Ligatures & informatique *

Jacques André

Irisa/Inria-Rennes
Campus universitaire de Beaulieu
F–35042 Rennes cedex, France
email: jandre@irisa.fr

Résumé. En typographie numérique, une ligature correspond au remplacement de n glyphes successifs par un seul. Les problèmes sont alors des problèmes de saisie : comment indiquer que l'on veut remplacer, par exemple, deux lettres comme c et t par la ligature ct ? Ou plutôt, comment savoir qu'une fonte o re cette ligature ct ? Ou encore mieux, comment faire en sorte que l'utilisateur n'ait pas à se soucier de ce problème? Cette note fait le point sur ce problème du codage des ligatures, que nous ravalons ainsi au niveau d'un problème de métrique de fontes !
Au mot ligature est aussi associée la notion de lien entre lettres manuscrites. Nous donnons alors quelques références sur les problèmes de simulation ou de reconnaissance de l'écriture manuscrite.

Abstract. *In digital typography a ligature consists in replacing n glyphs by a single glyph. This raises problems at keyboard level : how can two letters such as c and t be specified to be replaced by the ct ligature? How does the user know if the font in current use contains the necessary ligature? Is it possible to design a system in which the user does not have to deal with such problems? The present paper looks at such problems from the point of view of coding and font metrics.*
The final section provides a brief introduction to script ligatures and the related problems of handwriting simulation and recognition.

Dans son *Petit traité de la ligature* [38], Jérôme Peignot écrit : «Évidemment, [...], faudrait-il imaginer un processus tel que le rapprochement de deux lettres à ligaturer déclencherait le dessin du lien le mieux habilité à les réunir. Sans doute, ne serait-ce que pour des raisons financières, tout ceci est-il du domaine du rêve. Et si, à son tour, cet investisement n'était rien en regard de la perte qu'impliquerait un surcroît de désa ection d'avec la lecture ? »

* Cette note correspond à des exposés faits dans divers contextes, notamment lors de l'École Didot organisée à l'École Estienne à Paris en mai 1992. Elle a été partiellement financée par le projet CEE/Comett/Didot numéro 90/3697Cb. Une première version a été di usée sous forme d'un rapport de recherche [11] dont la partie correspondant au codage a été publiée dans un récent numéro des *Cahiers GUTenberg* [8] ; ce présent article est donc en quelque sorte la suite de ce dernier article.

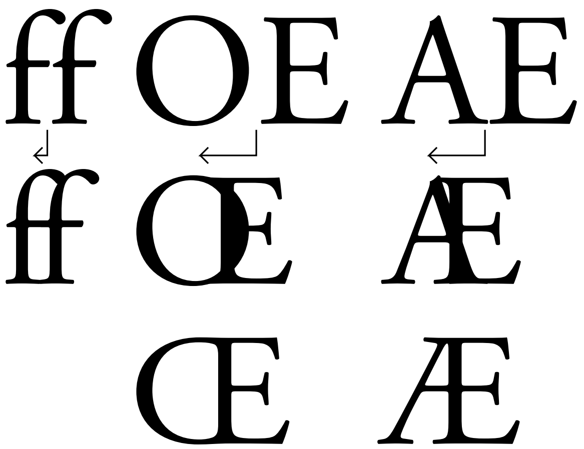

FIG. 1 – *Les ligatures ne peuvent pas toujours être obtenues par crénage. En haut : séquences normales de glyphes ; en dessous séquences de glyphes dont le second a été rapproché du premier (de la distance indiquée par la flèche) ; en bas : les glyphes désirés !*

Ainsi, dès 1987, Jérôme PEIGNOT fait-il allusion aux caractères contextuels garants d'une certaine qualité de lecture. Mais pour lui, c'est un problème d'argent. Il a tort et raison tout à la fois : ce n'est pas coûteux de faire une fonte contextuelle ou de dessiner des ligatures ; mais c'est un problème de sous, et même de très gros sous : c'est la raison des batailles – les media ont parlé de « guerre des polices » – entre Adobe, Apple et autres IBM pour les standards[1] de fontes, non pas de leur contenu mais de la façon de les présenter.

L'objet de cette note est de montrer ce qu'est une ligature pour un informaticien et pourquoi cette vision n'est pas obligatoirement agréable pour l'utilisateur, typographe, lecteur ou acheteur de fontes !

1. Informatique, caractères, glyphes et ligatures

Nous avons donné en introduction à ce *Cahier* [4, page 1], les divers sens généralement acceptés pour le mot ligature.

En informatique, ce sens est, raisonnablement, plus restreint. Mais les problèmes sont autres !

1. En donnant à ce mot son sens commercial et non celui normatif a ecté à « norme *de instituto* ». Voir [8].

1.1. Les ligatures informatiques ne sont pas faites par crénage

La photocomposition puis les fontes numérisées ont ouvert la voie à quelque chose que les typographes du plomb ne connaissaient pas : l'interlettrage (c'est-à-dire la réduction de l'approche entre tous les caractères d'une même ligne, notion que les typographes n'approuvent en général pas, voir section 4.3) et le crénage (cette notion s'appliquant d'avantage entre deux caractères spécifiques).

Même si c'est parfois utilisé, disons dès à présent que la typographie numérique ne confond pas ligatures et crénage, que ce n'est pas comme celà que les ligatures sont faites ! La figure 1 montre, ligne du haut, trois paires de caractères normalement imprimées puis, ligne du milieu, ces trois mêmes paires après avoir fait un crénage (c'est-à-dire un recul du second caractère). On n'obtient pas les «ligatures» escomptées de la ligne du bas : « » a des hauteurs de crosses di érentes pour le premier et le second f, «Œ» serait plutôt obtenu en crénant un «C» et un «E»[2] ; quant au «Æ» il serait di cile de l'obtenir par simple déplacement (ce qui est encore plus vrai du «ct» de la figure 2).

1.2. Caractères, glyphes et codage

Il convient d'abord de rappeler que, tout comme en typographie tarditionnelle, on distingue en matière de communication et d'informatique deux choses (voir [8]) :

1. La notion de **caractère** : un caractère est une unité d'information désignant un élément d'un alphabet (par exemple un «A» ou un «point virgule»).
2. La notion de **glyphe**[3] : c'est une des formes géométriques correspondant à un (ou plusieurs) caractères, utilisée pour imprimer sur papier ou a cher un texte sur écran.

Il n'y a pas de relation biunivoque entre ces deux notions : par exemple au caractère «A» peuvent correspondre plusieurs glyphes, tels que «A», «*A*», «A» voire «**A**» ; réciproquement, au glyphe «A» peuvent correspondre plusieurs caractères, comme «A latin majuscule» ou «Alpha grec majuscule».

Codage des caractères et des glyphes :

1. Un caractère est généralement codé (en ordinateur, ou sur les réseaux) en utilisant l'une des nombreuses normes d'échange, par exemple avec 7 bits comme dans Ascii, 8 bits dans IsoLatin1 ou 16 bits dans Unicode (le nombre de caractères disponibles étant respectivement d'environ 100, 220 ou 18 000 caractères) [8]. À chaque

2. Peignot fait d'ailleurs remarquer qu'autrefois on désignait «Œ» du nom de «E dans l'O» et que le glyphe correspondant était plutôt «Œ» [38].

3. Ce néologisme d'origine américaine correspond en gros à l'un des sens du mot œil en typographie française [21, page 483].

