



المعهد الملكي للثقافة الأمازيغية
ⵎⴰⴷⵓⴷⴰ ⵏ ⵏⵓⴷⵓⴷⴰ ⵏ ⵏⵓⴷⵓⴷⴰ
INSTITUT ROYAL DE LA CULTURE AMAZIGHE

+ΣΘΣΘRη+ ΧΟ ΣΧΟ | +⌘8QΣ
Λ ΠΣΙ +ΘΙCοηογ+

La Typographie entre les domaines
de l'art et de l'informatique

Direction et coordination
Lahbib ZENKOUAR
Youssef AIT OUGUENGAY



***LA TYPOGRAPHIE ENTRE
LES DOMAINES DE L'ART
ET DE L'INFORMATIQUE***

Royaume du Maroc
Institut Royal de la Culture Amazighe
Centre des Etudes Informatiques, des
Systèmes d'information et de
Communication



ⵜⴰⴷⵓⴷⴰ ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴷⵓⴷⴰ
ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴷⵓⴷⴰ ⵜⴰⴳⵓⴷⴰⵜ
ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ
ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ

1^{er} atelier international sur les technologies de l'information
et de la communication sous le thème :

La typographie entre les domaines de l'art et de l'informatique

ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ
ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ ⵜⴰⴳⵓⴷⴰⵜ

Actes de l'atelier

Rabat, 23-24 septembre 2004

ⴰⴳⵓⴷⴰⵜ, 23-24ⵜⴰⴳⵓⴷⴰⵜ 2004

Publications de l'Institut Royal de la Culture Amazighe

2005

Publications de l'Institut Royal de la Culture Amazighe
Centre des Etudes Informatiques, des Systèmes d'Information et de
Communication (CEISIC)

Série: Colloques et séminaires N°-13

Titre	: La typographie entre les domaines de l'art et de l'informatique
Direction et coordination	: Lahbib ZENKOUAR et Youssef AIT OUGUENGAY
Editeur	: Institut Royal de la Culture Amazighe
Réalisation éditoriale	: Mounia BOUMEDIANE
Suivi éditorial	: CTDEC
Couverture	: Nadia KIDDI
Imprimerie	: EL Maârif Al Jadida-Rabat
Dépôt légal	: 2008/0108
ISBN	: 9954 - 439 -93 - 5
Copyright	: © IRCAM

Table des Matières

Présentation	7
Introduction à Unicode et à 'ISO10646	13
Patrick ANDRIES	
Propriétés et Formats d'Unicode	37
Lahbib ZENKOUAR	
Unicode et les Polices : Deux Mondes	55
Patrick ANDRIES	
المغاربة وفن الكتابة الجديد الطباعة الحجرية (الليثوغرافيا) نموذجاً لطيفة الكندوز	95
Typographie, Historique et Pratique	97
Christian PAPUT	
La Naissance de l'Ecriture et de la Typographie	101
Christian PAPUT	
La Calligraphie Tifinaghe : Une Expression Artistique en Construction	115
Lahbib FOUAD	
La création Typographique contemporaine : Les Grecs du Roi	141
Frank JALLEAU	
Dessiner une police tifinaghe	149
Alexis LOGIE	
Quelques aspects de la numérisation des polices de caractères : cas de Tifinaghe	159
Youssef AIT OUGUENGAY	
La Police Open Type Hapax berbère	183
Patrick ANDRIES	
Codage Pluriel des Caractères	197
Jacques ANDRE	
Patrimoine, Fontes Typographiques et WEB	209
Jacques ANDRE	
Le Web Multilingue ou l'Intégration du Web	211
Najib TOUNSI	
Les Normes du Tri, du Clavier et Unicode	223
Mohamed OUTAHAJALA	

Les Graphes UNL : un Concept Unificateur pour l'Intégration de l'Amazigh dans des Documents Multilingues 239
Ali RACHIDI

Ce Que Vous Devez Savoir au Sujet du Bidi et du Balisage au Sein des Blocs 253
Richard ISHIDA

TABLES RONDES:

Aspects typographiques et écriture cursive de l'alphabet amazighe 265
Coordinateur : Franck JALLEAU, Rapporteur : Youssef AIT OUGUENGAY

Langue amazighe et technologies d'information et de communication 269
Coordinateur : Richard ISHIDA, Rapporteur : Mohamed OUTAHAJALA

Atelier international sur « La typographie entre les domaines de l'art et de l'informatique »

Dans le cadre de ses activités, le centre des Etudes Informatiques, des Systèmes d'Information et de Communication (CEISIC), IRCAM, a organisé les 24 et 25 septembre 2004, un séminaire sur la typographie entre les domaines de l'art et de l'informatique.

Ce séminaire s'est tenu à l'institut Goethe sis à Rabat.

ARGUMENTAIRE

Parmi les choses qui distinguent l'être humain des autres créatures est le fait qu'il écrit. L'écriture possède l'intéressante particularité de conserver l'oral. Elle permet ainsi aux civilisations et aux cultures de se transmettre à travers les âges et de subsister quelque soient les vicissitudes de l'époque.

Parmi les derniers outils de l'écriture, l'ordinateur est devenu un véritable espace à l'intérieur duquel le texte vit et se métamorphose au rythme des algorithmes informatiques et tout ceci par un simple clic de la souris ou d'une simple touche de clavier.

L'écriture informatique utilise les codages des caractères ; ceux-ci utilisent à leurs tours les fontes et les polices. Ces deux aspects sont liés mais répondent à des besoins différents. Le codage est une conceptualisation informatique de l'écriture. Il a pour but de caractériser l'information dans ces composants élémentaires. Un codage est une table de caractères. Un caractère étant une notion abstraite, immatérielle. Par contre, une fonte ou police est un ensemble de glyphes. Un glyphe est un dessin donnant forme à ce caractère informatique abstrait. Il appartient à l'artiste de créer les meilleurs glyphes; son art de la typographie nous inspirera des formes appartenant à tel ou tel caractère.

Le lecteur doit être conscient qu'en entrant dans l'espace de l'écriture informatique, il participe à un va et vient incessant entre caractères et glyphes, en appuyant sur une touche un caractère est transmis du clavier portant des glyphes, en faisant de la recherche morphologique dans un texte, l'ordinateur affiche des glyphes à l'écran.

Le document électronique est une prolongation de cet aspect afin d'aboutir à un codage international de tous les glyphes reconnus des écritures terrestres. Cet effort les introduira dans l'ordinateur et les inscrira dans ces métamorphoses nécessaires aux langues et à leur maintien dans le monde de l'écriture.

Le but de ce séminaire international est de trouver des solutions aux problèmes qui se posent à l'introduction des caractères, en particulier de l'alphabet amazighe, dans les outils informatiques tant du côté codage que celui de la beauté des glyphes et de l'art de les dessiner.

Ces deux journées du workshop débattront des sujets correspondant aux axes suivants :

AXES :

- √ La typographie traditionnelle
- √ La typographie numérique
- √ La problématique d'un passage réussi de la typographie traditionnelle à la typographie informatique
- √ Le document électronique multilingue

Comité d'organisation

Lahbib ZENKOUAR

Youssef AIT OUGUENGAY,

Mohamed OUTAHAJALA,

Mounia BOUMEDIANE

PROGRAMME
Vendredi 24 septembre 2004

9h 00 -10h 00 : Séance d'ouverture

Président : M. Lahbib ZENKOUAR

Allocution de M. le RECTEUR

Pause

10h 00 -12h 00 : Première séance « Le standard de codage unicode »

Président : Jacques ANDRE

Professeur, INRIA, Rennes, France

10h 00 - 10h 30 : Patrick ANDRIES

Expert Iso Unicode, canada « Aspects du standard unicode »

10h 30 - 11h 00 : Lahbib ZENKOUAR

Chercheur, Ceisic, ircam « Propriétés et formats d'unicode »

11h 00 - 11h 30 : Patrick ANDRIES

Expert iso-unicode, Canada «Polices et unicode »

12h 00 - 14h 30 : Déjeuner

Deuxième séance : 15h 00 - 18h 30 « La typographie traditionnelle »

Président : Abdelmoumen BENABDELJALIL

Directeur Art'COM, Casa

14h 30 - 15h 00 : Latifa GUENDOOUZ "المغاربة وفن لكتابة الجديد : الطباعة الحجرية نموذجاً"

Chercheur, IURS, Rabat

15h 00 - 15h 30 : Christian PAPUT « La typographie : historique et pratique »

Imprimerie Nationale, Paris, France

15h 30 - 16h 00 : Fouad LAHBIB « La calligraphie tiffinaghe »

Chercheur, Ceaelpa, ircam

16h 00 - 16h 30 : Azzouz TNIFASS « Aspects et commentaires sur l'écriture »

Critique d'art, chercheur, Ministère de la culture

16h 30 - 17h 00 : Pause café

17h 00 - 19h 00 : Troisième séance « La typographie numérique »

Président : Christian PAPUT

Imprimerie Nationale de Paris, France

17h 00-17h 30 : Franck JALLEAU « La création typographique contemporaine »

Professeur à l'Ecole des Beaux-Arts Estienne, Paris

17h 30-18h 00 : Alexis LOGIE « Dessiner une police de caractères tiffinaghe »

Professeur à l'Ecole Art'Com, Casa

18h 00 - 18h 30 : Youssef AIT OUGUENGAY « Numérisation du Tifinaghe : quelques aspects et problèmes »

18h 30 - 19h 00 : Patrick ANDRIES « La police Hapax berbère »

Expert iso-unicode, Canada

Samedi 25-09-04

9h 00-10h 30 : Première séance « Le document électronique multilingue »

Président : Patrick ANDRIES

Expert iso-unicode, Canada

9h 00 - 9h 30 : Ould Braham OUAHMI, ZGHIBI Rachid MSH, Paris “Ecriture berbère et Unicode”

9h 30 - 10 h 00: Jacques ANDRE « Patrimoine, fontes, typographie et web »

Professeur, INRIA, Rennes, France

10h 00 - 10h 30 : Najib TOUNSI « Les nouvelles technologies du web »

Professeur, EMI, Rabat

10h 30 - 11h 00 : Pause café

11h 00 - 12h 30 : Deuxième séance « Le document électronique multilingue »

Président : Najib TOUNSI professeur, EMI, Rabat, Responsable du bureau marocain W3C

11h 00 - 11h 30 : Mohamed OUTAHAJALA « Les normes du tri, du clavier et unicode»

Chercheur, ceisic, ircam

11h 30 - 12h 00 : Ali RACHIDI « Les graphes UNL : « un concept unificateur pour l'intégration de l'amazighe dans les documents multilingues » Chercheur, ENCG, Agadir

12h 00 - 12h 30: Richard ISHIDA”An introduction to writing systems”

W3C Internationalization

12h 30 - 15h 00: Déjeuner

15h 00 - 18h 30: Troisième séance Table ronde

15h 00- 16h 00 : « Aspects typographiques et écriture cursive de l'alphabet amazighe »

Coordinateur : Franck JALLEAU

Rapporteur : Youssef AIT OUGUENGAY

16h 00 - 16h 30 : pause café

16h 30 - 17h 30 : Langue amazighe et Technologie d'information et de communication

Coordinateur : Richard ISHIDA

Rapporteur : Mohamed OUTAHAJALA

17h 30 : Clôture

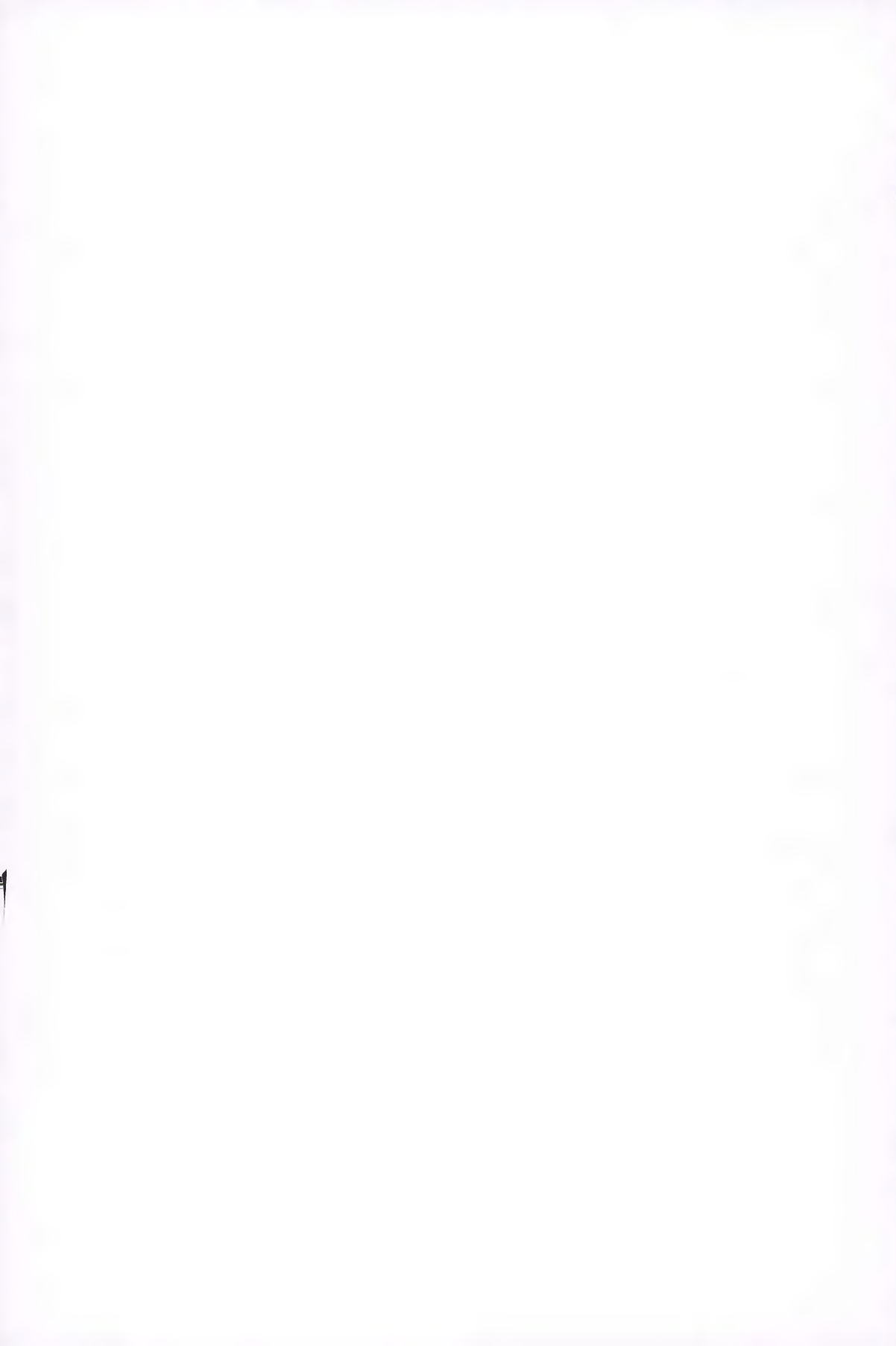
REMERCIEMENTS

Je tiens à remercier et à féliciter l'équipe d'organisation qui a veillé au bon déroulement des opérations et démarches ayant abouti à la tenue de ces journées.

