

ネットワーク・コラボレーションによる 大規模フォントコンテンツの移植と応用

～情報基盤整備の試み～

谷田貝 常夫
文字鏡研究会

谷本 玲大
東洋大学大学院

文字鏡研究会は JIS や ISO 規格に未収録の文字を収集・整理してフォントセットと入力ツールを開発し、無償で公開している。文字鏡フォントは現在、約 90000 字を有し、その内容は国字や俗字のみならず、変体仮名、梵字、甲骨文字など多岐にわたる文字を収録する。これらを用いれば、歴史的文献の本文や書誌情報の記述も可能になる。これらの収録・公開作業は、各分野の研究者が集い、それぞれ持っている専門知識を活かす事に支えられて成り立っている。

1999年2月以降、Windows用 TrueType Font を Macintosh や TeX などの諸形式に移植する作業がネットワークを通じた共同作業として行われ、各種の環境でフォントが使えるようになった。本発表では、その経緯を報告する。

キーワード：情報共有 / フォント / 共同開発

Implantation and the Application of large scaled Font

Contents in collaboration on Network

— an attempt to arrange Information Ground —

YATAGAI Tuneo
Mojikyo Institute

TANIMOTO Sachihiro
Graduate school of Toyo University

Mojikyo Institute is collecting and arranging the missing characters out of JIS or ISO, and is making the fonts, and is also developing input tools, and is releasing them on website free of charge.

Mojikyo Fonts set contains more than 90000 characters, and not only Japanese local and popular characters but Kana variants, Sanskrit letters and Oracle Bone texts are collected in the set. Using these Mojikyo character fonts of various ranges, the original texts of historical documents or philological information are enabled to describe as the present time data.

These collecting and releasing works have been supported by many scholars of various fields with their specialties.

On Feb. in 1999, the implanting works of TrueType fonts for Windows into Macintosh and TeX were done collaboratively through network. Ever since, Mojikyo fonts have been able to use in various configuration of platforms. We are reporting the course in this meeting.

Keywords: Information sharing / Fonts / Collaboration

- III. 各フォントに、文字を割り当てる位置は JIS 第一水準の区点位置 16 区～45 区まで(第一水準範囲内で 46、47 区には割り当てない)と、第二水準の 48 区～77 区(の区点位置 07794 番)まで。JIS の 60 区分の $94 \times 60 = 5640$ 字分。30 区ずつ二つのグループ。
- IV. Mojikyo M101 の Shift_JIS CodePoint 889F ～Mojikyo M109 の Shift_JIS CodePoint E371 までに、大漢和辞典[修訂版]の 1 番から 49964 番までの文字が順番通りに割当。(なお、所々空きになっているところは、大漢和の飛び番や、文字鏡側 で大漢和に重複していると認定したもの。)
- V. Mojikyo M109 の Shift_JIS CodePoint E3FB～Mojikyo M109 の Shift_JIS CodePoint E6C8 までが文字鏡番号 50101～50614 に当たり、これは諸橋『大漢和辞典 [修訂版]』で番号に[]または[]がついている文字に便宜上付けた番号。(なお、この付き番号の順列リストは文字鏡研究会の WebPage <http://www.mojikyo.gr.jp/> に掲出。)
- VI. Mojikyo M109 の Shift_JIS CodePoint E6C9 以降は『大漢和辞典』未収録漢字を入れている。これ以降のコードポイントと文字鏡番号には、規則性はほとんどない。

3 開発物の内容と開発の経緯

3-1 フォントの移植

1999/02/10、nifty の FTEX MES:10(#00313)にて小野康男氏(横浜国立大学教育人間科学部)、文字鏡フォント紹介。

1999/02/15、同会議室(#00318)にて稲垣淳氏、文字鏡の Windows 用 TrueType Font を LaTeX から使用するためのマクロ公開。

1999/02/24、同会議室(#00322)にて谷本、METAFONT ソース、PK などの諸形式作成を提案。以降、稲垣氏を中心に FTEX アクティヴ諸氏の協力を得てフォント移植の活動が会議室上、メールなどで活発に行われ、1999/04/01 までに一応の完成。それ以降も会議室上でメンテナンスが行われ、乙部巖己氏(東京大学大学院)氏の配慮により dviout for windows にも文字鏡向けの手当が行われた。