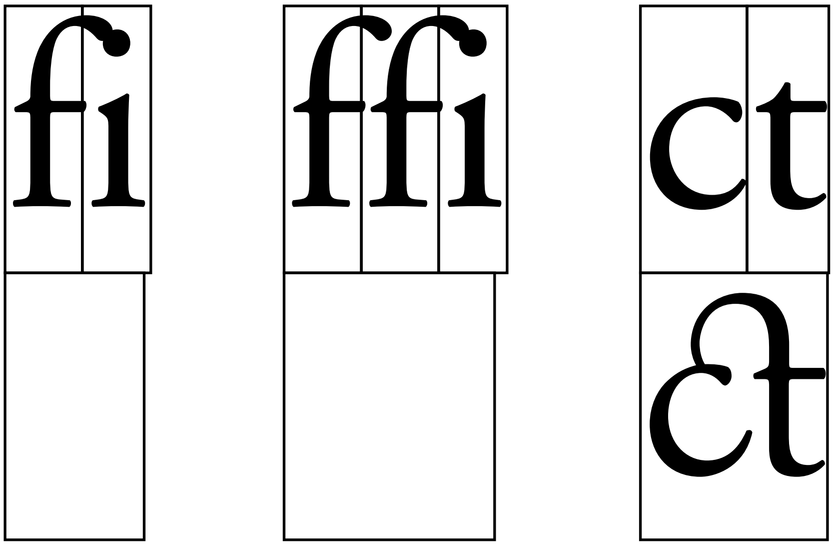

Fig. 2 – *En haut : séquence normale de caractères ; en dessous : les ligatures correspondantes.*

caractère est associé un numéro, par exemple, en Ascii, les caractères «A» et «;» ont respectivement pour codes 101_8 et 073_8.

2. À chaque glyphe (dans une imprimante ou dans un gestionnaire d'écran) est aussi associé un code, mais souvent différent de celui d'un caractère correspondant ! À la notion de glyphe correspond maintenant la notion de «caractère vectoriel», c'est-à-dire d'algorithme permettant de dessiner ce caractère au corps voulu, avec l'orientation voulue, dans la famille voulue, etc.[4]. Ces glyphes sont alors désignés soit par des noms (par exemple *semicolon* pour «; ») soit par des codes (par exemple pour Ibm-Ebcdic «A» est codé 193 tandisque pour Apple-QuickDraw c'est 65) mais le dessin effectivement produit dépend du contexte (fonte utilisée, corps, angle, etc.).

1.3. Définition de ligature en informatique

Plusieurs principes sont à la base de la définition des ligatures en informatique :

1. Puisqu'une ligature ne peut généralement pas s'obtenir par des moyens géométriques simples, force est donc de considérer qu'à chaque ligature, quelque soit le sens donné

4. On trouvera dans [3, 28, 31, 36] diverses introductions à la numérisation des caractères.

à ce mot, correspond un glyphe. La figure 2 montre, ligne du haut, la séquence de glyphes de «f» et «i», puis «f» «f» et «i», et enfin de «c» et «t» (les rectangles indiquent en gros l'emplacement des bords du caractère en plomb correspondant). La ligne du bas montre les glyphes «fi», « » et «ct». Tout se passe comme si on avait écrit (pour la colonne de gauche) :

```
ligne du haut:
    imprimer caractère de code 146   (ou de nom << f >>)
    imprimer caractère de code 151   (ou de nom << i >>)
ligne du bas :
    imprimer caractère de code 256   (ou de nom << fi >>)
```

2. Puisque, dans une famille de caractères, le choix des ligatures est décidé par son dessinateur, par exemple «ct» présent dans le *Garamond* n'a pas été retenu par Stanley Morison pour le *Times*[5] et puisque ces choix peuvent être très variés, pratiquement indécidables, il n'est pas possible de dresser un inventaire de toutes ces ligatures potentielles et il n'est donc pas possible de prévoir un codage absolu c'est-à-dire d'associer systématiquement au même glyphe le même numéro. Force sera donc «à l'ordinateur» (c'est-à-dire au formateur ou système d'impression) de détecter les ligatures, voire de créer les glyphes *ad hoc* dans certains contextes (langues non latines par exemple, voir [26, ce cahier page 87]).

3. En revanche, les abréviations usuelles (comme «#»), les symboles que nous avons appelés «*p*er se» comme «&», les ligatures linguistiques comme «œ», etc., faisant partie de la langue, sont considérés comme des caractères et les normes de codage leur attribuent[6] une place au même titre que le «;» ou le «A».

Compte-tenu de ces principes, on peut alors dire qu'en informatique,

Faire une ligature consiste à remplacer une suite de glyphes $G_1 G_2 \ldots G_n$ par un glyphe unique G_0, ce remplacement pouvant être systématique.

Remarques :

1. Par glyphe on entend aussi bien des traces imprimables que certaines traces sans graphie, comme par exemple le «début de mot» ou la «fin de mot» ce qui permet donc

5. Si on veut prendre un exemple plus contemporain : Robert Slimbach a décidé de garder pour l'*Adobe-Garamond* (que nous utilisons ici) la ligature *st* en italique mais pas celle du romain : «st» !

6. Ou devraient leur attribuer une place : nous faisons évidemment allusion ici au «œ» qui ne fait pas partie d'IsoLatin1, mais sûrement pas pour des raisons techniques [37] ; d'ailleurs cette «erreur» a bien été réparée dans Unicode ; voir [8].

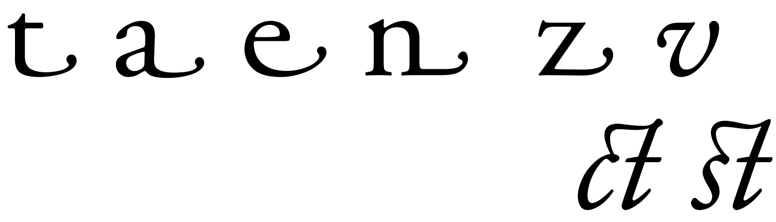

Fig. 3 – *Quelques-unes des lettres finales (en haut) et ligatures (en bas, en italique) d'*Adobe-Garamond.

de considérer comme ligatures les formes initiales ou finales (comme le « t final » du *Garamond*, figure page 2 de ce *Cahier*, comme les lettres finales de *Poetica* 6) ou comme certains caractères cambodgiens «inexistants» (voir [24, 26]). C'est probablement Donald Knuth [34] qui le premier a associé à cette notion celle de «caractère vide», l'équivalent en typographie du zéro en mathématiques. Cette notion apparaît maintenant dans les codes comme Unicode et Iso-10646, mais avec de légères di érences entre ces normes : codes FEFF *(Zero Width No Break-Space)* et FEFF *(Byte Order Mark)*.

2. Le principe de systématisation fait que l'on ne considère pas comme ligatures des glyphes utilisables automatiquement, comme «&» (on n'écrit plus aujourd'hui «&c.» ni « un ball& »), comme « @ » (on n'écrit ni « b@aud » ni « b@eau » selon que l'on considère que @ signifie *ad* ou *at*) ni comme «œ» (on n'écrit pas «cœ cient»).

3. Les norme d'échange comme Ascii, IsoLatin1, Unicode, etc. ne connaissent que les caractères, pas les glyphes [8]. On ne trouve donc pas de « ligatures » dans ces normes, ce qui ne veut pas dire que l'on ne sache pas imprimer en PAO «fi» ou «ct» (la preuve est que ce texte est composé en PAO !).