Je ne saurais oublier de remercier le directeur de l'école Art'Com de Rabat et de Casablanca, l'Ecole Nationale des Beaux Arts de Casablanca, mes collègues experts des organismes internationaux, professeurs et universitaires pour l'excellence des communications et des débats qui ont lieu au cours de ces deux journées qui ont concerné tout particulièrement le caractère informatique amazighe de l'alphabet Tifinaghe tant du côté codage et représentation informatique que du côté esthétique des polices et écritures.

Ce premier workshop international sur le caractère informatique a pour thème :
« La typographie entre les domaines de l'art et de l'informatique »

Lahbib ZENKOUAR



INTRODUCTION A UNICODE ET A L'ISO 10646

Patrick Andries

Conseils Hapax <http://hapax.qc.ca>

Membre du consortium Unicode

1. Introduction

1.1. Pourquoi Unicode ?

Les jeux de caractères utilisés avant l'avènement d'Unicode possédaient des architectures très différentes les uns des autres. Pour plusieurs, la simple détection des octets représentant un caractère était un processus contextuel complexe. Les jeux de caractères classiques ne pouvaient au mieux prendre en charge que quelques langues. La prise en charge de plusieurs langues à la fois était difficile, voire impossible. Aucun jeu de caractères ne fournissait toutes les lettres, les signes de ponctuation et les symboles techniques en usage courant utilisés pour une seule langue comme le français.

Produire des logiciels destinés à différents marchés se résumait à une alternative simple : écrire des versions différentes du logiciel pour chaque jeu de caractères ou rendre le code du logiciel nettement plus compliqué pour traiter tous les jeux de caractères nécessaires. Ces deux options impliquent plus de codage, plus de tests, plus d'entretien et plus de soutien aux utilisateurs. Elles rendent la production de logiciels destinés à l'exportation plus chère et retardent leur lancement ce qui n'est pas sans conséquence sur les revenus.

Le marché actuel rassemble souvent des données provenant de différentes sources (il suffit de penser à l'Internet). Le caractère codé correspondant à une lettre aussi simple que «A» peut varier en fonction du jeu de caractères et rendre le repérage, le tri et d'autres opérations fort difficiles. D'autres problèmes surgissent quand on mélange des données provenant de différents jeux de caractères (comme c'est le cas lors d'un copier-coller entre deux fichiers codés différemment). Il faut alors étiqueter ou baliser les données avec des informations sur le jeu de caractères d'origine et créer de la sorte un nouveau codage à états. Perpétuer les anciennes méthodes de production dans ce contexte ne peut qu'entraîner des coûts prohibitifs ou, pire, conduire au désastre.

Le standard Unicode^(1,2) est un mécanisme universel de codage de caractères. Il définit une manière cohérente de coder des textes multilingues et facilite l'échange de données textuelles. Obligatoire pour la plupart des nouveaux protocoles de l'Internet⁽³⁾, mis en œuvre dans tous les systèmes d'exploitation et langages informatiques modernes⁽⁴⁾, Unicode est la base de tout logiciel qui veut fonctionner aux quatre coins du monde.

Grâce à Unicode, l'industrie informatique peut assurer la pérennité des données textuelles tout en évitant la prolifération de jeux de caractères et en augmentant l'interopérabilité des données. Enfin, Unicode simplifie le développement de logiciels et en réduit les coûts. En effet, Unicode permet de coder tous les caractères utilisés par toutes les langues écrites du monde (plus d'un million de caractères sont réservés à cet effet). Tous les caractères, quelle que soit la langue dans laquelle ils sont utilisés, sont accessibles sans aucune séquence d'échappement. Le codage de caractère Unicode traite les caractères alphabétiques, les caractères idéographiques et les symboles de manière équivalente, avec comme conséquence qu'ils peuvent se côtoyer dans n'importe quel ordre avec une égale facilité.

Le standard Unicode attribue à chacun de ses caractères un numéro⁽⁵⁾ et un nom. À ce titre, il ne diffère guère des autres standards ou normes de codage de caractères. Cependant Unicode fournit d'autres renseignements cruciaux afin de s'assurer que le texte codé sera lisible : la casse des caractères codés, leur **directionnalité** et leurs propriétés alphabétiques. Unicode définit également des renseignements sémantiques et comprend des tableaux de correspondance de casse ou des conversions entre Unicode et les répertoires de divers autres jeux de caractères importants.

À l'heure actuelle, les données Unicode peuvent être codées sous trois formes principales : une forme codée sur 32 bits (UTF-32), une forme sur 16 bits (UTF-16) et une forme de 8 bits (UTF-8) conçue pour faciliter son utilisation sur les systèmes ASCII préexistants. Le standard Unicode est identique à la norme internationale ISO/CEI 10646 en ce qui concerne l'affectation des caractères (leur numéro) et leurs noms⁽⁶⁾. Toute application qui se conforme à Unicode se conforme donc à l'ISO/CEI 10646.

(1) - Généralement toute mention au « standard Unicode » s'applique également à la norme internationale ISO/CEI 10646.

(2) - Une traduction française complète annotée et mise à jour du Standard Unicode 3.2 est disponible à l'adresse <<http://hapax.qc.ca>>. Cette traduction a bénéficié du concours financier du gouvernement du Québec.

(3) - XML, HTML, WML, Corba 3.0, LDAP, etc.

(4) - Exemples : Java, ECMAScript (Javascript), MS Windows 2000 et XP, Mac OS/X.

(5) - Une « valeur scalaire Unicode », dans le jargon d'Unicode.

(6) - Le standard Unicode ne précise pas de noms français, cependant la norme internationale ISO/CEI 10646 a été publiée en anglais et en français. Les noms utilisés ici sont les noms officiels de la version française de l'ISO/CEI 10646.

1.2. Principes directeurs

L'objectif principal du standard Unicode était de remédier à deux écueils sérieux et fréquents dans la plupart des programmes informatiques multilingues : l'ambiguïté des polices qui utilisent souvent les mêmes valeurs pour coder des caractères et des symboles totalement différents et l'utilisation de multiples jeux de caractères incompatibles provenant de normes nationales et industrielles contradictoires. Ainsi, même en Europe occidentale, des conflits existent entre la page de code 850 et l'ISO/CEI 8859-1. Dans le cas de logiciels qui prennent en charge les idéogrammes d'Extrême-Orient, le même ensemble d'octets utilisés en ASCII peut également servir de deuxième octet pour les caractères à deux octets. Ces logiciels doivent donc être capables de distinguer les caractères ASCII codés sur un octet des caractères à deux octets.

Afin de résoudre ces problèmes, le consortium Unicode a conçu une méthode uniforme de codage des caractères à la fois plus efficace et plus souple que les systèmes de codage dits historiques. Le nouveau système devait satisfaire les besoins techniques multilingues tout en codant un grand éventail de caractères qui permettent d'assurer les besoins typographiques de qualité professionnelle et de la micro-édition dans le monde entier.

1.3. La norme ISO/CEI 10646 et le standard Unicode

En 1984, le JTC1/SC2⁽⁷⁾ de l'ISO/CEI définissait le mandat du groupe de travail numéro 2 (GT2) : « élaborer une norme établissant un répertoire de caractères graphiques des langues écrites du monde et son codage ». À la même époque, Joe Becker de Xerox à Palo Alto travaillait sur un jeu de caractères universel qu'il nomma Unicode. Quelques années plus tard, en 1988, plusieurs industriels, dont Xerox, se réunirent pour former ce qui allait devenir le consortium Unicode. Conscients des bénéfices d'une seule norme universelle de codage de caractères, les membres du Consortium Unicode collaborèrent avec les représentants de l'Organisation internationale de normalisation (ISO) à la fin 1991.

À la suite de négociations officielles sur la convergence des deux codes, les deux répertoires fusionnèrent en janvier 1992. Depuis lors, une étroite collaboration et une liaison officielle entre les deux comités ont permis d'assurer la synchronisation des amendements et additions aux deux normes de sorte que leurs répertoires respectifs et leurs codages sont aujourd'hui identiques.

Le répertoire du standard Unicode est rigoureusement identique à celui de l'ISO/CEI 10646. Cette identité point par point s'applique à tous les caractères codés des deux normes y compris les caractères idéographiques extrême-orientaux

(7) - Il s'agit du sous-comité conjoint de l'ISO et du CEI qui se penche sur la codification des caractères.

(également appelés caractères han). La norme ISO/CEI 10646 attribue à chaque caractère un nom et une valeur de code, le standard Unicode assigne les mêmes noms et valeurs de code mais il fournit également d'importants algorithmes de mise en œuvre, des propriétés de caractères et d'autres renseignements sur la sémantique de ceux-ci.

Le comité technique du consortium Unicode (CTU) est le groupe de travail au sein du consortium chargé de la création, de la mise à jour et de la qualité du standard Unicode. Le CTU contrôle toutes les données techniques fournies au consortium et prend des décisions quant au contenu de ces données. Les membres de plein droit du consortium, qui doivent s'acquitter d'une cotisation annuelle, se prononcent sur les décisions du CTU. Les membres associés, les membres experts ainsi que les dirigeants du consortium font partie du CTU, mais ne bénéficient pas du droit de vote. Le président du CTU peut inviter d'autres personnes à participer aux discussions techniques, sans droit de vote toutefois.

La dernière version papier d'Unicode est la version 4.0, publiée sous la forme d'un épais livre en 2003. Entre-temps, elle a été mise à jour par en mars 2005 à l'aide d'un rapport séparé de sorte que la version actuelle d'Unicode est la version 4.1. La prochaine version papier d'Unicode, la 5.0, n'est pas prévue avant 2006.

2. Quels caractères Unicode et l'ISO/CEI 10646 normalisent-ils ?

La base de tout jeu de caractères est une correspondance entre des valeurs de code (des numéros) et des caractères. Il ne faut pas confondre un caractère avec sa représentation visuelle (le glyphe représentatif qui l'illustre dans les tableaux de caractères). C'est pourquoi Unicode fournit les renseignements sémantiques nécessaires pour décrire les caractères qu'il code. Nous reviendrons sur cette distinction fondamentale entre caractère et glyphe par la suite (v. _3.1, Caractère abstrait, caractère codé et glyphe).

L'espace de codage du standard Unicode se divise en plans de 64K cellules (voir la figure 1). Chaque cellule peut se voir affecter un caractère. Le premier plan (plan 0016), le plan multilingue de base (PMB), comprend des caractères usuels dans les écritures alphabétiques, syllabiques et idéographiques ainsi que divers chiffres et symboles. Le contenu des zones à usage privé du PMB et des plans à usage privé (0F16 et 1016) n'est pas prescrit par Unicode, en d'autres mots on est libre d'y affecter des caractères de son choix. La zone d'indirection est un ensemble de cellules réservées pour UTF-16 (voir la figure 2 et la section de l'article du prof. Zenkour dans ces mêmes actes relative aux formes en mémoire des caractères) afin de pouvoir distinguer les caractères codés sur un seize de ceux codés sur deux. La figure 3 donne le détail des premiers blocs du PMB.

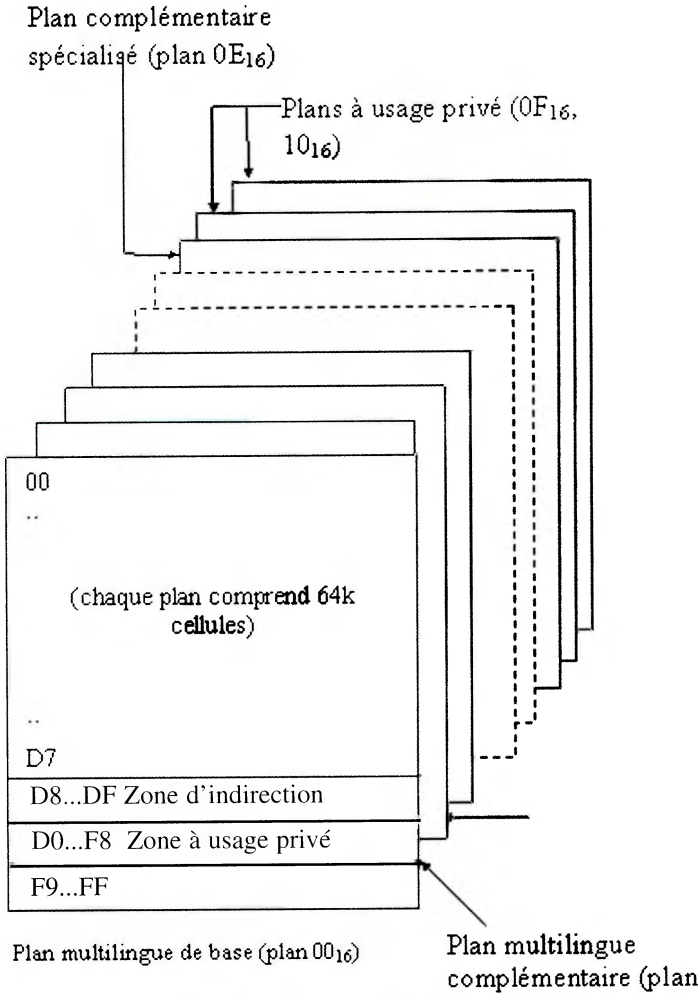


Figure 1. Plans et zones de codage

00	1.1 Écritures générales (voir ci-après)					
33						
34	1.2 Supplément A aux idéogrammes unifiés CJC					
4C	1.3			1.4 Hexagrammes du Livre des Mutations		
4D						
4E	1.5 Idéogrammes unifiés CJC					
9F						
A0	1.6 Syllabaire yi des Monts frais					
A3	1.7			1.8 Clés yi		
A4						
A5	1.9					
A6						
A7	1.10 Suppl. modif. de ton		1.11			
A8	1.12 Sylotí nâgrî		1.13 Phags-pa		1.14	
A9	1.15					
AB						
AC	1.16 Syllabaire hangûl					
D7						
D8	1.17 Zone d'indirection					
DF						
E0	1.18					
F8	1.19 Zone à usage privé					
F9	1.20 Idéogrammes de compatibilité CJC					
FA						
FB	1.21 Formes de présentation alph.			1.22		
FC	1.23 Formes de présentation arabes A					
FD						
FE	1.24 Sél. Var.	1.25 For. Vert.	1.26 Demi-sign. comb.	1.27 Compat CJC	1.28 Petites variantes	1.29 Formes ara. B
FF	1.30 Formes de demi et pleine chasse				1.31 Spéciaux	