3-2 マクロパッケージ

1999/02/15、同会議室(#00318)にて稲垣淳氏、文字鏡の Windows 用 TrueType Font を LaTeX から使用するためのマクロ公開。

以降、稲垣氏の丁寧かつ頻繁なメンテナンスにより、文字鏡 TeX 用マクロ(MOTEXDOC.LZH)は Unicode タグ対応など、大幅に機能を拡張して、Ver2.3 に至る(1999/8/14 現在)。

なお、文字鏡 TeX 用マクロは、コマンドなどの詳細な使い方はマクロそのものの作成・メンテナンスは稲垣氏(著作権も当然、稲垣氏にある。)が行い、全般的な README やアーカイブのパッケージングは文字鏡研究会が行っている。

4 Macintosh 用 Font

1999/01/24、弥永信美氏、TrueType font の Windows <-> Mac コンバータ True Keys による Font コンバートに成功。

1999/03/08、弥永信美氏、Mac 用の入力用マクロテーブルβ版完成。

1999/03/21、弥永信美氏、Mac 用の入力用マクロテーブル完成。

5 TeX 用 Font

5-1 文字鏡フォント分割概要

上記フォントを、1バイト圏の OS やソフトウェアで扱えるようにコンバートするには、フォントのコード割り当て位置を2バイトから1バイトに変更する必要がある。そこで、オリジナルの Windows 用の TrueType Font フォントを16区の開始位置から順番に256文字ずつ区切って分割した。

JIS の KUTEN 番号16区(KU)~45区までと48区~77区までをこのように分割すると、一つのオリジナルファイルが、24個に分割される。

16区の先頭から256文字分のファイルの「フォント名称」と「ファイル名」には枝番号"06"を付け、その直後の256文字分には枝番号"07"を、以下、256文字ずつに"08"、"09"..."29"を付ける。

文字鏡のオリジナルフォントは、101~121と181~183の合計24個だが、コードポイントの割り当て位置に移動予定がある181~183は除外してコンバートした。よって、101~121の合計21個のファイルをそれぞれ分割することになり、21*24=504個のファイルが出来る。

5-2 文字鏡のフォント名の規則性

ここで言う「フォント名」とはファイル名ではなく、ソフトウェア側に認識されるフォントの名称を指す。フォント名は以下のように付けた。

MOJikyoMINchoHWxxxyy-Regularzz

- I. HWはHalfWideの意。
 - II. xxxはオリジナルのフォントについている番号。
 - III. yyは、xxxを分割した時の分割ナンバ。
 - IV. zzは、ファイル分割時にコンバータが自動的に付けた番号で、06~29までである。
- 例:

元の Windows TTF ファイル名 MOJIK101.TTF

変換先名ファイル名 mo101m06.pfb

変換先名フォント名 **MOJikyoMINchoHW-BABAG-Regular06**

フォント名称中の数字は、ATM ではフォント名の違いとして認識されるには 優先順位が低く、小文字のローマ字も同様に低いので、文字鏡のフォントを確実にユニークなフォント名称として認識させるため、オリジナルフォントの番号を 0123456789→ABCDEFGHIJ として割当。よって、BABAG は 10106 の意味となる。

5-3 文字鏡フォントのコンバート先フォーマット

まず、オリジナルの Windows 用の TrueType Font フォントから、PS フォントの ソースを作成し (この PS ソースは配布していない。)、その PS ソースファイルから、各種変換 Tool を用いて、以下のフォーマットのファイルを生成。

- I. **mt1、bma** (アーカイヴのファイル名は"MO???MT1.LZH")
mt1 は Mac Type1 Printer Font で、MAC 用の PSfont。
bma は Mac Screen Font で、メトリック情報を持つビットマップ font。
Mac 上の ATM で使用する時にこの二つを組み合わせて使用する。
- II. **pfm、inf**(アーカイヴのファイル名は"MO???INF.LZH")
ASCII 形式で拡張情報が書かれた inf ファイルと併せて使うことにより、pfm ファイルが生成可能。
pfm はプリンタ用のバイナリ形式のメトリック情報ファイルで afm ファイルと内容的には同じ。Windows 上 (PC 上) の ATM で使用する時には pfb、pfm が必要。
- III. **pfb、afm**(アーカイヴのファイル名は"MO???PFB.LZH")
pfb は TeX と DVI ware が使用するバイナリの PC 用の Postscript Type1 形式 font で、font outline data を含む。
afm は ASCII 形式の Adobe Font Metric 情報ファイルで、font 名、文字幅情報、カーニング情報などを含む。Mac では使用しない。
- IV. **tfm、map**(アーカイヴのファイル名は"MO_TFM.LZH")
tfm は、1 バイト font の TeX 用メトリック情報ファイル。(JFM ファイルとは異なる。)プレビューやプリントに実際どのような形式の font を利用できるかは DVI ware に依存。
map は、tfm ファイル名と PostScript 上の font 名の組み合わせを定義したファイルで、TeX と DVI ware が使用。
- V. **mf、PS2MFBAS.LZH**(アーカイヴのファイル名は"MO???MFLZH")
METAFONT ソース。