Deux problèmes sont alors à résoudre point de vue informatique :

1. de quelles ligatures dispose-t-on pour une famille de caractères donnée ?

2. comment quémander ces ligatures ?

FIG. 4 – *Quelques ligatures du Galahad dessiné par Alan Blackman pour la collection Adobe Originals*[7].

2. Inventaire des ligatures

Les normes d'échange, on l'a vu plus haut, ne connaissent pas les ligatures puisque, pour elles, ce sont des glyphes et non des caractères. Quant aux fontes elles-mêmes (c'est-à-dire les collections de glyphes), on peut distinguer plusieurs cas[8] :

1. Certaines fontes n'o rent pas de ligatures ; c'est par exemple le cas des vieilles *Chicago, New York, Geneva* d'Apple ; on ne peut pas en dire grand chose, sauf qu'elles sont un peu à l'origine de la mauvaise presse qu'a pu avoir la PAO aux yeux des typographes !

2. Beaucoup de fontes n'ont que quelques ligatures notamment les deux plus courantes : « fi » et « fl » ; c'est pratiquement le cas de toutes les fontes numérisées (comme le *Times*) probablement car ces deux ligatures font partie des tables de codage standard d'Adobe.

3. Quelques fontes en disposent de plus ; par exemple *Lucida* de Bigelow & Holmes ; pour pouvoir les utiliser, il faut alors les faire entrer dans le vecteur de codage (voir [6, 8]) soit à la place d'un autre caractère, soit à la place d'un code inutilisé par PostScript (c'est en gros ce qu'on fait avec (LA)TEX en utilisant le standard de Cork [8, table 9] où on voit que, par exemple, «   » est à la place du caractère de contrôle inutilisé « US » *(Unit Separator)* d'ASCII-ISOLATIN1).

4. On assiste depuis deux ou trois ans à l'apparition de familles de caractères o rant un très grand nombre de ligatures ou caractères spéciaux (lettres finales, ornées, etc.). On est alors très loin des seuls fi et   ! On relève, uniquement dans les figures 3, 4 et 5 de cet article, les combinaisons suivantes :

Ch, ch, ck, cl, ct, ff, ffi, ffl, fi, fj, fl, ft, gg, Hi, ll, sf, sp, ss, st, TH, Th, th, tt, OE, UE

7. Extrait de *Adobe Magazine*, 3, 1995, avec l'aimable autorisation de Bernard Weyer.

8. Jadis, la place des cassetins de ligatures variaient d'une casse à l'autre ; aujourd'hui, quand Adobe propose une « fonte expert », c'est un peu la même chose que lorsque Didot proposait sa « casse d'anglaise » (voir figure 4 page 21 de ce *Cahier*) ou Vincart sa casse spéciale (*idem*, page 28) !

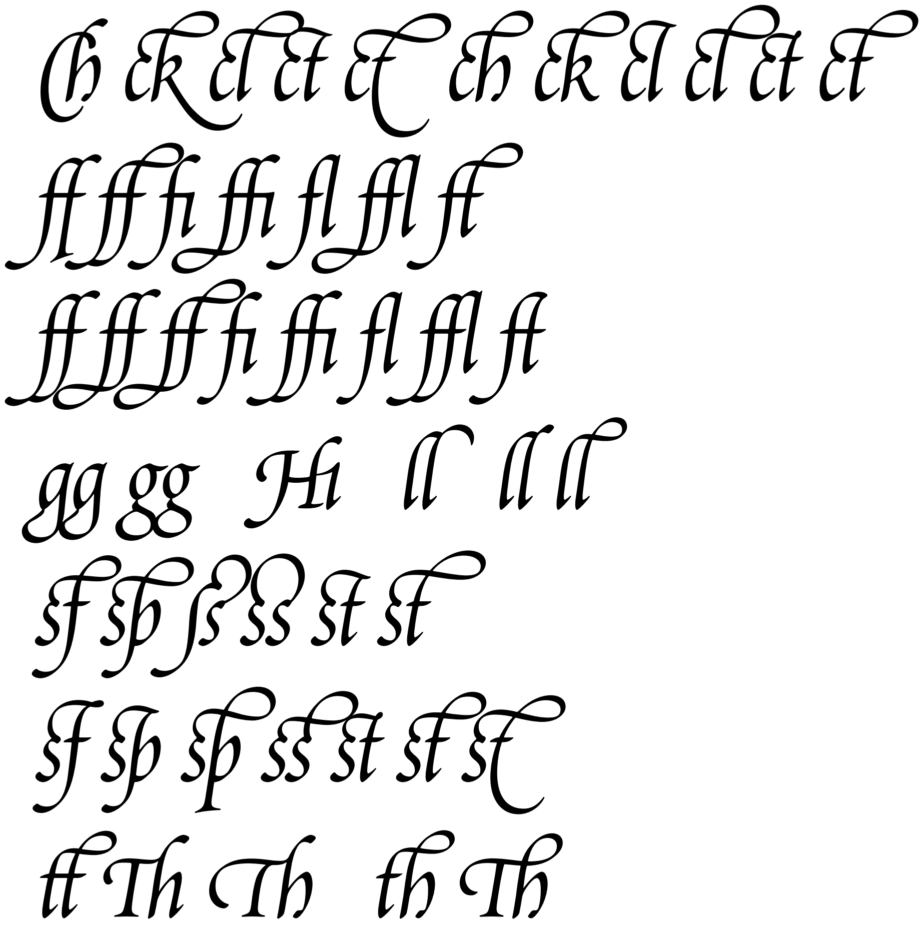

Fig. 5 – *Casse de la fonte Adobe Poetica Ligatures*

ChanceryI	*KLMNPQRWXYZ* *klnyz*
ChanceryIII	*KLMNPQRWXYZ*
RomanSmallCapsAlt	K K L M N N Q R R W X X Z
SuppSwashCapsI	*KLMNPQRWXYZ* *KLMNPQRWXYZ*
SuppSwashCapsII	*KLMNPQRWXYZ* *KLMNPQRWXYZ*
SuppSwashCapsIII	*KLMNPQRWXYZ* *KLMNPQRWXYZ*
SuppSwashCapsIV	*KLMNPQRWXYZ* *KLMNPQRWXYZ*
SuppInitialSwashCaps	*K L M N P Q R W X Y Z*
SuppLowercaseAltI	*k k l l y y z z*
SuppLowercaseAltII	*k k y y z*
SuppLowercaseBegI	*k k l l n n y y*
SuppLowercaseBegII	*k k l l n n y y*
SuppLowercaseEndI	*l l n n y z*
SuppLowercaseEndII	*l l n n*

FIG. 6 – *Variantes de glyphes en Adobe Poetica*

FIG. 7 – *Ces ligatures de l'*Avant-Garde *dessiné par Herb Lubalin ne font malheureusement pas partie de la fonte numérisée par Adobe.*

Ces caractères sont alors ventilés dans plusieurs fontes[9].

(a) *Adobe-Garamond* [1, 2, 14] o re beaucoup de ligatures traditionnelles (voir par exemple figure 3) mais nécessite pas moins de trois fontes pour les ligatures en romain droit :
   – «fi» est dans la fonte *Adobe Garamond Regular*,
   – « » dans la fonte *Adobe Garamond Expert Regular*, à la plce de Y,
   – «ct» dans la fonte *Adobe Garamond Alternate Regular*, à la place de c ;
   de même pour les italiques, italiques-gras, etc.