1.32

1.34 = absence de caractères

1.35 1.36 = réservé à une normalisation ultérieure

1.37 NOTE - Les frontières n'indiquent que des divisions approximatives.

1.38 **Figure 2. Plan multilingue de base (PMB)**

00	1.41	1.42 Latin de base	1.43	1.44 Supplément Latin-1
01	1.45 Latin étendu A			1.46 Latin étendu B
02	1.45 Latin étendu B	1.48 Alph. phon. internat.	1.49 Modificateurs	
03	1.50 Signes combinatoires		1.51 Grec et copte	
04	1.52 Cyrillique			
05	1.53 Suppl. cyrillique	1.54 Arménien		1.55 Hébreu
06	1.56 Arabe			
07	1.57 Syriaque	1.58 Supp. arabe	159 Thâna	1.60
08	1.61			
09	1.62 Dévanâgarî		1.63 Bengali	
0A	1.64 Gourmoukhî		1.65 Goudjarati	
0B	1.66 Oriya		1.67 Tamoul	
0C	1.68 Télougou		1.69 Kannara	
0D	1.70 Malayalam		1.71 Singhalais	
0E	1.72 Thaï		1.73 Lao	
0F	1.74 Tibétain			
10	1.75 Birman			1.76 Géorgien
11	1.77 Jamos hangûl			
12	1.78 Éthiopien			
13	1.79	1.78 Supp. éthiopien		1.81 Chérokî
14	1.82 Syllabaires autochtones canadiens			
16	1.83	1.84 Ogam		1.85 Runes
17	Tagalog	Hanounóo	Bouhid	Tagbanoua
17				Khmer
18	1.86 Mongol			1.87
19	Limbu		Täi-le	
1A	1.88 Bougui	1.89		
1B	1.90			
1C	1.91			
1D	1.92 Suppl. phonétique			1.93 Suppl diacritiques
1E	1.94 Latin étendu additionnel			
1F	1.95 Grec étendu			

1.96 **Figure 3.** Zone des écritures générales du PMB (U+02000 à U+1FFF)

20	1.99 Ponctuation générale	1.100 Exposants, indices	1.101 Devises	1.102 Sign. comb. symbo.		
21	1.103 Symboles de type lettre	1.104 Formes numériques	1.105 Flèches			
22	1.106 Opérateurs mathématiques					
23	1.107 Signes techniques divers					
24	1.106 Pictogrammes de commande	1.109 R.O.C.	1.106 Alphanumériques cerclés			
25	1.111 Filets	1.112 Pavés	1.113 Formes géométriques			
26	1.114 Symboles divers					
27	1.115 Casseau			1.116		
28	1.117 Combinaisons Braille					
29	Supplément B de flèches		Divers symboles math. B			
2A	1.118 Opérateurs mathématiques supplémentaires					
2B	1.119 Supplément de flèches					
2C	1.120 Glagolitique	1.121 Supp. Latin C	1.122 Copte	1.123		
2D	1.124 Suppl. géorgien	1.125 Tifinaghe	1.126 Suppl. éthiopien	1.127		
2E	1.128 Ponctua. suppl.		1.129 Formes supplémentaires clés CJC			
2F	1.130 Clés chinoises (K'ang-hsi ou Kangxi)		1.131	1.132 Descr. idéog		
30	1.133 Symboles et ponctuation CJC	1.134 Hiragana	1.35 Katakana			
31	1.136 Bopomofo	1.137 Jamos de compatibilité	1.138 Kanbun	1.139 Bopo. 2 Étendu.	1.140 Traits CJC	1.141 Kata-kana
32	1.142 Lettres et mois CJC cerclés					
33	1.143 Compatibilité CJC					

1.144 Figure 4. Zone des écritures générales du PMB (U+2000 à U+33FF)

Le plan multilingue complémentaire (plan 0116, voir la figure 4) est destiné aux systèmes d'écritures non idéographiques et aux symboles similaires à ceux que l'on retrouve dans le PMB, mais contrairement au PMB la plupart des écritures qui y sont codées (et qui y seront codées) ne sont plus utilisées de nos jours. Le Plan idéographique complémentaire (plan 0216) normalise un ensemble de caractères utilisés en Extrême-Orient. Le Plan complémentaire spécialisé (plan 0E16) normalise un ensemble de caractères graphiques à vocation particulière (certains de ces caractères n'ont pas d'apparence visuelle).

La dernière version d'Unicode, la version 4.1, code actuellement 97 655 caractères, la grande majorité dans le PMB. Il reste environ 5 300 positions de code non affectées dans le PMB, elles sont réservées à une future normalisation. Ajoutons que 870 000 positions de code restent disponibles dans les autres plans.

00	1.145 Syllabaire linéaire B		1.146 Idéogrammes du linéaire B	
01	1.147 Nombres égéens	1.148 Anciens nombres grecs		1.149
02	1.150			
03	1.151 Italique	1.152 Gotique	1.153 Ougaritique	1.154 Vieux perse
04	1.156 Déséret	1.157 Shavien	1.158 Osmanya	1.159
05	1.160			
..				
07				
08	1.161 Syllabaire chypriote		1.162	
09	1.163			
0A	1.164 Kharochthî	1.165		1.166
..	1.167			
CF	1.168			
D0	1.169 Symboles musicaux byzantins			
D1	1.171 Symboles musicaux occidentaux			
D2	1.171 Ancienne notation musicale grecque		1.172	
D3	1.173 Symb. du Classique du mystère suprême		1.174	
D4	1.175 Symboles alphanumériques mathématiques			
..	1.176			
D7	1.177			
D8	1.178			
..	1.179			
..	1.180			
FF				

1.181 Figure 5. Plan multilingue complémentaire (PMC)

3. Caractères

3.1. Caractère abstrait, caractère codé et glyphe

Un des défis les plus redoutables dans la conception d'un jeu de caractères universel réside dans la conception souvent différente que se fait chaque langue de ce qui constitue un élément textuel fondamental (une lettre dans le cas des écritures alphabétiques). Ce concept peut même changer, pour une même langue, selon les applications textuelles considérées. Ainsi, en typographie traditionnelle allemande,

la combinaison de lettres « ck » constitue un élément textuel pour ce qui a trait à la coupure de mots (où cette combinaison se transforme en « k-k ») mais pas en ce qui concerne le tri lexicographique. En français, les objets « a » et « A » sont habituellement considérés comme des éléments textuels distincts pour ce qui a trait au rendu, mais on les confond fréquemment, à dessein, lors de recherches textuelles.

Une norme de codage de caractères fournit des unités fondamentales de codage (que l'on nomme caractères abstraits) mises en correspondance biunivoque (de un à un) avec le numéro qu'on leur attribue. Ces numéros de code sont les plus petites unités adressables du texte stocké.

Le graphème constitue une des classes d'éléments textuels importantes, il correspond typiquement à ce que l'utilisateur considère être un caractère. La figure 5 illustre le rapport qui existe entre les caractères abstraits et les graphèmes.

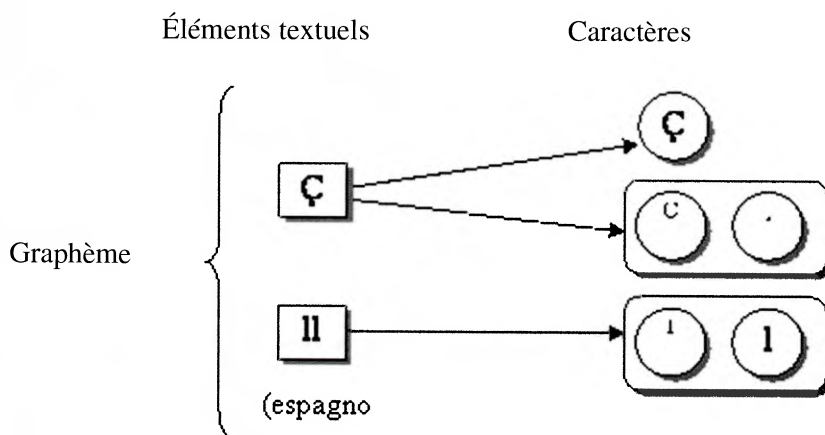


Figure 6. *Éléments textuels et caractères*

Comment passe-t-on des éléments textuels ou des graphèmes aux caractères abstraits d'un jeu de caractères ? Il faut comprendre qu'aucune architecture de jeu de caractères ne peut satisfaire de manière optimale tous les processus textuels (rendu, coupure de lignes, détection de mots, correction orthographique, indexation, synthèse vocale, sélection et mise en évidence à l'écran, édition, comparaison, tri, compression, etc.). L'ensemble des caractères abstraits choisis forme donc un compromis entre les besoins des différents traitements textuels pour un ensemble de langues utilisant la même écriture. En effet, pour des raisons d'efficacité et d'échange de données, Unicode et l'ISO/CEI 10646 ne codent pas nécessairement les graphèmes d'une langue particulière mais des caractères abstraits correspondant aux caractères communs à une écriture (pouvant servir à plusieurs langues).

En résumé, on dira qu'un caractère abstrait est une unité d'information utilisée pour organiser, commander ou représenter des données textuelles. Le caractère abstrait n'a pas de forme concrète (il ne faut pas le confondre avec ses glyphes) et ne correspond pas nécessairement à ce que l'utilisateur imagine être un « caractère » dans sa langue (un graphème). Le glyphe⁽⁸⁾ représente les différentes formes qu'un caractère abstrait peut prendre (voir la figure 6). Un caractère codé, pour sa part, est l'association entre un caractère abstrait et son numéro. Notons que certains caractères abstraits sont codés plus d'une fois dans Unicode⁽⁹⁾.

Glyphes (œils)	Caractères Unicode/ ISO 10646
A A A A A <i>A</i>	U+0041 lettre majuscule latine a ⁽¹⁰⁾
f fi ffi ffi ffi ffi	U+0066 lettre minuscule latine f +
f fi ffi ffi ffi ffi	U+0066 lettre minuscule latine f +
f fi ffi ffi ffi ffi	U+0069 lettre minuscule latine i
ف ف	U+0647 lettre arabe fa'

Figure 7. *Caractères et glyphes*

3.2. Caractères combinatoires et diacritiques

On nomme caractères combinatoires les caractères destinés à s'afficher en association avec un caractère de base. Si l'on décide de représenter « ö » à l'aide d'un « o » (U+004F) suivi d'un caractère « ¨ » (U+0302), « o » est le caractère de base et « ¨ » est un caractère combinatoire. Pour bien indiquer qu'il s'agit d'un caractère combinatoire, on le représente surmontant un cercle pointillé : C. Lors du rendu, les glyphes qui représentent ces caractères seront dessinés dans un certain ordre près du glyphe qui représente le caractère de base. Le standard Unicode distingue deux types de signes combinatoires : ceux avec chasse et ceux sans. Les

-
- (8) - On oppose parfois l'image de glyphe, dessin concret et invariant (sauf, sans que ce soit explicite ni toujours vrai, pour un facteur d'échelle) au glyphe plus abstrait, qui fait abstraction des variations typographiques (autre que le facteur d'échelle) mais pas des variantes de forme contextuelles comme en arabe. Quand on voudra éviter la polysémie du mot glyphe, on utilisera le terme traditionnel typographique œil (pluriel des œils) pour désigner sans équivoque l'image de glyphe. La lettre arabe sîn (un caractère abstrait) peut être représentée, selon le contexte, par un de ses quatre glyphes archétypiques, eux-mêmes représentés concrètement par des œils fruits du travail soigné d'un fondeur.
- (9) - C'est le cas des équivalents de compatibilité avec d'autres jeux de caractères ou des rares doublons (dans le cas du coréen, ils sont reproduits sciemment à des fins de compatibilité).
- (10) - On représente le numéro des caractères Unicode à l'aide de la notation U+xxxx (où xxxx est la valeur hexadécimale d'un caractère codé du PMB) ou U+yyyyy (où yyyyy est la valeur hexadécimale d'un caractère codé dans un plan complémentaire). La forme à cinq chiffres (U+zzzzz) semble exclue par Unicode.

caractères combinatoires sans chasse (encore appelés à chasse nulle) n'occupent pas de place par eux-mêmes. Toutefois, la combinaison du caractère de base et du caractère à chasse nulle peut occuper au rendu plus d'espace latéral que le caractère de base seul. Ainsi, un « î » chasse légèrement plus qu'un simple « i ».

Les diacritiques constituent la principale classe des caractères combinatoires sans chasse utilisés dans les alphabets européens. Dans le standard Unicode, la définition du terme « diacritique » est relativement vague et comprend à la fois les accents ainsi que d'autres signes sans chasse. Tout caractère de base peut se voir adjoindre n'importe quel diacritique quelle que soit l'écriture de ce caractère. On peut donc théoriquement adjoindre un « ^ » à une lettre arabe.

3.3. Suite de caractères de base et diacritiques

Tous les caractères combinatoires doivent suivre le caractère de base qu'ils qualifient. La suite de caractères Unicode U+0075 LETTRE MINUSCULE LATINE U + U+0308 C DIACRITIQUE TREMA + U+0065 LETTRE MINUSCULE LATINE E représente sans équivoque « üe » et non « uë ». La convention d'ordonnement utilisée par le standard Unicode correspond à l'ordre logique des diacritiques arabes et indiens dont la grande majorité suit (au clavier et dans la chaîne parlée) les caractères de base par rapport auxquels ils se placent.

Dans les tableaux consacrés aux écritures de l'Inde, certaines voyelles sont représentées à la gauche d'un cercle pointillé dans les tableaux de caractères (cf. figure 7). Il faut distinguer attentivement ce cas spécial de celui des diacritiques habituels. Ces signes-voyelles doivent être rendus à la gauche de la consonne ou du groupe consonantique même si logiquement (en mémoire) ils suivent la consonne dans leur représentation Unicode. On dit que ces signes-voyelles sont antéposés.

p + I → T

Figure 8. *Signe vocalique indien antéposé*
(i bref)

3.4. Caractères combinatoires multiples

Il arrive que plusieurs diacritiques s'adjoignent à un seul caractère (cf. figure 8). Le standard Unicode ne limite pas le nombre de caractères combinatoires qui peut suivre un caractère de base.



Figure 9. *Empilement de diacritiques*

Si des caractères combinatoires peuvent affecter une même zone typographique - par exemple un U+0308 C DIACRITIQUE TRÉMA et un U+0301 D DIACRITIQUE ACCENT AIGU - l'ordre des caractères codés détermine l'ordre d'affichage (cf. figure 9). Les diacritiques sont disposés à tour de rôle à partir du glyphe correspondant au caractère de base pour ensuite s'en éloigner. Les caractères combinatoires placés au-dessus d'un caractère de base s'empilent verticalement, en commençant par le premier dans l'ordre de stockage logique. On poursuit l'empilement vers le haut d'autant de signes qu'il est nécessaire selon le nombre de codes de caractère qui suivent le caractère de base. La situation est inversée pour les caractères combinatoires placés sous un caractère de base : les caractères combinatoires s'empilent vers le bas. On appelle cet ordre l'ordre centrifuge.