6 TeX 用文字鏡フォント作成手順

Windows TrueType Font → *.pfb、*.afm、*.tfm、*.mf、*.pfm、*.inf 変換バッチシステム。
稲垣氏の教示により作成された。IBM PC 互換機上の Windows95/98/NT のいずれかの環境で動作する。

6-1 ソフトウェア環境

- I. TEdit : シェアウェアの TTF 作成ソフト
旧バージョン ver1.2 あたりの SJIS コード仕様のバージョンが必要。
- II. Ghostscript : フリーの PS 互換ソフト (日本語対応 gswin 5.50 以降のバージョン)
<ftp://ftp.u-aizu.ac.jp/pub/tex/ptex-win32/> (gs550-j-wapi.zip)
<ftp://ftp.riken.go.jp/pub/tex-archive/support/ghostscript/aladdin/gswin550/>
- III. PS4MF : Postscript Type1 フォントをメタフォントに変換するためのツール
- IV. <ftp://ftp.u-aizu.ac.jp/pub/tex/CTAN/fonts/utilities/ps4mf/ps4mf28.zip>
- V. rsx.exe : メモリーマネージャ
- VI. <ftp://ftp.u-aizu.ac.jp/pub/tex/CTAN/systems/msdos/emtex/emxrsx.zip>
- VII. TTF2PFB : CJK の TrueType Font を欧文 Postscript のフォントソースに変換する
- VIII. <ftp://ftp.u-aizu.ac.jp/pub/tex/CTAN/fonts/utilities/ttf2pfb/ttf2pfb.zip>

6-2 作業手順

TEdit で、Mojik101.ttf、Mojik102.ttf…をコピー元に、コピー先を新規作成にして数字名称 101.ttf、102.ttf…という新規和文 TTF で一括コピー。

できた TTF を作業フォルダに置く。

作成用 bat ファイルを実行。

実行には Ghostscript、t1asm.exe、afm2tmf.exe、refont.exe、ps2mf.exe、rsx.exe、ttf2pfb.exe と、gawk が必要。他に、独自の*.cfg ファイルを使用する。

bat ファイル中、*.01 から*.05 と、*.30 から*.33 は文字鏡の TTF の場合は文字が割当られている領域の関係上、不要。他の TTF 用では必要。

なお、*.mf は内部から PS4MFLZH に含まれる ps2mfbase.mf ファイルを呼ぶので、*.mf を配布するのであれば、必ずこれをひとつ同梱しておく必要がある。

なお、作成用 bat ファイルの内容は以下の手順を踏んでいる。

- I. ttf2pfb で ps ソースファイルを作成。
- II. t1asm で pfb ファイルを作成。
- III. Ghostscript で AFM ファイルを作成。
- IV. ps2mf で メタフォントソースを作成。
- V. afm2tfm で TeX が使う tfm と、dvips が使う map ファイルを作成。
- VI. refont で pfm ファイルを作成。
- VII. refont を /I オプションで動かして inf ファイルを作成。
- VIII. compmap.awk で フォントの埋め込み指定をするよう map ファイルを書き換え。

【Ghostscript での表示例】

'20x										"8x
'21x										
'22x										"9x
'23x										
'24x										"Ax
'25x										
'26x										"Bx
'27x										
'30x		一	丁	彡	ㄥ	𠂇	𠂈	七	丄	"Cx
'31x	丁	彡	万	丈	三	上	下	卅		
'32x	卍	𠂇	𠂈	不	与	𠂉	𠂊	丑		"Dx
'33x	且	𠂋	丈	𠂌	𠂍	且	𠂎	世		
'34x	世	𠂏	𠂐	𠂑	𠂒	𠂓	𠂔	𠂕		"Ex
'35x	丞	𠂖	𠂗	𠂘	𠂙	𠂚	𠂛	𠂜		
'36x	𠂝	𠂞	𠂟	𠂠	𠂡	𠂢	𠂣	𠂤		"Fx
'37x	𠂥	𠂦	𠂧	𠂨	𠂩	𠂪	𠂫	𠂬		
	"8	"9	"A	"B	"C	"D	"E	"F		