(b) Comme le précédent, *Poetica* est un caractère qui a été dessiné récemment par Robert SLIMBACH[10] ; la figure 5 montre quelques unes des ligatures disponibles. Celles-ci sont dans une seule fonte (mais il faut savoir qu'il faut taper «A» pour avoir la première ligature «Ch», B celle «ck», etc.
   Par contre *Poetica* o re aussi de nombreuses variantes de lettres (lettres initiales, finales, etc.) dont on trouvera des exemples en figure 6. Ici, le principe est di érent[11] ; pour avoir une variante, par exemple un «k bouclé», il su t de taper la touche «k» (ou K) dans la fonte correspondante : *Poetica* est alors faite d'une vingtaine de fontes, certaines n'ayant éventuellement que quelques caractères (15, par exemple, dans *LowerCase Alternate II*).

(c) Il est donc bien dommage que Adobe n'ait pas employé une méthode équivalente pour son *Avant-Garde* qui n'a pas toutes ces ligatures (figure 7) qui faisaient le charme de la version initiale d'Herb LUBALIN et ne propose pas non plus les ligatures de SCORSONE [40, ce cahier, figure 13 page 31]!

9. En donnant à ce mot le sens d'ensemble de glyphes (et de métriques correspondantes) ; par exemple *Helvetica-Bold*.

10. Voir la très belle étude sur ce caractère dans la *Revue suisse de l'imprimerie* [16]. Ce caractère étant très di cile à bien utiliser (choix de caractères, métrique, etc. ; voir ci-après section 4.5) il fait l'objet de démonstration de logiciels d'utilisation automatique de ligatures ; voir par exemple Ω [25, figure1] et les actes du congrès *TUG-95* à paraître dans un prochain numéro de *TUGboat*.

11. Sauf pour les esperluettes qui constituent une fonte à elles seules (on en trouvera quelques unes dans ce *Cahier* pages 131) ; elles sont donc codées A, B, C, etc.

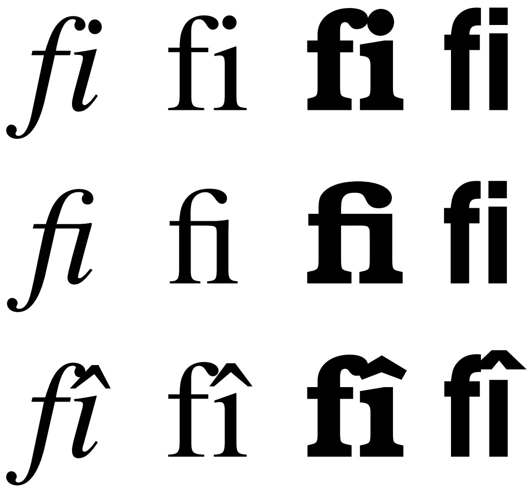

FIG. 8 – *Certaines ligatures ne sont pas toujous bien utiles (lignes du haut et du milieu) alors que certaines suites de glyphes, comme celles de la ligne du bas, devraient bien être remplacées par des ligatures pour éviter le chevauchement de l'accent circonflexe avec la crosse du «f» précédant ! De gauche à droite :* Times-Italic*,* NewCenturySchoolBook*,* Bookman-Demigras*, et* AvantGarde-Demigras*.*

(d) On trouvera, ailleurs dans ce *Cahier GUTenberg*, d'autres exemples de ligatures ou de lettres initiales, notamment celles dessinées par François BOLTANA [15, pages 111 et 120], elles aussi réparties dans plusieurs fontes.

5. Enfin, nous le verrons plus bas (section 5), les algorithmes traçant les glyphes de certaines fontes sont capables de déterminer eux-mêmes de la création de ligatures ; celles-ci sont alors en nombre indéterminé et ne font donc partie d'aucun catalogue puisqu'elles sont créées à la demande.

Il ressort de ces exemples que d'une part la présence de ligatures reste bien liée au seul choix du dessinateur et qu'aucune véritable contrainte technique ne limite le nombre de ligatures pour un caractère ; mais qu'en revanche, la répartition des ligatures dans les fontes manque complètement d'homogénéité, sans même parler de normalisation.

Un corollaire de ce dernier point est qu'il est pratiquement impossible de détecter automatiquement quelles sont les ligatures possibles pour un caractère donné. La seule solution est de se référer aux catalogues fournis par le «fondeur». Il est dommage qu'Adobe n'ait pas étendu à une famille complète la notion d'AFM liées à une fonte donnée (voir ci-dessous section 3).

Enfin signalons, une fois de plus, que le marché des fontes n'étant pas du tout d'origine française, les «fondeurs» américains ne se posent pas toujours les bonnes questions au sujet des ligatures : par exemple il est assez curieux de trouver dans des catalogues de fontes des ligatures pas toujours « techniquement » indispensables [12] alors qu'il y a des dessins intolérables pour d'autres combinaisons de lettres comme celles de la figure 8.

3. Saisie des ligatures

L'utilisateur voyant dans un catalogue de fontes la présence d'une ligature, comment va-t-il pouvoir s'en servir par le biais d'un formateur ? Tout dépend de ce dernier, et il y a plusieurs façons de faire (supposons que l'on veuille obtenir la ligature « fi » dans le mot « afin »).

1. Cas le plus simpliste : certains formateurs, Word par exemple, ne connaissent pas spécialement les ligatures. Il faut alors taper le code interne de cette ligature, par exemple en Word/Macintosch on tapera « `a commande option Q 222 n` » après avoir trouvé (annexe D du manuel Word) que le code de « fi » est 222. Si de plus la ligature est dans une fonte spéciale, c'est le cas du « ct » d'*Adobe-Garamond*, il faut alors non seulement donner ce numéro de code, mais en plus préciser quelle fonte on utilise. Même avec des remplacement automatiques, ce n'est pas une façon très naturelle de saisir un document et on comprend donc que les usagers de ces systèmes rechignent à employer des ligatures !

2. Dans certains cas, le dessinateur d'une fonte considère qu'elle se su t à elle-même ; il peut alors décider que les ligatures sont à utiliser de façon spécifique et proposer des modifications du codage des claviers et/ou des tables de codages ; c'est par exemple le cas de *Champion* [15] où, par exemple, pour obtenir un «s» initial il faut taper la touche $, pour obtenir un d cursif « ∂ » il faut taper @ et pour obtenir l'une des arabesques [15, figure 6], il su t de taper « capital, option, tiret » ; cette façon de faire est bien sûr agréable mais peu portable ; et pose alors un autre problème : comment obtenir, par exemple, « @ » ?

3. Pour de nombreux systèmes, utiliser des ligatures autres que « fi » et « fl » nécessite d'appeler une autre fonte, par exemple *Adobe Garamond Expert* pour disposer de la ligature « » ; mais alors, il faut appeler ce glyphe soit avec son code (132), soit avec le caractère qui a le même code dans le vecteur de codage standard [13].

---

12. Elles ne sont souvent là que parceque les tables de codage les prévoient – c'est en particulier le cas de *Courier* pour lequel on remplace `fi` par `fi` alors qu'il n'y a aucun problème de chevauchement étant donnée la chasse du « `i` ».

13. Ici il s'agit de «Z»). Cette valeur se trouve dans la *Character Access Chart* que fournit Adobe avec une fonte ; pour *Adobe Garamond*, cette charte comprend pas moins de 10 volets.