J	lettre minuscule e accent circonflexe lettre minuscule e + accent circonflexe
L	lettre minuscule e + diacritique point en chef
M	lettre minuscule e accent circonflexe + diacritique point souscrit lettre minuscule e + accent circonflexe + diacritique point souscrit lettre minuscule e + diacritique point souscrit + accent circonflexe
P	lettre minuscule e + diacritique point en chef + diacritique point souscrit lettre minuscule e + diacritique point souscrit + diacritique point en chef
Q	lettre minuscule e tréma + diacritique accent aigu lettre minuscule e + diacritique tréma + diacritique accent aigu
R	lettre minuscule e accent aigu + diacritique tréma lettre minuscule e + diacritique accent aigu + diacritique tréma

Figure 10. *Interaction des caractères diacritiques*

Certains caractères dérogent à l'empilement centrifuge. C'est le cas du grec, lorsqu'un esprit précède un accent aigu ou grave, les deux diacritiques doivent alors s'écrire côte à côte au-dessus de la lettre de base. Ici l'ordre des caractères est code du caractère de base + code de l'esprit + code de l'accent. Cet exemple démontre bien que la disposition correcte des signes diacritiques dépend de l'écriture employée.

4. Principes de conception du standard Unicode

Le standard Unicode énumère dix principes qui ont guidé son élaboration. Pour un jeu de caractères aussi vaste qu'Unicode, tous ces principes ne s'appliquent pas toujours de manière égale. Parfois, l'adhésion à un principe entraîne la violation d'un des autres principes. Unicode est le résultat d'un délicat équilibre entre différents besoins parfois contradictoires.

UNIVERSALITÉ - Le standard Unicode fournit un répertoire unique et universel.

EFFICACITÉ - L'analyse et le traitement d'un texte Unicode sont simples.

CARACTÈRES ET NON GLYPHES - Le standard Unicode définit des caractères et non des glyphes.

SÉMANTIQUE - Les caractères ont une sémantique bien définie.

TEXTE BRUT - Le standard Unicode normalise du texte brut.

ORDRE LOGIQUE - La représentation implicite en mémoire correspond à l'ordre logique.

UNIFICATION - Le standard Unicode unifie les caractères identiques à l'intérieur des écritures et non à l'intérieur d'un même alphabet associé à une seule langue.

COMPOSITION DYNAMIQUE - On compose les formes à diacritique(s) dynamiquement.

SÉQUENCE ÉQUIVALENTE - À chaque forme statique précomposée correspond au moins une suite équivalente de caractères décomposés que l'on pourra recomposer dynamiquement.

CONVERTIBILITÉ - Unicode définit des conversions exactes entre son répertoire et d'autres normes largement répandues.

4.1. Universalité⁽¹¹⁾

Le standard Unicode décrit un seul jeu de caractères de très grande taille qui comprend tous les caractères nécessaires aux écritures du monde. Le passage d'une écriture à l'autre ne nécessite pas, contrairement à certains jeux de caractères d'Extrême-Orient, de caractère d'échappement. Les caractères de toutes les écritures peuvent se mêler sans aucun encombre (voir la figure 10).

(11) - Ce principe a remplacé dans la version 4.0 le premier principe d'Unicode encore officiel dans la version 3.2 d'Unicode : « Unicode est un code à 16 bits ». Ce principe est caduc depuis l'introduction d'UTF-32. Unicode peut se représenter de manière aussi légitime sous trois formes stockées différentes (UTF-8, UTF-16 ou UTF-32).

Glyphe représentatif	Numéro	Nom officiel français de l'ISO/CEI 10646
D	U+0044	LETTRE MAJUSCULE LATINE D
v	U+0076	LETTRE MINUSCULE LATINE V
o	U+006F	LETTRE MINUSCULE LATINE O
ř	U+0159	LETTRE MINUSCULE LATINE R CARON
a	U+0061	LETTRE MINUSCULE LATINE A
k	U+006B	LETTRE MINUSCULE LATINE K
	U+0020	ESPACE
天	U+5929	IDÉOGRAMME UNIFIÉ CJC-5929
心	U+2F3C	CLÉ CHINOISE CŒUR
س	U+0633	LETTRE ARABE SÎN
ل	U+0644	LETTRE ARABE LAM
ا	U+0627	LETTRE ARABE ALIF
م	U+0645	LETTRE ARABE MÎM
α	U+03B1	LETTRE MINUSCULE GRECQUE ALPHA
ψ	U+10338	LETTRE GOTIQUE TH
♩	U+1D110	SYMBOLE MUSICAL POINT D'ORGUE
₯	U+2270	NI PLUS PETIT NI ÉGAL À
U	U+0F06	SIGNE TIBÉTAIN D'INSERTION YIG MGO PHUR SHAD MA ⁽¹²⁾

Figure 11. *Universalité du jeu de caractères*

4.2. Efficacité

Afin de permettre des mises en œuvre efficaces, Unicode n'inclut pas de caractères d'échappement ou d'états de passage. Tous les caractères Unicode ont la même qualité, tous les codes sont d'une égale facilité d'accès⁽¹³⁾. Par convention et

(12) - Il s'agit d'une translittération (« étymologique ») d'origine anglaise du tibétain, la transcription française (et donc sa prononciation approximative) est fournie par son synonyme non normatif défini par l'ISO/CEI 10646: yik go p'our ché ma.

(13) - Comme on le verra par la suite (v. l'article du Pr Lahbib Zenkour), ceci dépend de la forme stockée choisie. Ainsi les caractères stockés à l'aide d'UTF-8 et UTF-16 ne sont-ils pas aussi faciles à accéder que les caractères stockés sous la forme UTF-32.

dans la mesure du possible, les caractères d'une même écriture sont regroupés en un bloc de caractères consécutifs. Non seulement ceci facilite-t-il la consultation des tableaux de caractères, mais cela permet également des mises en œuvre plus compactes. La majorité des caractères de ponctuation communs aux différentes écritures sont regroupés dans un bloc séparé.

Cette universalité et cette uniformité de codage permettent une analyse, un tri, un affichage, un repérage et une édition efficaces des chaînes textuelles Unicode.

4.3. Caractères et non glyphes

Le standard Unicode distingue les caractères, qui forment les plus petites unités distinctives et significatives d'une langue écrite, des glyphes qui représentent les différentes formes visuelles qu'un caractère peut prendre. Un seul glyphe peut représenter un seul ou plusieurs caractères (on parle alors d'une ligature). De même, de multiples glyphes peuvent représenter un seul caractère (b, b et b sont tous des U+0062 b).

On représente un caractère par un numéro qui ne réside qu'en mémoire ou sur disque. Le standard Unicode ne s'intéresse qu'au codage de caractères et non aux glyphes. Contrairement aux caractères, les glyphes apparaissent à l'écran ou sur papier et représentent une forme particulière d'un ou plusieurs caractères. Un répertoire de glyphes constitue une police. La forme des glyphes et les méthodes d'identification et de sélection de glyphes sont la responsabilité des fonderies et des normes appropriées. Celles-ci ne font pas partie du standard Unicode.

Pour certaines écritures, comme l'arabe et différentes écritures du sous-continent indien, le nombre de glyphes nécessaires à l'affichage peut être nettement plus important que le nombre des caractères qui codent les unités de base de cette écriture. Le nombre de glyphes peut également dépendre du style calligraphique de la police. Ainsi, une police arabe de style nastaliq peut comprendre plusieurs milliers de glyphes. Toutefois, quelle que soit la police choisie, un texte arabe est toujours représenté par la même cinquantaine de caractères codés (la même cinquantaine de numéros).

4.4. Sémantique

Les caractères sont munis d'une sémantique bien définie. Le standard Unicode fournit des tableaux de propriétés de caractère que l'on peut utiliser pour le tri, l'analyse ou d'autres algorithmes qui ont besoin d'une connaissance sémantique des caractères traités. Voici quelques-unes des propriétés de caractère définies par Unicode :

- la casse⁽¹⁴⁾ du caractère : majuscule A, minuscule a, casse de titre Dz ;
- sa directionalité : gauche, droite, faible, neutre ;
- s'il est à symétrie miroir : < ∫ → (à ajuster selon directionalité du texte) ;
- s'il s'agit d'un numéral : iv 7 ;
- s'il est combinatoire : O C I A .

4.5. Texte brut

Un texte brut est constitué d'une suite de codes de caractères, dont aucun ne représente du balisage. Un texte brut Unicode est donc une suite simple de codes de caractères Unicode. À l'inverse, un texte enrichi, également connu sous le nom de texte de fantaisie, est constitué de texte brut et d'un ensemble d'informations comme la force de corps employée, la couleur du texte, des hyperliens, etc. Le texte de cet article, par exemple, est un texte formaté à l'aide de plusieurs polices, il s'agit d'un texte enrichi.

La simplicité du texte brut lui permet de servir de base aux textes enrichis. SGML, HTML, XML ou TEX sont des textes enrichis représentés à l'aide de textes bruts émaillés de suites de caractères qui représentent des structures de données supplémentaires (ici des balises). On définit parfois le texte brut de la façon suivante : un texte brut doit comprendre suffisamment d'information pour que le texte puisse être affiché de manière lisible, sans plus. Enfin, notons que, si le texte brut est public, normalisé et lisible par toute, la représentation du texte enrichi est souvent propriétaire ou peut varier selon les mises en œuvre.

4.6. Ordre logique

Les textes Unicode, quelle que soit l'écriture, sont stockés en mémoire en ordre logique. Cet ordre correspond grosso modo à l'ordre dans lequel le texte est saisi au clavier. Parfois, l'ordre des caractères à l'affichage ou à l'impression diffère de l'ordre logique, c'est le cas des écritures bidirectionnelles comme l'arabe ou l'hébreu. La figure 11 illustre la différence entre les ordres logique et d'affichage.

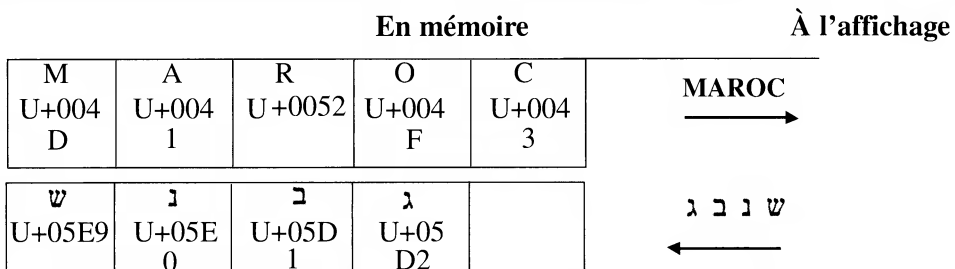


Figure 12. Ordre logique

(14) - Casse, dans ce sens générique, est un néologisme puisque la casse se référait uniquement à l'origine à la boîte contenant tous les caractères d'imprimerie. On parle, toutefois, depuis longtemps de bas de casse pour désigner une minuscule (v. Le Robert).

À l’affichage, le glyphe qui représente le premier caractère du texte latin de la figure 11 s’affiche à gauche. En revanche, le premier caractère logique du texte hébreu est représenté par le glyphe hébreu le plus proche de la marge droite. Les glyphes hébreux suivants sont disposés vers la gauche.

On utilise l’ordre logique même lorsque des caractères de différentes directions dominantes se mêlent : écritures gauche-à-droite (grec, cyrillique, latin), droite-à-gauche (arabe, hébreu) ou même verticale. Les propriétés de directionnalité implicites aux caractères suffisent généralement à déterminer l’ordre correct d’affichage. Toutefois, il arrive parfois que cette directionnalité implicite s’avère insuffisante pour afficher le texte de manière lisible. Unicode prévoit des caractères de commande qui permettent d’influencer l’ordre d’affichage.

Rappelons que certains caractères, comme l’i bref dévanâgarî (I), s’affichent avant les caractères qu’ils suivent logiquement en mémoire. Les signes combinatoires (comme les accents en français, en grec, en cyrillique, les points voyelles arabes, les signes voyelles dévanâgarî, etc.) n’apparaissent pas de manière linéaire dans le texte rendu. Dans une chaîne de caractères Unicode, tous ces signes suivent le caractère de base qu’ils modifient, contrairement à ce qui fait généralement avec les machines à écrire. Ainsi, un « b » latin est-il stocké sous la forme d’un « b » suivi du signe combinatoire (ou diacritique) « C ».

4.7. Unification

Le standard Unicode évite de coder plusieurs fois le même caractère et n’attribue qu’un seul numéro de code à un caractère d’une écriture donnée, quel que soit le nombre de langues qui utilise ce caractère (le a de chat, Katze, gato et kat est donc codé à l’aide d’un seul numéro : U+0061). Il en va de même pour les idéogrammes chinois/japonais/coréens (CJC).

Le standard Unicode ne tente pas de coder des caractéristiques de texte comme la langue, la police, la force de corps, l’emplacement, l’œil des caractères (glyphes), etc. Ainsi, ne conserve-t-il pas d’information sur la langue lors du codage de caractères : l’i grec français, l’ypsilon allemand et le wye anglais sont tous représentés par le même code de caractère, U+0057 « Y », il en va de même pour le zi (tseu) chinois, le ji japonais et le ja coréen représentés par le même code de caractère, U+5B57 丿, car ils partagent origine et sens et ne connaissent que des différences d’œil anodines.

Toutefois un autre principe (voir _4.10, Convertibilité) s’oppose parfois à l’unification des caractères. Ainsi, si un des jeux de caractères fondateurs d’Unicode distingue deux caractères d’apparence similaire ou identique, Unicode maintient cette distinction et code deux fois le caractère. C’est le cas en latin pour le U+00C5 Å et U+212B Å. C’est également le cas pour les caractères han (丿, 丿, 丿, 丿, 丿) qui désignent tous l’épée ou le poignard et sont de structure identique mais

qui sont codés séparément dans le jeu de caractères japonais source JIS X 0208-1990 ; Unicode les distingue donc pour assurer une conversion aller-retour sans perte vers ce jeu de caractères.

4.8. Composition dynamique

Le standard Unicode permet de composer dynamiquement les formes accentuées des lettres et des syllabes hangûl. Ce processus est extrêmement productif, car il est ouvert. On peut donc créer de nouvelles formes à l'aide d'une combinaison de caractères de base⁽¹⁵⁾ suivis d'un ou plusieurs diacritiques. Ainsi, le tréma « C » peut-il s'adjoindre à des voyelles ou à des consonnes dans des langues qui utilisent l'écriture latine ainsi que dans d'autres écritures. Dans les écritures comme l'arabe ou l'hébreu, les diacritiques représentent souvent des voyelles. Les écritures de l'Inde comportent de nombreux caractères combinatoires dont les glyphes suivent habituellement, à la même hauteur sur la ligne, le caractère de base.