7 MOJIKYO パッケージ

稲垣淳氏の作成による LaTeX2e で今昔文字鏡フォントを使用するためのパッケージ。これらのマクロファイルは主に nifty の FTEX を拠点にして議論と開発・改修が進められている。本田博通氏のページ (<http://member.nifty.ne.jp/kansha/tips/mojikyo.html>) からは Vine Linux での VFlib と、それに対応した dvipsk, xdvi, ghostscript などによる Windows 用 TrueType Font と mojikyo.sty の使用方法に関する情報を得ることができる。

8 謝辞

本発表及び本発表の内容である成果物は、全面的に稲垣淳氏及び FTEX アクティブ諸氏の技術供与・指導・助言・助力・協力に拠っており、本来ならば、発表者として稲垣氏が中心になって行うべき発表であることをここに特記する。
稲垣氏の無私の親切に深謝申し上げる次第である。

次ページは、稲垣氏作成の「Mojikyo.sty バージョン 2.3 コマンドの使い方と使用例」よりの抜粋である。リストに挙げられているようなコマンドを用いて文字鏡フォントを使用することができる。

mojikyō.sty ver 2.3 コマンドリファレンス

<code>\boldwidth</code>	<code>\futome</code> を指定したときや、ボールド指定のときの太らせ程度のパラメータ。デフォルト値は、0.02ex。あまり大きくするとゴーストのようになるので注意。
<code>\CharacterMap</code>	文字鏡の文字の一覧表を作る。 <code>\CharacterMap</code> [開始文字番号]{終了文字番号+1}のように使う。開始番号はオプションで、無指定時は、000001 番からになる。
<code>\enablebold</code>	ボールド指定のときに文字を太くするか、否かを決めるフラグ。デフォルトは true。文字が潰れてしまうときには、 <code>\enableboldfalse</code> とすれば、以降の文字は太くなくなる。 <code>\enableboldtrue</code> とすれば、ふたたび太字になる。
<code>\futome</code>	文字を少し太らせる。文字鏡の文字の細さが気になる時につかう。 例：補正なしの太さ <code>\PMO{047268}</code> → 鷗, 補正あり <code>\futome{\PMO{047268}}</code> → 鷗, ボールド指定の中で文字鏡の文字が使われたときは、自動的に少し太めにする。明示的に <code>\futome</code> で指定するよりもやや太くなる。
<code>\hash</code>	ハッシュ (#) を出力するコマンド。 例： <code>\hash</code> → #
<code>\T(P)MO</code>	文字鏡の文字を文字鏡の 6 桁番号で指定する。 <code>\TMO</code> , <code>\PMO</code> は、使用フォントフォーマットを強制する。 <code>\usePostScript(true or false)</code> を指定して <code>\M??????</code> ; で指定することもできる。 例：森 <code>\PMO{058562}</code> 外 → 森鷗外
<code>\usePostScript</code>	<code>\usePostScripttrue</code> とすると、 <code>\TMO</code> 以外のコマンド・環境で 1 バイトフォントを使う。逆に、 <code>\usePostScriptfalse</code> とすると、 <code>\PMO</code> 以外のコマンドで 2 バイトフォントを使う。デフォルトは true
XMLtag 環境	文字鏡の文字を XML タグ風の表記方法で指定する。 例： <code>\begin{XMLtag} &M047268;</code> → 鷗 <code>\end{XMLtag}</code>

Unicode フォントを使用するためのコマンド (フォントの入手とコンバートが必要)

<code>\UniD</code>	Unicode 文字を十進数で指定する。 <code>\#?????</code> ; という指定の仕方も可能。 例： <code>\UniD{40407}</code> → 鷗 XMLtag 環境の中では、 <code>&#?????</code> ; (? は、Unicode の十進表記、 <code>\UniD</code> の引数と同内容) で指定することもできる。
<code>\UniH</code>	Unicode 文字を十六進数で指定する。 例： <code>\UniH{9dd7}</code> → 鷗