4. Certains systèmes, comme QuarkXPress, permettent de déclarer, en début de travail, quelles sont les ligatures qui seront utilisées. Le remplacement se fait alors automatiquement par la suite. L'inconvénient de cette méthode est que si l'on change de fonte, il faut refaire cette initialisation.

5. Enfin, quelques rares systèmes, comme TeX, utilisent les informations données dans les tables de métrique, AFM [6] ou TFM [18]. En e et, une fonte est fournie avec une «métrique» qui, outre les tables de chasse, contient aussi des informations comme la présence de ligatures. Par exemple, une fonte PostScript pourra avoir la définition suivante :

```
C 102 ; WX 410 ; N  f ; B ... ; L f ff ; L i fi ; L l fl ;
...
C 174 ; WX 663 ; N fi ; B ... ;
C 175 ; WX 663 ; N fl ; B ... ;
...
C  -1 ; WX 723 ; N ff ; B ... ; L i ffi ; L l ffl ;
C -1 ; WX 1000 ; N ffi; B ... ;
C -1 ; WX 1000 ; N ffl; B ... ;
```

Cette table peut se lire ainsi :

– le caractère de code (C) 102 a une chasse (Wx) de 410/1000 de cadratin, a pour nom (N) «f», … et il existe une ligature (L) avec «f» qui s'appelle « », une avec «i» qui s'appelle «fi» et une avec «l» qui s'appelle «fl» ;

– le caractère de code (C) 174 a une chasse (Wx) de 663/1000 de cadratin et a pour nom (N) «fi» ;

– le caractère de code (C) 175 a une chasse (Wx) de 663/1000 de cadratin et a pour nom (N) «fl» ;

– le caractère de nom (N) « » a une chasse (Wx) de 723/1000 de cadratin et il existe une ligature (L) possible avec le caractère «i» qui s'appelle « » et une avec «l» qui s'appelle « » ; ce caractère n'a pas de code standard (C=–1) et doit être entré spécialement dans le vecteur de codage (voir figure?) ;

– le caractère de nom (N) « » a une chasse de 1000/1000 cadratin et n'a pas de code standard (C=–1) ;

– le caractère de nom (N) « » a une chasse de 1000/1000 cadratin et n'a pas de code standard (C=–1) ;

En suivant ce chaînage, il est alors possible, pour un formateur, de décider des ligatures à utiliser sans que l'utilisateur ait à s'en soucier. C'est ce que nous faisons ici : ce texte a d'abord été préparé en utilisant du *Times* où il n'y a pas les ligatures « ». Sans le moindre changement à notre texte, il a su t d'appeler le *Garamond* ine   cace   inefficace

ine   cace   inefficace

ine   cace   inefficace

FIG. 9 – *Les modifications d'interlettrage ou de crénage ne touchent pas aux glyphes des ligatures. Ligne du milieu : espacement normal ; en haut après augmentation de l'approche de chaque lettre ; en bas après diminution. À gauche avec la ligature «   », à droite sans.*

pour que les ligatures «   » soient prises, automatiquement, en compte. Toutefois, l'auteur doit pouvoir interdire certaines ligatures : ainsi, par exemple en allemand, n'utilise-t-on pas de ligatures «fl» dans un mot composé d'un premier mot se terminant par «f» et d'un commençant par «l»[14]. Malheureusement, ces AFM ne concernent qu'une fonte au sens PostScript : rien ne permet de savoir que «*st*» existe dans *Adobe-Garamond* (en italique, mais pas en romain !), sauf en allant chercher «à la main» dans les tables de codage des diverses fontes «expertes», «tradition», etc.

En résumé, disons que

1. Il ne faut pas confondre les genres : puisqu'une ligature informatique est un remplacement de glyphes, c'est de glyphes qu'il doit s'agir ! Or, l'utilisateur ne connaît que les caractères. C'est donc au système de décider du remplacement des glyphes.

2. Cette lapalissade n'a probablement pas été comprise par les fondeurs de caractères numériques : comme nous l'avons dit plus haut, nous regrettons qu'il n'y ait aucun format de fonte qui o re aujourd'hui un moyen simple de trouver les ligatures d'un caractère donné ! cette lapalissade n'a pas non plus été comprise par les auteurs de formateurs (voir par exemple section 4.1).

14. On écrira par exemple `auf{}lage`, ou on utilisera le caractère «à chasse nulle». Voir pour plus de détails l'article de Yannis HARALAMBOUS ci-après [26]. Alors que ce *Cahier* est pratiquement sous presse, septembre 1995, le réseau Internet *comp.font* fait l'objet de très nombreux échanges sur l'utilisation des ligatures dans les mots composés, notamment en anglais. Assez paradoxalement, de nombreux auteurs de ces messages confondent manifestement division des mots et emploi des ligatures. Comme le rappelle Blanchard dans sa postface ci-après (page 128), l'emploi des ligatures ne relève que de ce que Vox appelait «la sémantique de l'œil».

4. Remarques

Si l'emploi de ligatures est donc assez courant désormais en PAO, certains problèmes restent souvent méconnus. En voici quelques-uns :

4.1. Ligatures et correcteurs orthographiques

L'emploi des ligatures se fait donc souvent (à tort) par remplacement de caractères et non de glyphes. Si on écrit le mot «e uve», il sera codé (voir ci-dessus note 13 page 72) soit `e\132uve` soit `eZuve` (en cachant le changement de fonte). Dans les deux cas, ce mot sera rejeté par un correcteur orthographique. Ceci n'arrive pas avec LaTeX puisqu'en e et on ne touche pas au codage et on saisit «`effluve`».

4.2. Ligatures et division des mots

La même chose s'applique pour la division des mots. Le mot «ine cace» peut être coupé en fin de ligne «inef-ficace», mais si « » est codé ou si l'algorithme de division des mots était employé après le choix automatique des ligatures, cette division ne serait plus possible.

4.3. Ligatures et interlettrage

L'interlettrage, le crénage automatique (*track kerning* des anglo-saxons), les modifications d'approche, etc. sont des techniques peu appréciées[15] des bons typographes qui font confiance aux métriques données (voire imposées, du temps du plomb) par les dessinateurs de caractères. Voici une bonne raison supplémentaire de ne pas en faire : lorsqu'on modifie la distance entre les lettres d'un mot, les ligatures (qui sont alors assimilables à un seul caractère) voient leur glyphe non modifié (on ne modifie pas la distance entre les fûts des f et i dans « ») ce qui rompt le rythme du mot ; voir figure 9.

4.4. Ligatures et grands corps

Certains auteurs ont fait remarquer depuis longtemps que les ligatures et le crénage suivent des nécessités opposées : plus le corps est grand, plus le crénage doit être fort (au point d'arriver à faire se toucher deux caractères consécutifs [30, page 61]). Au contraire, plus les ligatures sont inutiles. Une fonte faisant correctement de l'ajustement optique *(optical scaling)* comme disent le faire les fonte *MultipleMaster* d'Adobe, devraient donc proposer des ligatures fortes pour les petits corps et disparaissant au delà d'une certaine force de corps.

15. À quelques rares exceptions près, comme pour supprimer une veuve ou pour éviter de rejeter le dernier mot d'un titre à la ligne suivante.