4.9. Séquence équivalente

Certains éléments textuels peuvent être codés de plus d'une façon : une forme précomposée ou une ou plusieurs compositions dynamiques. Pour des raisons de compatibilité avec les jeux de caractères actuels, on a inclus dans le standard de nombreuses formes précomposées. Les caractères précomposés sont équivalents à leur suite de caractères décomposée. Ainsi « ä » est-il équivalent à « a » + « C ». Unicode fournit pour chaque forme précomposée incluse dans la norme son équivalent canonique décomposé. Ces suites sont fournies dans la base de données Unicode et dans les tableaux de caractères (v. 5.5, Décompositions).

$$\begin{array}{l}
 \text{LJ} + \text{U} \equiv \text{L} + \text{J} + \text{U} \\
 \text{A} + \text{C} + \text{O} \equiv \text{A} + \text{O} + \text{C} \\
 \text{A} + \text{C} + \text{D} \neq \text{A} + \text{D} + \text{C} \\
 \text{E} + \text{O} + \text{E} \neq \text{E} + \text{E} + \text{O}
 \end{array}$$

Figure 13. *Séquences équivalentes et distinctes*

Des suites de diacritiques qui interagissent au niveau typographique (par exemple, deux accents suscrits ou deux diacritiques souscrits) sont considérées distinctes si les diacritiques se présentent dans des ordres différents. Unicode attribue pour chaque caractère une classe combinatoire et définit, à l'aide ces classes combinatoires, un algorithme de mise en ordre canonique des suites de

(15) - Par exemple, dans le cas des diacritiques doubles qui recouvrent les deux caractères de base qui précèdent.

signes combinatoires. Cet ordre canonique permet de trier les séquences équivalentes de caractères combinatoires dans un ordre prévisible. Les caractères de base sont de classe combinatoire zéro. Tous les diacritiques qui interagissent l'un sur l'autre au niveau typographique sont de même classe combinatoire, des suites d'ordre différent les contenant ne sont donc pas équivalentes.

	Original			Décomposition				Ordre canonique								
Chaîne	Ä	+	O	→	A	+	C	+	O	→	A	+	O	+	C	
Classe combi.	0		220		0		230		220		0		220		230	
Chaîne	A	+	O	+	C	→					→	A	+	O	+	C
Classe combi.	0		220		230							0		220		230

Figure 14. Classes combinatoires et mise en ordre canonique

4.10. Convertibilité

Le standard Unicode et l'ISO/CEI 10646 formaient à leur naissance un jeu de caractères complètement neuf. Il a donc fallu définir des méthodes d'importation et d'exportation de données codées dans des jeux de caractères fondamentaux à l'époque⁽¹⁶⁾. Un grand soin a été apporté afin de garantir une conversion aller-retour sans perte entre Unicode et ces jeux de caractères. C'est pourquoi, lorsque des variantes de forme (ou même la même forme) se sont vues affectées des numéros de caractère différents dans une norme de base, Unicode les garde séparées.

En général, toute valeur de code dans une autre norme correspond à une valeur de code en Unicode. Il arrive, cependant, qu'une valeur de code dans une autre norme correspondra à une suite de valeur de code Unicode ou vice-versa.

5. Le standard Unicode : mode d'emploi

Le standard Unicode est un ouvrage monumental, il répertorie en effet plus de 95 000 caractères et définit les règles qui les régissent. Cet inventaire planétaire, fruit de plus d'une décennie de travail international, se présente sous la forme d'une longue liste. Cette liste de plus de 700 pages est annotée et elle est précédée de 15 chapitres. Les premiers chapitres sont normatifs et décrivent des aspects communs à toutes les écritures (et résumés dans cet article), ils sont suivis de chapitres «introductifs» essentiels qui portent, chacun, sur une famille d'écritures particulières.

(16) - Unicode ne prévoit de conversions que dans le cas de jeux de caractères élaborés avant mai 1993.

Les tableaux du standard Unicode représentent les caractères de la norme ISO/CEI 10646 et du standard Unicode. Les caractères sont regroupés en blocs apparentés. Habituellement, ces blocs comprennent des caractères provenant d'une seule écriture. Un bloc de caractères suffit généralement à représenter une écriture. Il existe, toutefois, des exceptions notables, plus particulièrement dans le domaine des caractères de ponctuation.

Une liste de noms de caractère suit chaque tableau de codes⁽¹⁷⁾, à l'exception des idéophonogrammes CJC et des syllabes hangûl dont un algorithme définit les noms. La liste de noms de caractère reprend tous les caractères du bloc et fournit le plus souvent des renseignements complémentaires sur ceux-ci.

5.1. Liste des noms de caractère

L'exemple suivant illustre les différentes parties de chaque entrée de la liste de noms de caractère.

<i>Code</i>	<i>Glyphe</i>	<i>Description</i>	
00A E	®	SYMBOLE MARQUE DÉPOSÉE	(nom ISO 10646)
00AF F	-	MACRON = barre supérieure APL •ce caractère chasse ' 02c9- lettre modificative macron ' 0304 -diacritique macron ' 0305-diacritique tiret haut	(nom ISO 10646) (nom optionnel) (renseignement) (renvois)
		≈ 0020 , 0304-	(décomposition de compatibilité)
00E5	å	LETTRE MINUSCULE LATINE A ROND EN CHEF •danois, norvégien, suédois, wallon	
		≡ 0061 a 030A	(décomposition canonique)

(17) - La liste intégrale des noms de caractère est disponible sous forme de texte brut à l'adresse suivante : <
<http://hapax.qc.ca/ListeDesNoms-4.1.0.txt>>

5.2. Images dans les tableaux de codes et dans les listes de caractères

Chaque caractère dans ces tableaux de codes est représenté à l'aide d'un glyphe représentatif. Ce glyphe n'a pas de valeur normative, il permet simplement à un utilisateur averti de reconnaître le caractère visé ou de retrouver facilement le caractère dans les tableaux. Dans de nombreux cas, il existe des représentations concurrentes, plus ou moins établies, pour un même caractère.

Les concepteurs de polices de haute qualité devront effectuer leur propre recherche pour déterminer l'apparence la plus appropriée des caractères Unicode. En outre, de nombreuses écritures nécessitent des formes de glyphes différentes selon le contexte, un positionnement contextuel du glyphe ou la formation de ligatures. Aucune de ces informations n'est illustrée dans les tableaux de codes.

Pour les écritures non européennes, on a choisi comme styles de caractère ceux dont les œils se différencient le plus les uns des autres.

5.3. Renvois

Les caractères de renvoi (précédés de ') désignent plusieurs types de renvoi : l'inégalité explicite, les autres casses du même caractère ou d'autres rapports linguistiques.

Inégalité explicite. Les deux caractères ne sont pas identiques bien que leurs glyphes soient identiques ou fort similaires.

003A	:	DEUX-POINTS ' 0589 : point arménien ' 2236 : rapport
------	---	--

Autres rapports linguistiques. Parmi ces rapports, on compte les translittérations (entre le serbe et le croate, par exemple), des caractères sans rapport typographique mais qui servent à représenter le même son, etc.

01C9	lj	LETTRE MINUSCULE LATINE LJ = digramme soudé lj ' 0459 _ lettre minuscule cyrillique lié ≈ 006C l 006A j
------	----	--

5.4. Renseignements sur les langues

Quand cela peut-être utile, on trouve parfois une note informative identifiant les langues qui utilisent ce caractère. Pour les écritures bicamérales, cette information n'est fournie que pour les lettres de bas de casse afin d'éviter une répétition inutile.

Les points de suspension «...» indiquent que la liste des langues n'est pas limitative et qu'elle ne reprend que les langues principales.

5.5. Décompositions

La séquence de décomposition d'un caractère (constituée d'une ou plusieurs lettres) peut être de deux types : canonique ou de compatibilité. La correspondance canonique est indiquée à l'aide d'un symbole identique à ≡.

00E5	å	LETTRE MINUSCULE LATINE A ROND EN CHEF •danois, norvégien, suédois, wallon ≡ 0061 a 030A°
212B	Å	SYMBOLE ANGSTRÖM ≡ 00C5 Å lettre majuscule latine a rond en chef

Les correspondances ne fournissent pas des décompositions complètes, c'est l'application récursive des correspondances qui fournit la décomposition canonique (après sa mise en ordre canonique).

1EA	–	LETTRE MINUSCULE LATINE A ACCENT CIRCONFLEXE ET ACCENT AIGU ≡ 00E2 â 0301
00E2	â	LETTRE MINUSCULE LATINE A ACCENT CIRCONFLEXE ≡ 0061 a 0302

La correspondance de compatibilité s'indique à l'aide d'un signe presque égal à ≈ . Une balise de formatage peut accompagner la décomposition.

01F3	dz	LETTRE MINUSCULE LATINE DZ ≈ 0064 d 007A z
FF25	A	LETTRE MAJUSCULE LATINE E PLEINE CHASSE ≈ <large> 0045 E

6. Ce qu'Unicode n'est pas

On pense souvent qu'Unicode est un produit. Rien n'est plus faux. Unicode (ou l'ISO/CEI 10646) n'est ni un logiciel ni une police.

Unicode est un jeu de caractères qui définit un numéro unique pour chaque caractère, quelle que soit la plate-forme, quel que soit le logiciel, quelle que soit la langue. Unicode précise également des propriétés pour ces caractères afin d'en préciser la nature et le traitement.

La mise en œuvre d'Unicode seul n'est pas complexe, tout dépend de la langue que vous voulez prendre en charge. En effet, la conformité à Unicode peut s'énoncer sommairement et de manière officieuse comme suit : on peut ignorer un caractère qu'on ne comprend pas, mais pas le corrompre ; la mise en œuvre d'un sous-ensemble ne dépend que de vous. Ainsi, si vous ne voulez prendre en charge que les écritures de l'Europe occidentale, vous (ou votre système) n'aurez guère plus à mettre en œuvre que ce que vous avez déjà en ASCII.

Bibliographie

Le consortium Unicode, The Unicode Standard, Version 4.0, Reading, Addison-Wesley, 2003.

Patrick Andries, Traduction et annotation du standard Unicode 3.0, Longueuil, 2002, <<http://hapax.qc.ca>>.

ISO/CEI 10646:2003, Jeu universel de caractères codés sur plusieurs octets (JUC), Genève, 2005.

Propriétés et Formats d'Unicode

Lahbib Zenkouar

1-Unicode/UCS

Le code de caractères universel Unicode/UCS destiné à remplacer progressivement le code ASCII est mis en pratique. En effet, le code ASCII, est spécifiquement américain et a été le standard de codage des caractères pour plus de 30 ans. Conçu spécifiquement pour la langue anglaise, il a posé un certain nombre de problèmes de représentation de caractères en informatique aux autres langues. Mais au début de l'ère informatique, on pensait plutôt à la réussite de l'informatique en tant qu'outil technologique plutôt que celui d'un moyen de communication multilingue intégrant au mieux les droits fondamentaux des langues et écritures [1],[2].

Au tout début des années 1990, deux initiatives de remplacement du code ASCII par un nouveau code universel de caractères ont été lancées, l'une par l'ISO intitulée UCS (Universal Character Set) et l'autre par un consortium de fabricants, Unicode. Les deux entités se sont entendues pour mener cette entreprise en commun depuis 1993. Pour être plus précis, la définition d'Unicode est calquée sur la norme ISO/CEI 10646, UCS-2, c'est à dire la norme ISO/IEC 10646-1:1993

Le standard Unicode se définit comme un système de codage mettant en œuvre un mécanisme cohérent et universel de codage de caractères permettant ainsi aux textes multilingues de coexister. Ce système facilite l'échange de données textuelles à l'échelle planétaire et permet d'engendrer et de créer une informatique à l'échelle internationale. C'est un point positif qui montre si besoin est, l'ascension encore exponentielle de l'informatique.

Le standard Unicode est devenu le codage supportant HTML et XML. IL constitue de ce fait le codage utilisé pour l'Internet et les nouveaux protocoles relatifs aux technologies de l'information et de la communication tels que normalisés et standardisés par le consortium W3C.

Grâce à ce nouveau standard Unicode, l'industrie informatique assurera la stabilité des caractères, ce qui aura pour effet d'augmenter l'interopérabilité et l'échange de données au niveau mondial. Enfin, le système unicode simplifie le développement de logiciels et leur internationalisation. Il en réduit les coûts par ce

système de codage universel. Il faut quand même signaler que bien qu'il ait été normalisé depuis huit ans, le code UCS/Unicode est seulement depuis peu réellement mis en œuvre [3],[4].

Dans cet article décrivant les propriétés et les formats de stockage et de transfert utilisés par le système de codage unicode, je commencerai par décrire le modèle de codage des caractères, ensuite je décrirai les propriétés des caractères et finalement les formes normalisées des caractères dans ce système de codage.

1. Le modèle de codage des caractères

Le modèle de codage s'appuie autant que faire se peut sur des définitions aussi précises que possible des lettres des écritures de notre monde; en ce sens que chaque notion qui se révèle imprécise n'est plus utilisée malgré son maintien dans les tableaux de codage afin d'éviter toute ambiguïté dans les formes de codage qui s'ensuivront. Les possibilités d'amendement aux différentes propositions soumises aux instances de l'ISO/CEI permettent de pallier aux différents problèmes pour que chaque langue trouve son compte sur cet outil.

Cet organisme de normalisation a instauré une procédure d'examen des propositions à plusieurs niveaux. Cette organisation rend un grand service à la mise au point définitive d'un projet d'amendement. De sorte que la présence d'éventuelles erreurs restent relativement peu fréquente.

Pour ce système, nous définissons d'abord les notions suivantes [4], [5]:

Le répertoire : C'est un ensemble de caractères et non de glyphes. Il peut ne pas être ordonné. En plus, le système de codage Unicode maintient ce répertoire ouvert pour tout nouvel examen de ce même répertoire.

Et le jeu de caractères codés : Ce répertoire devient un jeu de caractères codés dès le moment où on établit une correspondance entre l'ensemble des caractères abstraits du répertoire et un ensemble d'entiers non négatifs pour définir le répertoire de manière numérique. Bien entendu, ce dernier ensemble peut ne pas être contigu.

En définitive et de manière générale, nous dirons qu'un caractère abstrait est codé dans un jeu de caractères donné si un numéro de caractère existe pour ce caractère.

La forme en mémoire

C'est la forme des caractères stockés en mémoire. Cette forme naturelle en mémoire s'appuie sur les unités de stockage. L'unité de stockage est un nombre binaire d'une certaine largeur ou nombre de bits (octet, seize,...) qui sert d'unité de base à l'expression des numéros de caractère en mémoire de l'ordinateur.

