では指定できる漢字部品として ISO/IEC 10646 に含ま

れる文字だけでなく、CDP データベースで利用されている文字や、GT フォントの文字も含まれている。さらに重なりの構造をあらわす IDC の U+2FFB の利用も認められている。この CHISE 独自の IDS 表記を KAGE エンジンで処理するために、漢字部品データベースの部品命名規則の柔軟性が活用されている。具体的には、U+2FFB を含む IDS を丸ごと 1 つの部品として用意することである程度の回避が可能であると考えている。また CDP 部品や GT 部品は直接部品として定義することが可能である。

KAGE エンジンの実質的な開発者は著者一人であるが、エンジンの開発ソースは CHISE プロジェクトの一部として CVS レポジトリに登録され、CVS 環境で実装が継続されているため、誰でもソースを入手することが可能である。

4.2 生成グリフのライセンス

KAGE システムの目的は、生成された任意の漢字グリフを複数の人が共有する漢字処理モデルの実現である。このため、グリフの利用がライセンスによって制限されることは望ましくない。KAGE エンジンが出力するグリフの元となる漢字部品データベースは、Web ブラウザから自由にアクセスできる形態となっているが、認証せずにあらゆるデータが登録できるようになっていると、悪意の有無にかかわらず、第三者によるライセンス違反の可能性を持ったデータが混入し、KAGE エンジンの出力グリフが汚染されてしまう。そこで漢字部品データベースへのデータ登録は、データが完全にオリジナルであり、かつ自由な利用を認める方針を許容するユーザにのみ許可されるという制限が課せられている。この制限によって、KAGE エンジンから出力されるグリフは、完全なる自由利用が保証されることになる。

5 今後の課題

5.1 漢字部品の拡充

KAGE エンジンそのものの実装は概ね完了したが、漢字部品の不足により、現状では実用段階には達していない。そこでグリフエディタを活用した、複数人による部品の共同デザインが可能かどうかの実験を行う。具体的には、複数人による作業の際の、デザインの揺れ（不統一感）がどの程度影響してくるのか、また、それを解決することは可能かどうかを調査する予定である。

5.2 エンジンのライブラリ化

KAGE システムを OS やアプリケーション内部に組み込むためには、現状のクライアント&サーバ方式では手続きが繁雑となる。このため、エンジン部分を直接呼び出すことのできるライブラリとして別に実装することを考えている。現状では、随時更新される漢字部品データベースの更新処理をどうするかが課題となっている。

5.3 肉付け部分の独立

肉付け（書体修飾）アルゴリズムは、現在ゴシック体と明朝体の 2 種類が用意されているが、非常に簡略化されたものとなっている。このアルゴリズムをエンジンから独立させ、自由に記述できるようにすることで、より実用的なゴシック体、明朝体を再現することや、楷書体などの新たな書体の生成へと発展させることが可能であると考えている。

5.4 フォントのパッケージング

漢字部品の拡充を行うことによって、表現（出力）できる漢字集合は多くなる。主たる目的では無いが、JIS X 0208 や ISO/IEC 10646 の符号化文字集合だけでなく、JIS X 0213、Big5 や中国 GB 規格などの漢字集合を、グリフとして出力し、フォントファイルにまとめて配布できないかと考えている。

5.5 ユーザによる調整情報の知識化

構想の段階であるが、漢字部品データベースに登録されている、ユーザが施したデザインの調整を分析し、グリフ生成ルールそのものを再構築する機構を考えている。第一段階としては、部品複雑係数に対する各部品の適応度を計算し、誤差の大きい部品に対して補正が施されるようにする。また、ユーザの調整だけでなく、既存のフォントの部品配置情報をデータ化し、配置ルールを抽出して KAGE エンジンに適用することも可能であると思われる。

6 おわりに

KAGE システムによる多漢字具視化環境の構築に向けた現状と展望について紹介した。

将来的には、著者の本来の目的であった漢字グリフ自動生成の文字品質の向上に向けて、生成ルールのアイデアを簡単に検証できるためのツールキット的な機構を用意することで、漢字の字形に対する知識を蓄積し、共有財産として活用できる形にしたいと考えている。

参考文献

[1] CHISE プロジェクト. http://www.kanji.zinbun.kyoto-u.ac.jp/projects/chise/

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

[3] The Unicode Consortium. *The Unicode Standard, Version 3.0* , February 2000.

[4] Wiki. http://c2.com/cgi/wiki

[5] 上地宏一.「漢字フォント自動生成サーバ “影KAGE” の構築 －文字コードの枠組みを越える次世代漢字処理の提案－」, (漢字文献情報処理研究 第３号, pp.4-13, 漢字文献情報処理研究会, 好文出版), 2002.

[6] 田中一男、乙田清次、岡田守.「文字フォントの性質に着目した任意倍率文字サイズ変換法」(電子情報通信学会論文誌 J69-D No.3 pp.460-466), 1986.

[7] 田中哲朗、石井裕一郎、竹内幹雄、和田英一.「プログラム肉付けによる複数漢字書体間のスケルトンデータの共有」(情報処理学会論文誌 Vol.36 No.1 pp.177-186), 1995.

[8] 田中哲朗、岩崎英哉、長橋賢児、和田英一.「部品合成による漢字スケルトンフォントの作成」(情報処理学会論文誌 Vol.36 No.9 pp.2122-2131), 1995.

KAGE - An Automatic Glyph Generating Engine For Large Character Code Set

K. Kamichi

Abstract

This paper discribes the system which generates glyphs of Kanji Character by combination of some parts of Kanji on the fly, calls KAGE. The purpose of KAGE system is specific visualization of non-impremented glyphs for enabling information processing of Kanji which are not assigned by existing coded character sets. Positions and sizes of parts are calcurated by some rules which are aim at characteristics of its shapes and structures of target character.

There is a glyph designing tool for adjustment of automatic generated glyphs and for creation of new parts. The result of adjustment can regist to shared database, and can reuse for glyph generation.