4.5. Ligatures et composition fine

Enfin, le problème sans doute le plus difficile avec les ligatures (au sens large et notamment avec les lettres initiales, finales, etc.) est que les fontes qui en disposent proposent bien des métriques (chasse notamment) pour ces caractères, mais qu'elles sont hors contexte. Or, par exemple, on ne fait pas suivre une lettre initiale ornée d'une bas de casse sans un ajustement de chasse pour compenser la différence de blancs entre ces caractères. Bien peu de systèmes de formatage sont assez « intelligents » pour composer des textes avec de telles fontes (voir cependant la note 10 page 70). Il faut donc bien souvent faire ces ajustements «à la main », cas par cas. L'approche (*a priori* surprenante en PAO) de Boltana pour son *Aurore* mérite donc attention : puisque faire un titre avec, par exemple, une anglaise nécessite d'assembler les lettres une par une, alors autant travailler comme autrefois avec les «lettres transfert» ! Il propose donc un système où on clique sur une planche pour prendre un caractère que l'on transfère à l'aide de la souris, à sa place «visuellement» (mais bien sûr avec des marques, visibles sur écran mais invisibles sur papier, pour se guider).

C'est un peu du même esprit que relève l'emploi de « paraphes » pour transformer une capitale en lettre intiale ou finale dans *Trinity* (voir figure 2 page 130).

5. Caractères contextuels

Utiliser des tables de codage pour indiquer les ligatures existantes dans une fonte est une bonne solution pour les langues latines. Il en est tout autrement pour d'autres langues où les combinaisons peuvent parfois atteindre des milliers de glyphes (voir l'article de Yannis Haralambous ci-après page 87 [26]).

On utilise alors ce que l'on appelle des caractères contextuels, c'est-à-dire dont le glyphe dépend du contexte. Mais cette notion est applicable aussi aux langues européennes !

5.1. Choix de glyphe

Le cas le plus simple est celui de l'influence de la place d'un caractère dans un mot sur le choix du glyphe correspondant (voir par exemple les trois premiers glyphes, respectivement pour des positions médiale, initiale et finale de « a » d'*Aurore* de François Boltana dans ce *Cahier* [15, figure10 page 121] ou l'exemple du « chasseur sachant sacher » de Yannis Haralambous [26, ci-après page 93]).

Ce cas se traite de façon élégante en utilisant, dans le formateur, un automate d'état fini, qui à la rencontre de chaque caractère, passe dans un nouvel état (initial, dans un mot, etc.) et en fonction de cet état choisit tel ou tel glyphe. Les ligatures «intelligentes» *(smart ligatures)* doivent alors être codées selon une syntaxe spéciale (mais invisible pour l'utilisateur), comme dans le projet Ω [25, 26, 39].

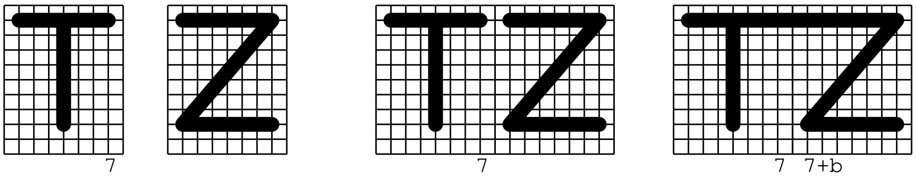

FIG. 10 – *Principe des ligatures « automatiques »*

5.2. Calcul contextuel de glyphe

Mais le contexte peut aussi avoir de l'influence sur autre chose que le choix de tel ou tel glyphe, par exemple sur le dessin du glyphe lui même. Prenons un exemple très simple et supposons que des glyphes soient tracés simplement à l'aide de traits. Le «T» de la figure 10-gauche peut être dessiné en suivant l'algorithme suivant :

```
Tracer trait horizontal en y=9
       de x=1 à x=7
Tracer trait vertical en x=4
       de y=2 à y=9
```

et le « Z » avec quelque chose d'équivalent. Lorsque l'on imprime ces deux glyphes, on obtient le «TZ» normal (figure 10-milieu).

Si maintenant on utilise l'algorithme suivant pour «T» :

```
Tracer trait horizontal en y=9
       de x=1-a à x=7+b
Tracer trait vertical en x=4
       de y=2 à y=9
```

et si on passe à `a` la valeur 0 et à `b` la valeur 2, alors on obtient la ligature de la figure 10-droite.

Plusieurs problèmes techniques sont à résoudre pour faire ceci.

1. Déterminer les valeurs de `a` et `b`, et de façon plus générale quelles parties des glyphes modifier. Même dans un cas simple comme celui-ci, il faut donner une certaine «intelligence» aux descriptions des glyphes. Dans le cas du *Delorme* (figure 11 et [7]) par exemple, nous avons affecté des propriétés topologiques booléennes à chaque caractère, l'algorithme de tracé devenant alors (pour notre TZ ici) :

    ```
    SI le caractère de gauche (ici T)
    ```

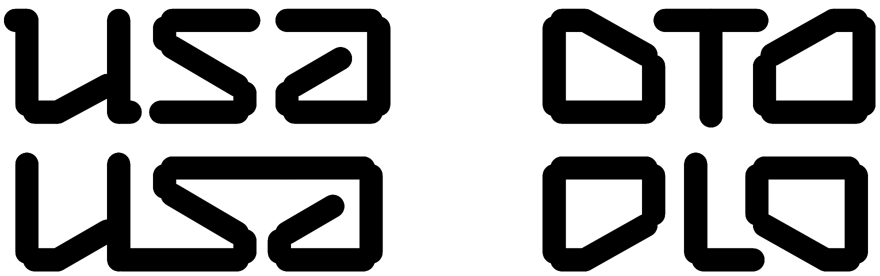

Fig. 11 – *Calcul automatique de ligatures (à gauche) et choix de glyphes (à droite) en fonction du contexte, avec le Delorme [7]*

```
         a une barre horizontale en haut à droite
   ET SI le caractère de droite (ici Z)
         a une barre horizontale en haut à gauche
   ALORS donner à « b » la valeur 2 ;
      SI ...
```

2. Mais dans le cas général, les choses peuvent être très compliquées à décrire d'une part parce que les caractères sont rarement définissables par des segments de droites mais plutôt par des splines que les courbes de Bézier n'arrivent pas à toujours bien simuler (voir ci-dessous section 6.1) et, d'autre part, parce que les modèles de description topologique des caractères ne sont encore qu'à leur tout début (voir par exemple [29]).

3. Par ailleurs, il ne su t pas de lier des caractères pour faire une « ligature » : il faut respecter les rapports de blanc et de noir : par exemple, le « T » est trop éloigné du « Z » dans la figure 10-droite alors que la version ligaturée du « usa » en *Delorme* chasse e ectivement moins que celle normale (figure 11) comme c'est le cas pour « fi » et « » de Garamond (figure 2). On peut évidemment réduire « à la main » l'approche entre les parties de glyphes ainsi ligaturés. Mais, il faut plutôt attendre l'aboutissement des études sur la détection automatique des approches compte-tenu des formes des glyphes, études qui ont par exemple démarré à l'École Polytechnique de Lausanne [12].

4. Il faut ensuite que la machinerie des fontes permette de passer des paramètres, ce qui n'est pas directement le cas en PostScript par exemple. Plusieurs solutions sont alors possibles :

   – Les calculs sont faits avant l'impression et on ne fournit que les glyphes utiles (par exemple « TZ » si on l'utilise dans un texte donné, mais pas « TT » si on ne l'utilise pas) sous forme de plans de bits *(bitmaps)*, ce qui implique une bonne entente entre le formateur (par exemple TEX) et un système de dessin de caractères (ici METAFONT). C'est la méthode employée par Yannis HARALAMBOUS [23].