Le système Unicode utilise la norme à 16 bits de référence dans l'industrie de l'internationalisation ; il s'agit d'un code de longueur fixe pouvant représenter toutes les langues modernes. La forme UCS-2 est la forme 16 bits de l'ISO 10646 qui est équivalente à celle du consortium Unicode définie sur 16 bits.

Les deux formes d'encodage utilisé dans ce système qui se veut multilingue sont:

UCS-n : la forme à n bits de l'ISO/IEC 10646 qui traduit la forme naturelle de représentation binaire de l'information en mémoire et UTF-n : UCS Transformation Format à n bits, un code d'échange et de transfert défini pour envoyer de l'ISO 10646 et qui permet d'exploiter la forme octale utilisée par la plupart des systèmes actuels.

Par conséquent, les données Unicode peuvent être codées sous deux formes : une forme naturelle de 16 bits et une forme de 8 bits (UTF-8) conçue pour faciliter une utilisation sur les systèmes ASCII préexistants. L'intérêt de l'UTF-8 est sa capacité à préserver les standards existants. Cette forme UTF-8 d'encodage s'impose de plus en plus ; elle est utilisée par Java, Netscape, oracle, Tango (Alis), etc.

Sa principale caractéristique est d'utiliser une méthode d'encodage variable, entre 1 à 4 octets. Les premiers octets constituent l'image de l'ensemble US-ASCII (8 bits). Le premier octet du caractère définit la séquence à laquelle il appartient. Si cet octet commence par 0, il définit alors l'encodage US-ASCII et sa longueur est 1. Si les n (n > 1) premiers bits sont égaux à 1, le n+1 étant égal à 0, alors la longueur du caractère sera n. Les bits restant définissent la séquence d'encodage.

Ces méthodes d'encodage conservent la taille d'encodage pour chaque caractère, ce qui est fondamentalement différent des anciennes méthodes, qui utilisent, par exemple des balises de contrôle pour passer d'un code de caractères à un autre et qui constituent des variantes de l'ASCII.

Ceci peut avoir son importance lorsque la taille des documents devient importante. Un texte en latin encodé en unicode est deux fois plus grand que le même texte encodé en ASCII. En revanche, l'encodage en unicode d'un texte japonais produit sensiblement la même quantité d'octets [3].

En définitive, nous pouvons résumer les formes d'encodage UCS, par le fait qu'elles permettent de coder tous les caractères de toutes les langues. Elles ont été normalisées en 1993 sous la référence ISO 10646:1993, en 2000 sous la référence ISO 10646:2000, en 2001 sous la référence ISO 10646-2 :2001 et en 2003 sous la référence ISO 10646 :2003 qui fut l'objet de l'amendement introduisant l'alphabet amazighe dans la rangée 2D du plan multilingue de base.

Les références correspondantes chez Unicode sont Unicode 1.0 et Unicode 3.0. UCS se compose essentiellement de tables de codes et de noms de caractères, tandis

qu'Unicode y ajoute des précisions et des recommandations d'ordre typographique.

Selon les mises en oeuvre, chaque caractère est exprimé au format UCS-2 (deux octets, BMP) ou au format UCS-4 (quatre octets, code complet). Lorsqu'une mise en oeuvre se heurte à une incompatibilité, la méthode de transformation UTF-8, vue précédemment, est disponible. Nous citons dans ce qui suit toutes les formes d'encodage utilisées avec quelques exemples pour bien illustrer nos propos.

Formes UCS-4 et UTF-32

Les formes UCS-4 et UTF-32 sont équivalentes. Dans ces deux formes, le numéro de caractère est représenté par 32 bits. La forme UTF-32 stocke tous les codes points sur quatre octets. Étant donné que l'espace d'encodage du standard unicode se situe entre les valeurs 0 et 0x10FFFF, tous les codes points peuvent être stockés tels quels, sur quatre octets et visualisés comme des valeurs comprises entre [00000000] et [0010FFFF]; c'est-à-dire entre 0 et 1665535 ; c'est la forme d'encodage la plus simple en ce sens que la valeur de chaque unité de quatre octets est exactement le code-point du caractère qu'elle représente; ce qui n'est pas toujours le cas avec les autres formes à nombre de bits plus petits. Le déchiffrage d'un fichier en UTF-32 ne requiert donc aucune conversion nécessaire à l'identification du caractère.

La forme UTF-32 est un format d'encodage à taille fixe. Cela constitue son principal avantage en même temps que son principal défaut. En effet, elle conduit à des fichiers et des flux de données beaucoup plus gros que ceux obtenus avec les deux autres formes.

L'espace de code de l'UTF-32 est arbitrairement limité entre les nombres hexadécimaux 0..10FFFF pour les besoins de compatibilité avec l'UTF-16.

Exemples UCS-4/UTF-32

LETTRE MINUSCULE GRECQUE DELTA δ

- N° de caractère : U+03B4
- Unité de stockage en UCS-4 : 0x000003B4

LETTRE GOTIQUE D : **Ɔ**

- N° de caractère : U+10333
- Unité de stockage en UCS-4 : 0x00010333

Forme UCS-2

Dans la forme UCS-2, le numéro de caractère est représenté par seize bits. On dit un seizet. Cette forme de stockage peut, par conséquent, contenir 216 valeurs d'unités de codage, c'est-à-dire 65 536 caractères ou emplacements mémoires contenant les numéros de caractères.

Il faut rappeler que cette forme n'existe qu'en ISO/CEI 10646 et ne permet d'adresser que les caractères du plan multilingue de base plus connu sous le sigle PMB.

Exemple UCS-2

LETTRE MINUSCULE GRECQUE DELTA δ

- N° de caractère : U+03B4
- Unité de stockage en UCS-2 : 0x03B4

LETTRE GOTIQUE D : \mathfrak{D}

- N° de caractère : U+10333
- Unité de stockage en UCS-2 : inaccessible

Forme UTF-16

L'unité de mesure de la forme UTF-16 est le double octet. Cela permet de représenter sans problèmes tous les codes points allant de U+0000 à U+FFFF. Leur représentation est simplement leur valeur numérique stockée sur deux octets que l'on peut visualiser comme allant de (00 00) à (FF FF). Mais les codes points se trouvant au-delà de la valeur 0xFFFF ne peuvent tenir sur deux octets. Ce sont ceux du code complémentaire qui vont de U+100000 à U+10FFFF. La forme UTF-16 convient de les représenter au moyen d'une paire de doubles OCTETS. CES PAIRES SONT DÉSIGNÉES SOUS LE NOM DE PAIRES D'INDIRECTIONS OU PAIRES SUBROGATIVES.

Dans chaque paire on parlera du subrogatif bas et du subrogatif haut pour désigner chacun de ses éléments. Les valeurs numériques retenues pour ces subrogatifs sont des codes points réservés qui ne sont pas affectés à des caractères. Cela garantit l'importante condition de non recouvrement ou de non recouplement de l'encodage. La forme UTF-16 est directement issue des premières versions d'Unicode qui entendaient tout coder au moyen de 65536 valeurs représentables sur deux octets ; c'est-à-dire sur 16 bits. C'est seulement lorsque cet espace s'est avéré insuffisant que le mécanisme des paires subrogatives d'indirections a été conçu. Il a permis d'étendre l'espace d'encodage tout en préservant la compatibilité avec les codes points déjà existants. Quoi qu'il en soit, la forme d'encodage UTF-16 sera obligatoirement un peu plus difficile à implémenter que la forme UTF-32 puisque il faudra gérer ces deux tailles possibles.

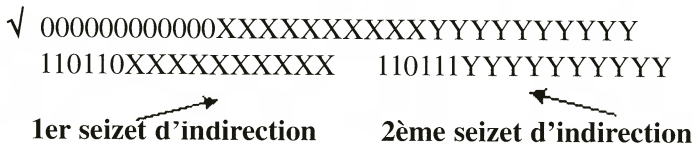
L'avantage de la forme UTF-16 sur la forme UTF-32 est une consommation moindre de l'espace mémoire et par conséquent de l'espace de stockage. Elle constitue un compromis raisonnable entre facilité d'accès et volume des données.

Dans cette forme, le codage est réalisé sur 16 bits comme pour la forme UCS-2. Il concerne directement le plan multilingue de base où les nombres sont codés sur des seizebits. Les caractères sont codés sur un seul seizebit. Les autres plans complémentaires au plan de base sont codés à l'aide de deux seizebits. Ces seizebits sont dits des seizebits d'indirection comme nous l'avons mentionné précédemment. Ces seizebits d'indirection se répartissent en un seizebit d'indirection supérieur compris entre les adresses hexadécimales D800 et DBFF et un seizebit d'indirection inférieur compris entre les nombres DC00 et DFFF.

En d'autres termes, les caractères complémentaires occupent les positions suivantes :

- [0xD800-0xDBFF] = 0x400 = 1024 positions
- [0xDC00-0xDFFF] = 0x400 = 1024 positions

C'est à dire permettant d'adresser un total de 1 048 576 caractères complémentaires obtenu par les produits des positions supérieurs et inférieurs.



Exemples UTF-16

LETTRE MINUSCULE GRECQUE DELTA: δ

- N° de caractère : U+03B4
- Unité de stockage en UTF-16 : 0x03B4

LETTRE GOTIQUE D : ð

- N° de caractère : U+10333
- Unité de stockage en UTF-16 : 0xD800, 0xDF33

La forme UTF-8

La forme d'encodage UTF-8 est sans doute la plus complexe, mais elle répond à des besoins de compatibilité et d'intégration à des protocoles existants. En particulier, pour toutes les ressources qui dépendent essentiellement du code des caractères ASCII, elle permet d'obtenir une équivalence exacte. Un flux de données en ASCII est déjà un ensemble de codes en UTF-8. C'est donc la forme qui permet la migration la plus facile vers Unicode.

Dans cette forme, les codes points sont représentés par un nombre variable d'octets (entre 1 et 4). L'unité de base est donc l'octet et ce sont les bits de ces octets qui permettent de déterminer sans ambiguïté le rôle et la place de l'octet dans une séquence multiple.

On peut déjà indiquer que le premier bit d'un octet sera 0 s'il s'agit d'un code-point qui se trouve dans l'intervalle des code ASCII (c'est à dire entre 0000 et 007F) ou 0 et 127 de l'ASCII, et qu'il sera 1 dans tous les autres cas; autrement dit, aucun octet représentant un code ASCII ne peut figurer dans une séquence représentant un autre caractère. De même les octets pouvant représenter le deuxième, troisième ou quatrième élément d'une séquence multiple sont pris dans des intervalles disjoints, ce qui permet de satisfaire l'exigence de non recoupement pour cette forme d'encodage.

La représentation des codes points en séquence simple entre U+0000 et U+007F sont représentés par un unique octet. Ce sont encore une fois les codes de l'encodage ASCII. Très exactement, ils sont représentés par leurs propres valeurs hexadécimales (de 0x00 à 0x7F), ce qui les rend impossibles à distinguer du code ASCII lui même; tous les codes points entre U+0800 et U+D7FF ou entre U+E000 et U+FFFF sont représentés au moyen de 3 octets : ce sont les codes entres U+0800 et U+FFFF, à l'exception des seizets d'indirection. Tous les codes points supplémentaires situés au-delà de U+FFFF sont représentés par quatre octets;

Les autres codes points sont représentés par deux octets. Cela concerne la plupart des scripts non idéographiques donc autres que les langues asiatiques.

L'encodage est, d'autre part, conçu pour faire en sorte que le premier octet d'une séquence permet par avance de déterminer la taille de la séquence : pour une séquence simple les premiers bits sont toujours 110; pour une séquence triple les quatre premiers bits sont 1110; pour une séquence quadruple les cinq premiers bits sont toujours 11110. Dans tous les cas, les octets qui suivent l'octet initial commencent par les bits 10 ce qui garantit qu'on ne puisse les confondre avec un octet initial. Du point de vue de la programmation et de l'identification des caractères une application n'aura pas à remonter de plus de trois positions dans un texte pour trouver le début d'une séquence.

UTF-8 joue un rôle primordial dans tout ce qui touche à l'Internet car il est l'encodage de de langages tels que HTML et XML.

Exemples UTF-8

LETTRE MINUSCULE GRECQUE DELTA: δ

- N° de caractère : U+03B4
- Unité de stockage en UTF-8 : 0xCE, 0xB4

LETTRE GOTIQUE D : Œ

- N° de caractère : U+10333
- Unité de stockage en UTF-8 : 0xF0, 0x90, 0x8C, 0xB3

Numéro de caractère	1 ^{er} octet	2 ^e octet	3 ^e octet	4 ^e octet
00000 00000000 0xxxxxxx	0xxxxxxx			
00000 00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
00000 zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
uuuuu zzzzyyyy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

Comparaison des trois formes UTF

Les trois formes d'encodage UTF-8, UTF-16 et UTF-32 représentent l'intégration des codes points du répertoire unicode de manière radicalement différente. Par exemple le code point U+10000 sera représenté comme : 0x00010000 en UTF32 mais comme la séquence de quatre octets 0xF0, 0x90, 0x80 et 0x80 en UTF-8. En UTF-8 les codes points sont impossibles à reconnaître directement à partir des octets qui les représentent autres que les simples codes ASCII.

Le choix de l'utilisation d'une forme plutôt que d'une autre dépend essentiellement du contexte : les mêmes considérations peuvent jouer en faveur de solutions différentes selon que l'on est concerné par l'échange des données, par la production de fichiers textes ou par la programmation d'une application et la représentation interne des données dans le cadre de cette application. Un processus peut opter de manipuler toutes les chaînes en interne en utilisant la forme d'encodage UTF-16 tout en exportant ces données en UTF-8.

Étant donné la simplicité de la correspondance, la forme UTF-32 peut apparaître comme le choix le plus évident pour le stockage interne des données : l'intérêt est à la fois la taille fixe des unités de code et la couverture intégrale du répertoire avec une seule de ces unités par code-point. Cela permet aussi de bénéficier de programmes plus simples pour la manipulation des chaînes. En contrepartie, la quantité de mémoire requise est beaucoup plus importante qu'avec les autres formes.

2. La sérialisation

Les unités de code utilisées par les encodages UTF-16 et UTF-32 correspondent du point de vue de la programmation à des types particuliers de données numériques. Le fait de dire que la valeur qui représente un code point est stockée sur deux ou sur quatre octets ne suffit pas à déterminer la manière dont les octets et les bits à l'intérieur de ces octets sont disposés en mémoire : la complication vient du fait que certaines architectures énoncent les octets d'un type particulier dans un ordre tandis que d'autres adoptent l'ordre inverse. On utilise les termes de big-endian (grand boutien) pour désigner le fait que la représentation interne d'une

valeur stockée sur plusieurs octets énonce l'octet le plus fort en premier et l'octet le plus faible en dernier. Inversement, on appelle little-endian (petit boutien) la représentation interne qui va de l'octet le plus faible à l'octet le plus fort.