– En désactivant le mécanisme de cache de PostScript, on peut recalculer chaque glyphe à chaque occurrence, donc pendant l'impression. C'est la notion de caractère «dynamique» [5, 10].
– Entre ces deux méthodes, se trouve la solution actuelle d'Adobe connue sous le nom de *Multiple Master* [41] : en fait tout se passe comme si, en chargeant les fontes, on remplaçait dans l'algorithme les paramètres par leurs valeurs (2 pour a et 0 pour b dans notre exemple de la figure 10) et si on recopiait les algorithmes autant de fois qu'il y a d'ensembles de paramètres : on multiplie les fontes en autant de variantes que nécessaires.

Cette notion de caractère contextuel ou paramétré permet de définir non seulement des ligatures, mais aussi de choisir des glyphes en fonction du contexte (figure 11 et [26]) et surtout de dessiner des glyphes en tenant compte de la taille du corps [10, 23].

6. Ligatures et écriture manuscrite

L'écriture manuscrite occidentale utilise abondamment les ligatures entre lettres. Il est *a priori* surprenant que l'informatique s'intéresse à ce genre de ligatures. Pourtant à bien y réfléchir … En e et, étudier l'écriture manuscrite par ordinateur permet au moins trois types d'applications :

1. la reconnaissance automatique de l'écriture manuscrite,
2. la simulation d'écritures manuscrites, y compris de la calligraphie, ce qui permet d'en garder l'esthétique sans en avoir la technique ou la pratique !
3. et de façon cachée, ces études obligent à définir des modèles d'écriture manuscrite et par là d'avancer dans la connaissance de l'humain.

Faire une synthèse de ces recherches sort complètement du cadre de cet article. Nous ne donnons donc ici que quelques éléments permettant, grace aux références bibliographiques, d'aller plus loin.

6.1. Simulation de l'écriture manuscrite

On trouve dans le marché des fontes pour ordinateurs de nombreux caractères simulant l'écriture manuscrite ou des caractères calligraphiques. On trouvera notamment ci-après

FIG. 12 – *Le recouvrement de caractères, ici en Snell, permet de donner l'illusion de continuité de l'écriture cursive*

dans ce *Cahier* divers exemples de calligraphie assistée par ordinateur dus à François BOLTANA [15, page 107].

L'écriture manuscrite comprend deux classes de caractères : ceux qui ne sont pas attachés [16] et qui ne nous intéressent pas ici, et ceux qui le sont.

Deux principes permettent généralement les liaisons ou ligatures entre lettres de ces alphabets :

1. Chaque lettre est dessinée de façon que les lettres se touchent.
   Pour certaines fontes, par exemple *Snell* de BitStream (figure 12), chaque lettre est a ectée d'un lien avec la suivante, la chasse de la lettre étant plus courte que la *bounding box* de façon que l'extrémité de ce lien soit recouverte par le début du caractère suivant.
   D'autres fontes prévoient que le lien à droite d'un caractère ait la même pente et les mêmes coordonnées finales que le lien à gauche du caractère suivant de façon à garantir une continuité du tracé (exemple figure13).

2. Les lettres qui posent des problèmes (lettres initiales ou finales notamment) font l'objet de traitements spéciaux : en général, une fonte en o re plusieurs séries (voir par exemple les diverses minuscules d'*Aurore* dans ce *Cahier* [15, figure 10 page 121]) ; ou bien, on peut ajouter des ornements à des caractères, par exemple des arabesques à *Champion* (voir ci après page 114), ou des «paraphes» aux capitales de *Trinity* (voir ci après page 130).

Mais ces caractères ont plusieurs inconvénients :

1. En général, ils ressemblent à des caractères typographiques : ils n'ont pas cette liberté, fantaisie ou absence de rigueur qu'a l'écriture manuscrite humaine.

2. Même si les ligatures sont continues, elles sont rarement belles car trop systématiques (il n'y a pas de raison d'avoir un angle constant entre deux lettres !).

16. On trouvera dans [19] une série de caractères «scripts» dessinés par ordinateur.

FIG. 13 – *Ligatures de lettres « manuscrites ». En haut : ligatures classiques ; en bas : ligatures contextuelles ; à gauche : caractères serrés, à droite : caractères plus espacés. D'après* KOKULA *[35].*

3. Elles ne permettent pas de jouer sur l'interlettrage, c'est-à-dire de resserer ou d'élargir son écriture comme on le fait à la main. La figure 13-haut montre quelques exemples de ces défauts.

Diverses études sont en cours pour une meilleure définition de la liaison des caractères manuscrits attachés. Citons des recherches introduisant un minimum de hasard [19], celles visant à définir de nouvelles splines, plus particulièrement des quintites mieux adaptées à la jonction agréable entre lettres [35], des études basées sur le mouvement de la main [20] et enfin des études sur la façon de simuler mathématiquement le ductus des lettres lors de l'emploi de brosses ou plumes [32] permettant d'aller plus loin que KNUTH dans sa première version de Metafont [33].

6.2. Reconnaissance de l'écriture manuscrite liée

La reconnaissance des caractères «imprimés» *(Optical Character Recognition)* a fait d'énormes progrès depuis quelques années à tel point que de nombreux produits commerciaux sont sur le marché. En revanche, celle des caractères manuscrits est encore à l'ordre du jour en matière de recherches. La synthèse de C.Y. SUEN [43] reste encore la référence de base. Quant aux travaux sur la reconnaissance des ligatures, ils sont relativement peu nombreux. Ils utilisent en général des technique de modélisation stochastique par chaînes de Markov et/ou des techniques d'apprentissage. Voir [17, 42].

7. Conclusion

Nous espérons avoir montré dans cet article qu'il y a plusieurs partenaires dans l'emploi des caractères informatisés et en particulier des ligatures. On ne dira sans doute jamais assez qu'il ne faut pas confondre une « fonte » et l'usage que l'on en fait. On n'a pas le droit de dire par exemple que «le Macintosch» n'o re pas la ligature « ». C'est vrai avec certaines fontes, ou avec certains logiciels, ou les deux. C'est faux avec d'autres. Mais l'utilisateur doit être conscient qu'il faut ou bien utiliser un logiciel «intelligent» (et nous pensons que LaTeX et Ω sont parmis les seuls à faire un traitement quasi-automatique des ligatures), ou bien s'investir beaucoup comme on le faisait du temps du plomb.

Nous voudrions enfin insister sur le fait que c'est aux utilisateurs de se battre pour disposer des ligatures utiles, mais aussi des caractères accentués comme «É» ou de certains comme «œ». Il ne faut surtout pas croire que c'est une question de technique : c'est une question de politique (qu'elle soit commerciale, étatique, etc.) !

Bibliographie

[1] Adobe Systems Incorporated, *Adobe Garamond*, publication n° 0399 1252, septembre 1992.

[2] Mark Agetsinger, «Adobe Garamond: a review», *Printing History*, vol. 26–27 (1991–1992), 69–98.

[3] Jacques André, « La fonte du jour : cmr », *La Lettre GUTenberg*, n° 5, avril 1995, 26–29.

[4] Jacques André, « Introduction : vous avez dit ligature ? », *Cahiers GUTenberg* n° 22 (ce cahier), septembre 1995, 1–4.

[5] Jacques André & Bruno Borghi, «Dynamic fonts», in *Raster Imaging and Digital Typography* (André and Hersch eds.), Cambridge University Press, 1989, 198–203.

[6] Jacques André & Justin Bur, «Métrique de fontes PostScript», *Cahiers GUTenberg*, n° 8, mars 1991, 29–50.

[7] Jacques André et Christian Delorme, «Le Delorme : un caractère modulaire et dépendant du contexte», *Communication et langage*, 86, 1990, 65–76.

[8] Jacques André & Michel Goossens, «Codage des caractères et multilinguisme : de l'Ascii à Unicode et Iso/Iec–10646», *Cahiers GUTenberg*, n° 20, mai 1995, 1–53.

[9] Jacques André et Phlippe Louarn, «La fonte du jour : Adobe-Garamond», *La Lettre GUTenberg*, n° 4, décembre 1994, 18–19.

[10] Jacques André & Irène Vatton, «Dynamic Optical Scaling and Variable Sized Characters», *EPODD, Electronic Publishing, Origination, Dissemination and Design*, vol. 7(4), décembre 1994, 231–250.

[11] Jacques André & Adolf Wild, *Ligatures, typographie et informatique*, Rapport de recherche INRIA n° 2429, décembre 1994.

[12] Claude Bétrisey, *Génération automatique de contraintes pour caractères typographiques à l'aide d'un modèle topologique*, thèse, École Polytechnique Fédérale de Lausanne, 26 mai 1993.

[13] Gérard Blanchard, « Nœuds & esperluettes – actualités et pérennité d'un signe », *Communication et langages*, n° 92, 1992, p. 85–101 ; et *Cahiers GUTenberg* n° 22 (ce cahier), septembre 1995, 43–59.

[14] *Actes des journées Garamond (Fontainebleau 1994)*, en préparation sous la direction de Gérard Blanchard pour les éditions des Rencontres de Lure.

[15] François Boltana, «Ligatures & calligraphie assistée par ordinateur», *Cahiers GUTenberg* n° 22 (ce cahier), septembre 1995, 107–124

[16] Max Caflish, «Die Cancelleresca in Vergangenheit und Gegenwart anhand der Poetica gargestellt », *Revue suisse de l'imprimerie*, n° 2, 1995, 1–16.

[17] J. Camillerapp, G. Lorette, G. Ménier, H. Oulhadj, J.C. Pettier, « O -line and on-line methods for cursive handwriting recognition», *From Pixels to features III*, International Workshop on Frontiers in Handwriting Recognition, Norht-Holland, 1992, 273–287.

[18] Alain Cousquer & Éric Picheral, «Polices, TEX et Cie», *Cahiers GUTenberg*, n° 9, juillet 1992, 3–31.

[19] L. Devroye and M. McDougall, « A Methodology for random fonts applied to hand written scripts », à paraître dans *EPODD, Electronic Publishing, Origination, Dissemination and Design*, vol. 8, 1995.

[20] E. H. Dooijes, « Rendition of quasi-calligraphic script defined by pen trajectory », in J. André and R. Hersch (eds.), *Raster Imaging and Digital Typography*, Cambridge University Press, 1989, 251–260.

[21] John Dreyfus et François Richaudeau, *La chose imprimée*, Éditions Retz, Paris, 1977.

[22] Thierry Gouttenègre, «Ligatures et bâtardes bourguignonnes : du xv au xx^{e} siècle», *Cahiers GUTenberg* n° 22 (ce cahier), septembre 1995, 101–105.

[23] Yannis HARALAMBOUS, «Parametrization of PostScript fonts through METAFONT – an alternative to Adobe Multiple Master fonts», *EPODD – Electronic Publishing, Origination, Dissemination and Design*, vol. 6, n° 3, (Raster Imaging and Digital Typography Proceedings), september 1993, 145–157.

[24] Yannis HARALAMBOUS, «Typesetting Khmer», *EPODD – Electronic Publishing, Origination, Dissemination and Design*, vol. 7, n° 4, december 1994, 197–215.

[25] Yannis HARALAMBOUS et John PLAICE, «Ω, une extension de TEX incluant UNICODE et des filtres de type Lex», *Cahiers GUTenberg*, n° 20, mai 1995, 55–79.

[26] Yannis HARALAMBOUS, «Tour du monde des ligatures», *Cahiers GUTenberg* n° 22 (ce cahier), septembre 1995, 87–99,

[27] *Cahiers GUTenberg*, numéro spécial «TeX et l'arabe», en préparation sous la direction de Yannis HARALAMBOUS, 1995.

[28] Roger HERSCH (ed.), *Visual and Technical Aspects of Types*, Cambridge University Press, 1993.

[29] Jacky HERZ and Roger HERCH, «Analyzing character shapes by string matching techniques», *EPODD – Electronic Publishing, Origination, Dissemination and Design*, vol. 6, n° 3, september 1993, 261–272.

[30] Peter KAROW, *Digital Formats for Typefaces*, URW Verlag, Hambourg, 1987.

[31] Peter KAROW, *Schrifttechnologie – Methoden un Werkzeuge*, Springer-Verlag, 1992. Version anglaise : *Font Technology*, URW Verlag, Hambourg, 1994.

[32] R. Victor KLASSEN, «Variable width splines: a possible font representation?», *EPODD, Electronic Publishing, Origination, Dissemination and Design*, vol. 6, n° 3, september 1993, 183–194.

[33] Donald KNUTH, «The concept of a meta-font», *Visible language*, XVI, 1, 1982, 3–27. Traduit en français : «Le concept de Metafonte», *Communication et langage*, n° 55, 1er trimestre 1983, 40–53.

[34] Donald KNUTH, «The new version of TEX and METAFONT», *TUGboat*, vol. 10, n° 3, novembre 1989, 325–327. Traduit en français : «TEX3.0 ou le TEX nouveau va arriver», *Cahiers GUTenberg*, n° 4, décembre 1989, 39–45.

[35] Michael KOKULA, «Automatic generation of script font ligatures based on curve smoothness optimization», *EPODD, Electronic Publishing, Origination, Dissemination and Design*, vol. 7(4), december 1994, 217–229.

[36] Piero de MACCHI, *L'avventura Didot: caratteri da stampa e nuove tecnologie*, De Macchi Progetti Grafici, Turin, 1994.

[37] Bernard MARTI et co-auteurs, *Télématique - techniques, normes, services*, Dunod, 1990.

[38] Jérôme PEIGNOT, «Petit traité de la ligature», *Communication et langages*, n° 73, 3e trimestre 1987, 20–35.

[39] John PLAICE, «Language-Dependent Ligatures», *TUGBoat*, vol. 14, n° 3, 1993, 270–274.

[40] René PONOT, « Le *Didot* a-t-il besoin de ligatures ? », *Cahiers GUTenberg* n° 22 (ce cahier), septembre 1995, 17–42

[41] Jonathan SEYBOLD, «Adobe's MultiMasters' technology: breeakthrough in type aesthetics», *The Seybold Report on Desktop Publishing*, vol. 5, n° 7, 1991, 3–7.

[42] B.K. SIN & J.H. KIM, «A Statistical Approach with HMMs for On-line Cursive Hangul (Korean Script) Recognition», *Proceedings of the ICDAR conference*, 1993, 147–154.

[43] C.Y. SUEN, M. BERTHOLD and S. MORI, «Automatic Recognition of Handprinted Characters : the State of the Art», *IEEE trans. on Pattern Analysis and Machine Intelligence*, 68(4), 469–487, 1980.