Dans le cadre d'un échange de données, l'ordre des données adopté est primordial. C'est ce que le standard Unicode appelle la sérialisation des données.

Mécanisme de sérialisation de caractères

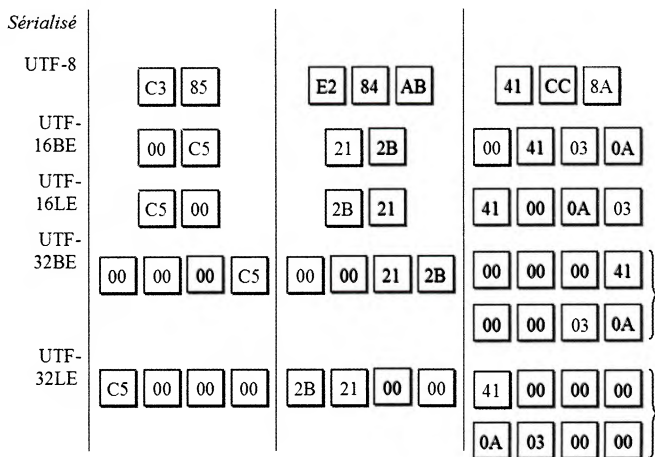
En fait, depuis Unicode 3.0, il existe quatre mécanismes de sérialisation de caractères dans Unicode 3.0 : UTF-8, UTF-16, UTF-16BE et UTF-16LE. Le consortium Unicode a également approuvé UTF-32BE et UTF32LE comme mécanismes de sérialisation.

La sérialisation est le fait de transformer un groupe de bits, reçus en parallèle, en une succession de bits, transmis dès lors en série. En pratique, on sérialise en octets, pour la raison que tous les supports de transfert utilisent cette unité pour le transfert.

Pour les unités de codage telles que l'octet UTF-8, seizets [UTF-16, UCS-2], « trente-deuzets » [UTF-32, UCS-4], le transfert peut être réalisée par deux techniques : d'abord octets de plus petit poids, c'est le transfert « Petit-boutien » ou au contraire par l'octet de grand poids et c'est le transfert Grand-boutien

LE SURCODAGE DE TRANSFERT

Le surcodage de transfert est une transformation réversible de données codées. Ce surcodage concerne les techniques de compression de données telles que les techniques de compression SCSU, BOCCU. Lors de cette compression, il faut éviter l'utilisation de certaines valeurs d'octets non compatible avec les protocoles de transfert de manière à ne pas provoquer une perte de l'information [4], [5]. C'est un point que nous traiterons dans une future publication.



3. Les propriétés de caractères

Il s'agit d'un certain nombre d'informations sur les caractères destinées à mieux servir unicode. On distingue les propriétés normatives qui sont la catégorie auquel appartient le caractère et les propriétés informatives destinées elles à renseigner le développeur sur l'importance de la langue considérée sans que celui-ci soit un locuteur de cette langue.

Ces propriétés servent à l'automatisation du traitement informatique de la langue. Les propriétés des caractères sont utilisées dans les algorithmes de rendu des caractères, de tri à base de classement des chaînes de caractères, de découpe en ligne et en général de tous ce qui peut identifier un ou un ensemble de caractères tels que les caractères liants ou antiliants avec ou sans chasse etc.;

Les Propriétés Unicode concernent toutes les propriétés répertoriées par le système de codage unicode. En général, nous avons la catégorie générale, la classe bidirectionnelle (direction d'écriture de la lettre), la décomposition, les classes combinatoires canoniques, la gestion des liaisons.

La liste officielle des noms de caractères selon leur position dans le codage se trouve dans le fichier à l'adresse <http://www.unicode.org/public/UNIDATA/unicodeData.txt>. Ce fichier texte contient un grand nombre d'informations formatées de manière à être traitées par les programmes informatiques. Une version française de ce fichier se trouve dans le site Hapax canadien. Il y'a au total quinze champs pour définir un caractère.

La catégorie générale

C'est la propriété la plus importante d'un caractère, celle qui va déterminer son comportement dans la plupart des traitements linguistiques et typographiques.

Nous citerons pour exemple les catégories suivantes [2], [4], [5] :

Les lettres (majuscule, minuscule, titrage, modificatrice, autre type), les signes diacritiques ou marques (sans chasse, combinatoire à chasse non nulle, englobante), les nombres (décimal, alphabétique, autre), les ponctuations (connecteur, tiret, ouverture, fermeture, guillemet ouvrant et fermant, autre), les symboles (mathématique, monétaire, modificateur, autre), les séparateurs (espace, de lignes), etc.

Unicode fait l'effort de classer les langues du monde autant que faire se peut. La complexité des langues et leurs ambiguïtés a obligé les rédacteurs de normes à inventer d'autres rubriques pour classer les caractères particuliers très souvent des langues non latines. On trouve également dans cet effort louable de classification les deux caractéristiques suivantes qui peuvent intéresser l'écriture amazighe de l'Afrique du nord (berbère), ce sont les extenseurs dont le rôle est d'étendre ou de

réitérer le caractère qui les précède ou relatifs à la gestion des liaisons par les liants sans chasse ou avec chasse.

La liste officielle des caractères selon leur position dans le codage se trouve dans le fichier à l'adresse : [http://www.unicode.org/UNIDATA/unicode Data.txt](http://www.unicode.org/UNIDATA/unicode%20Data.txt), Ce fichier texte contient un grand nombre d'informations formatées de manière à être traitées par les programmes informatiques. La version française de ce fichier se trouve dans le site francophone canadien Hapax. Il y'a au total quinze champs pour définir un caractère.

La décomposition

Unicode code parfois un même caractère sous deux formes différentes mais dont l'apparence est identique. Il s'agit de la forme précomposée essentiellement pour des raisons historiques avec un seul point de code et une autre forme décomposée (caractère de base en plus du caractère diacritique). Cette forme est plus productive et vivement encouragée par le système Unicode par rapport à la forme précomposée consommatrice d'emplacements mémoire.

Exemple : U+00C5 (Å) // U+0041 (A) U+030A (°)

Cette forme de codage bien qu'économique peut cependant créer des problèmes linguistiques pour certaines langues et poser des problèmes techniques dans le traitement informatique et même linguistique de ces caractères.

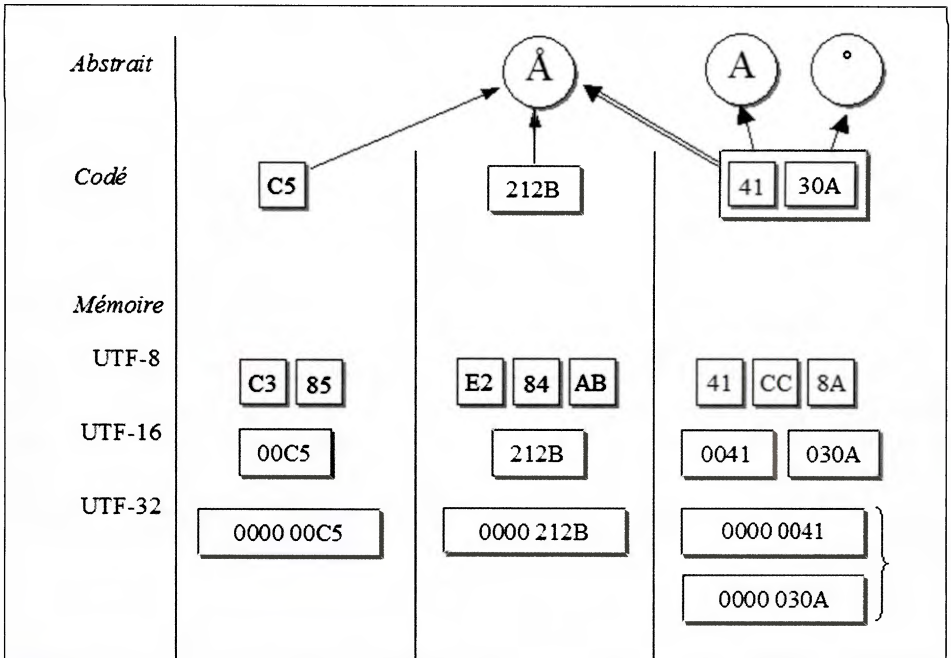
Équivalence canonique

L'équivalence canonique se rencontre lorsque des caractères sont considérés comme identiques et ne diffèrent pas au niveau visuel. Nous donnons comme exemple le caractère É (U+00C9) qui est une variante canonique de E (U+0045) + ` (U+0301)

De manière formelle, On dit que deux suites de caractères sont des équivalents canoniques si leurs décompositions canoniques complètes et récursives respectives sont identiques.

Classes combinatoires

On appelle caractères combinatoires des éléments (caractères combinants) qui se combinent à un caractère de base. Il s'agit essentiellement des accents et autres signes diacritiques qui se combinent à une lettre pour en constituer une autre. Les diacritiques se retrouvent dans de nombreuses langues différentes notamment l'arabe, l'hébreu, etc.

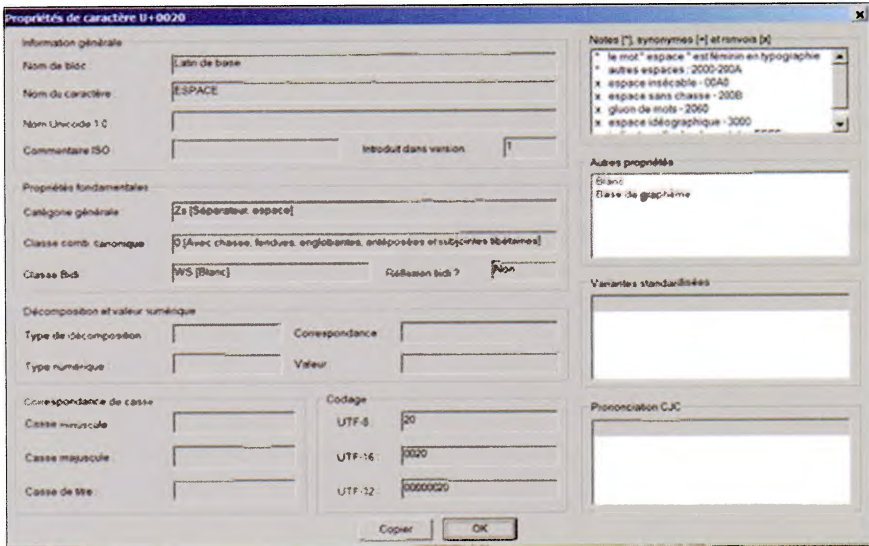
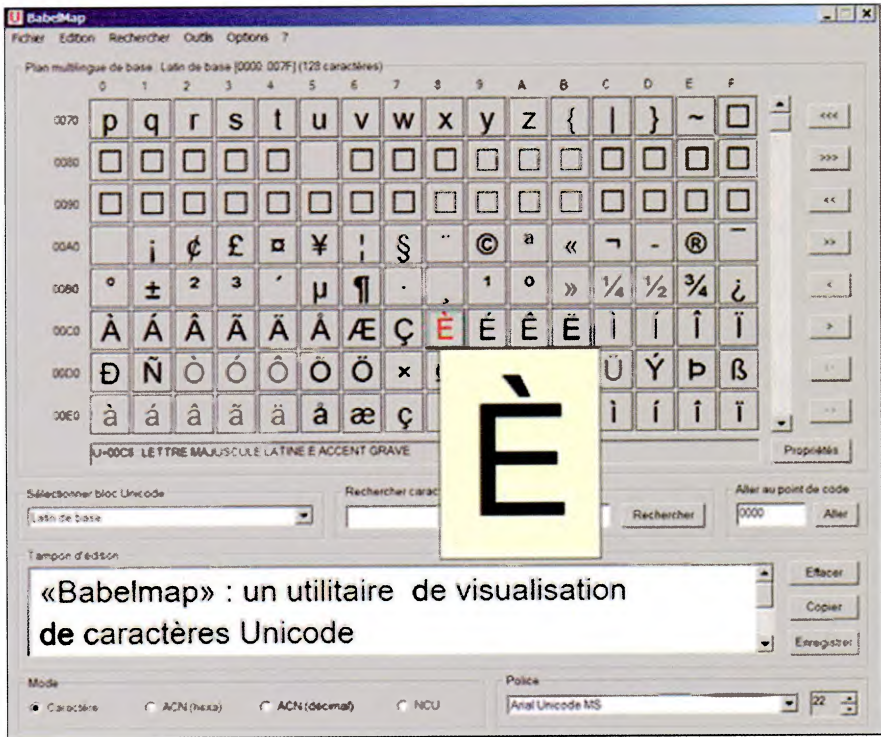


Le caractère de base se retrouve avec différentes positions réservées aux diacritiques qui constituent les classes. Ces classes sont utilisées par l’algorithme de mise en ordre canonique défini par le standard Unicode. Ils sont identifiés par des valeurs numériques décimales : 0: Avec chasse, fendues, englobantes, antéposées ; 1: Couvrantes et intérieures ; 7: Nouktas ; 208: Antéposées et jointes à gauche ; 218: Souscrites à gauche ; 224: Adscrites à gauche (d’un seul car de base) ; 230: Suscrites ; etc.

Par ailleurs, nous pouvons rencontrer les cas suivants : Dans le code UCS, tous les signes diacritiques ont un code qui leur est propre et ils peuvent être combinés avec n’importe quel autre caractère. Ces signes sont dits “ à chasse nulle “. Ainsi, un “ À “ peut aussi bien être obtenu par le recours à son code spécifique (U00C0) que par combinaison d’un A majuscule (U0041) avec le code de l’accent grave à chasse nulle (U0300).

L’utilitaire BabelMap ci-dessous [6] permet de renseigner sur les caractères Unicode et permet plusieurs types de requêtes très utiles pour l’identité complète du caractère concerné.

Le programme BabelMap



Propriétés de composition

Cette partie traite de la nature des caractères : caractères simples ou composites, caractères normaux ou caractères de compatibilité, symboles combinatoires espaçant ou non espaçant. Il fait le point sur le processus de composition et de décomposition sur les questions de normalisation et d'ordonnement.

Décomposition canonique

Certains caractères ont la propriété d'être décomposables en une séquence de caractères élémentaires. Ce sont les caractères composites, composés ou précomposés. C'est le cas des lettres accentuées en français. Le « e » est considéré comme une lettre de base et les accents comme des marques additives qui reçoivent le nom de marques combinatoires.

Dans la plupart des cas les marques ne comportent pas d'espace; mais il existe des marques combinatoires avec espace.

Il résulte de toutes ces considérations que les applications supportant Unicode doivent être capables de réaliser dans certaines circonstances des décompositions de caractères composites ou inversement de recomposer des séquences combinatoires pour trouver un composite correspondant. Ce sont les opérations de composition et de décomposition. La décomposition se fait en utilisant des décompositions fournies par le fichier de propriétés des caractères unicode. Il faut effectuer cette opération de manière répétitive jusqu'à ce que la séquence obtenue ne comporte plus que des éléments indécomposables. La forme à laquelle on aboutit s'appelle décomposition canonique et aussi appelé composite canonique.

Nous pouvons finalement accorder les caractéristiques suivantes aux caractères canoniques décomposés:

- Réversible
- Pas de perte d'information

Cette forme est utilisée dans l'échange normalisé de textes. En effet, cette forme permet d'effectuer une comparaison binaire tout en conservant une équivalence canonique avec le texte non normalisé d'origine

Décomposition de compatibilité

Les choses se compliquent lorsqu'on aborde la question des caractères de compatibilité. Ce sont des caractères qui ne devraient pas figurer dans le bon catalogue Unicode mais qui y ont été acceptés afin de préserver une correspondance avec les encodages préexistants pour, par exemple, garantir des conversions parfaites d'un standard à l'autre.

- Perte d'information visuelle (pas exactement la même apparence)
- La décomposition de compatibilité permet d'effectuer une comparaison binaire tout en conservant cette fois-ci une équivalence de compatibilité avec le texte non normalisé d'origine
- Peut s'avérer utile car elle permet d'éliminer des différences qui ne sont pas toujours pertinentes

Normalisation

Étant donné la multiplicité des représentations possibles pour un même caractère, certaines tâches relatives au traitement de texte se trouvent singulièrement compliquées. Les opérations de recherche d'une chaîne à l'intérieur d'un texte ou l'opérations de comparaison et par voie de conséquence de tri en sont un exemple. On ne peut se contenter d'effectuer des opérations sur le octets en mémoire puisqu'un même caractère pourrait être représenté par un nombre variable d'octets selon qu'il est sous forme composite ou sous forme (partiellement ou totalement) décomposée.

Il est donc indispensable de définir une forme canonique sous laquelle mettre les caractères afin d'éliminer les ambivalences. Une telle forme doit garantir une représentation unique pour chaque caractère.

Le processus de transformation qui met chaque caractère d'une chaîne sous une telle forme canonique est appelé normalisation. La chaîne transformée est dite sous forme normalisée. Le standard unicode définit quatre types de normalisation.

En conclusion, nous pouvons résumer ces notions en disant qu'afin de garantir une représentation unique de ce qui est considéré comme équivalent canonique ou de compatibilité, Unicode définit parfois plusieurs codes qui correspondent :

- à des entités qui peuvent être considérées comme identiques (variantes canoniques)
- ou qui ne sont que des variantes visuelles d'un même caractère (variantes de compatibilité)

	Sans composition canonique	Suivie d'une composition canonique
Décomposition Canonique	D	C
Décomposition de compatibilité (K)	KD	KC

4 formes normalisées

Formes normalisées d'« affligé »

Nom	Chaîne normalisée
<i>D</i>	$a + \text{ff} + i + g + e + \acute{e}$
<i>C</i>	$a + \text{ff} + i + g + \acute{e}$
<i>KD</i>	$a + f + f + l + i + g + e + \acute{e}$
<i>KC</i>	$a + f + f + l + i + g + \acute{e}$

Quelques définitions :

L'ISO est l'organisation internationale de normalisation, c'est-à-dire la fédération mondiale des organismes de normalisation nationaux de quelque 140 pays (le SNIMA au Maroc, l'AFNOR en France, l'institut DIN en Allemagne, l'ANSI aux Etats-Unis, etc.).

Elle a été créée en 1947 et a vocation à favoriser la définition de normes internationales en vue de faciliter les échanges économiques et la coopération intellectuelle, scientifique et technique entre les nations. Ses travaux aboutissent à des accords internationaux qui sont publiés sous forme de normes internationales.

Il faut noter qu'ISO était le sigle du nom initial de l'organisme (International Standards Organization), que ce nom a été modifié dans les deux langues officielles - International Organization for Standardisation désormais en anglais - et que le nom " ISO " est désormais présenté (joliment) comme le préfixe grec iso-, qui signifie " égal " : l'ISO s'attache à définir des systèmes de mesure égale pour tous.

Norme

Définition d'une technologie ou d'un procédé formalisé par un organisme habilité et indépendant.

Une norme est la définition complète et précise d'une technologie, d'un procédé, d'une procédure, etc., créée et promulguée par un organisme habilité à cet effet, par une collectivité nationale ou internationale et indépendant de tout fournisseur de ces technologie, procédé, procédure, etc.

Parmi les principaux organismes de normalisation internationaux, on compte l'ISO (International Standards Organization) et l'ITU-UIT (International Telecommunication Union - Union internationale des télécommunications); les organismes nationaux sont, par exemple, le SNIMA (Service de Normalisation Industrielle Marocaine) pour le Maroc, l'AFNOR (Association française de normalisation), etc.

Standards

En français, on fait une distinction entre norme et standard, le second terme étant réservé à des définitions normalisées par un consortium de fournisseurs et non pas par un organisme indépendant tel que l'ISO par exemple.

Bibliographie :

- [1] - « Éléments de la théorie de l'information et du codage », H. Mertens, Service de télécommunications, Faculté des sciences appliquées, université de liège, octobre 1987
- [2] - « De la logique câblée aux microprocesseurs », Jan Michel Bernard et all, édition Eyrolles, 1979, collection technique et scientifique des télécommunications.
- [3] - Passeport pour unicode, Bernard Desgraupes, Editions Vuibert Informatique
- [4]- Fontes et codage, par Yannis Yaralambous, O'Reilly, 2004
- [5]- Introduction à Unicode, Patrick Andries
<http://cooptel.qc.ca/~pandries/pdf/intro-Unicode.pdf>
- Unicode en français :
- <http://cooptel.qc.ca/~pandries/pdf/Chapitre-4.pdf>
 - <http://cooptel.qc.ca/~pandries/pdf/Chapitre-6.pdf>
 - <http://pages.videotron.com/hapax/UCD-4.0.0.fr.html>
- [6]- Babelmap : http://www.cooptel.qc.ca/~pandries/BabelMap_fr.html

UNICODE ET LES POLICES : DEUX MONDES

Patrick Andries

Conseils Hapax <http://hapax.qc.ca>

Membre du consortium Unicode

Comment passe-t-on du monde des caractères qui est celui d'Unicode à celui des glyphes qui représentent ceux-ci visuellement ? Comment sont stockés ces glyphes ? Qu'est-ce qu'une police de « caractères » comme on dit encore trop souvent ? Voilà les questions auxquelles cet article tente de répondre tout en introduisant une des dernières techniques en matière de polices : OpenType, le successeur des polices TrueType. L'article décrit également brièvement le moteur de rendu qui interprète les polices et équipe toutes les versions récentes du système d'exploitation Windows de Microsoft : Uniscribe.

1. Unicode

1.1. Des caractères et non des glyphes

Comme nous l'avons vu dans l'Introduction à Unicode et à l'ISO/CEI 10646 de ces actes, cette norme ne se préoccupe que des caractères et non de leurs aspects, leurs glyphes. Unicode préconise également de stocker les caractères en « ordre logique », c'est-à-dire dans l'ordre phonétique et le plus souvent dans l'ordre de saisie. Enfin, Unicode présume l'existence d'un moteur de rendu qui pourra, le cas échéant, réordonner les caractères en ordre visuel et pourra substituer aux caractères des glyphes adéquats tout en les plaçant de manière précise.

1.2. Pourquoi ce modèle ?

On pourrait se demander pourquoi Unicode a choisi ce modèle, appelé modèle caractère-glyphe, alors qu'a priori on pourrait penser qu'il aurait pu être tellement plus simple de ne coder et de ne stocker que des glyphes.

Au-delà du simple fait que la codification de variantes de caractères est une tâche herculéenne, car on crée chaque jour de nouvelles polices, il est nettement plus judicieux de se concentrer sur les caractères sans se soucier de leur apparence pour toute une série de traitements qui vont du correcteur grammatical et de la traduction automatique à l'indexation en passant par la recherche et remplacement de texte, le copier-coller, l'exportation de texte, le changement de police ou le changement de style de paragraphe ou de caractère. En codant des caractères et non

des glyphes, on s'assure que ces opérations fonctionneront quelle que soit l'apparence finale de la chaîne de caractères à traiter.

2. Fonctionnement d'un moteur de rendu

Nous décrivons ci-dessous le fonctionnement quelque peu simplifié et idéalisé d'un moteur de rendu. Il ne s'agit pas ici de décrire un moteur de rendu particulier, mais de mieux comprendre les différentes étapes reliées à la composition et la part que jouent Unicode et les polices à chacune de ces étapes.

2.1. Découpe en passages de même style

Une des premières tâches d'un moteur de rendu est de découper le texte stocké en mémoire en passages (ou parle aussi parfois de segments) de même style, même police, même système d'écriture et enfin de même directionnalité⁽¹⁸⁾. La figure 1 ci-dessous illustre une telle découpe. Remarquez que les caractères arabes sont représentés en ordre logique et non contextualisés.

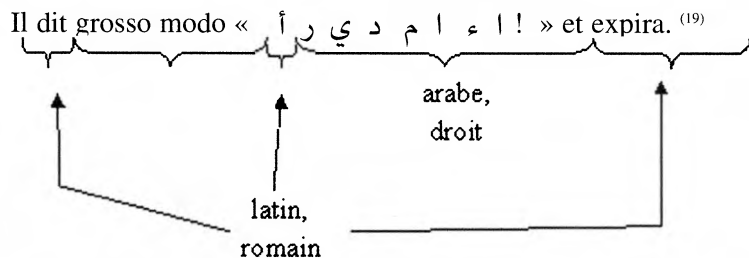


Figure 1. - Découpe de phrase bidirectionnelle

2.2. Coupure de lignes

Une coupure de lignes de base est relativement simple à mettre en œuvre pour l'écriture latine : il est permis de couper une ligne à l'espace, au trait d'union (réel ou virtuel) ou à la tabulation. Pour d'autres langues, telles que le thaï ou le lao, les mots se suivent sans qu'une espace ne les sépare. La coupure des lignes dans ces langues doit s'effectuer à la frontière de syllabes. Dans ces langues, une coupure de ligne de base acceptable nécessite un algorithme de syllabisation similaire à ce qu'on utilise en français pour effectuer la division de mots en fin de ligne. Notons qu'Unicode fournit des recommandations et propriétés de caractère qui permettent de mettre en œuvre un algorithme de coupure de lignes⁽²⁰⁾ de base.

(18) - C'est-à-dire droite-à-gauche (l'arabe, l'hébreu, le syriaque, le n'ko, etc.) ou gauche-à-droite (le latin, le grec, le cyrillique, etc.).

(19) - Ce que l'on peut représenter également sous la forme suivante, si on remplace les lettres arabes par des capitales : Il dit grosso modo « DE L'EAU ! » et expira. On voit donc que le mot arabe est stocké en mémoire en ordre logique, c'est-à-dire de gauche à droite alors que l'écriture arabe s'affiche de droite à gauche.

(20) - Voir <<http://hapax.qc.ca/pdf/Chapitre-14.pdf>>, section 14.2, Commandes de mise en page.

Il est important de noter que la coupure de lignes est un processus itératif qui implique souvent, plusieurs passages aux glyphes (voir 2.3. Réordonnement et 2.4. Sélection des glyphes, les deux étapes suivantes). En effet, pour déterminer une position de coupure permise, on utilise les propriétés des caractères Unicode ; mais pour savoir si la chaîne ainsi découpée est de bonne taille et peut rentrer dans l'espace réservée pour la ligne, il faut en connaître la taille en termes de glyphes, éventuellement crénés, ligaturés ou les deux. Si la chaîne ne rentre pas, on choisira un autre point de coupure plus proche, une nouvelle fois en se référant aux propriétés des caractères, pour s'enquérir ensuite de la taille des glyphes correspondants à cette chaîne de caractères raccourcie.

Le processus décrit ci-dessus est volontairement épuré. Ainsi, TeX n'effectue-t-il pas ce processus sur une base de lignes mais il détermine à chaque essai les meilleures positions de coupure de lignes au niveau du paragraphe tout entier.

2.3. Réordonnement

Après avoir découpé le texte en mémoire, le moteur de rendu réordonne habituellement les passages pour les faire passer de l'ordre logique à l'ordre visuel et ainsi faciliter le passage à l'affichage des glyphes. Cette remise en ordre implique alors l'application de l'algorithme bidirectionnel ⁽²¹⁾.

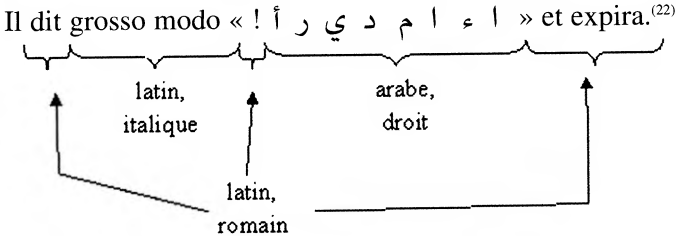


Figure 2. - Passage en ordre visuel

Il est important de noter que ce réordonnement doit avoir lieu après la coupure ligne. En effet, si on coupe la ligne après le premier mot arabe il faut que celui-ci reste sur la première ligne après le passage en ordre visuel.

2.4. Sélection des glyphes

L'étape suivante consiste à passer de l'espace de caractère qu'est Unicode à la représentation de ces caractères en termes de glyphes, les composants des polices. Le passage des caractères aux glyphes est la première étape obligatoire et systématique. Par la Suite, selon le type de police utilisée, la forme précise du

(21) - Voir <<http://hapax.qc.ca/pdf/Chapitre-3.pdf>> section 3.12, Comportement bidirectionnel, et section 9.2, Arabe dans <<http://hapax.qc.ca/pdf/Chapitre-9.pdf>>. Unicode ne prescrit pas la mise en œuvre de cet algorithme bidirectionnel, mais bien le résultat que la mise en œuvre doit produire.

(22) - En remplaçant les lettres arabes par des capitales : Il dit grosso modo « ! UAE'L ED » et expira.

glyphe à afficher s'effectuera par un nombre de substitutions optionnelles. Nous reviendrons en détail sur ce processus par la suite.

2.5. Justification

Les suites de glyphes sont ajustées pour respecter les règles de justification en vigueur. Pour ce faire, l'algorithme de justification changera l'espace intermot ou, si la police contient ces informations, utilisera une variante d'œil plus étroit ou plus large pour certains glyphes.

Exemple de texte dit en drapeau, aligné à gauche ou encore fer à gauche. Exemple de texte dit en drapeau, aligné à gauche ou encore fer à gauche. Exemple de texte dit en drapeau, aligné à gauche ou encore fer à gauche.

Exemple de texte dit justifié, en bloc ou en encore en paragraphe réparti, avec au besoin la coupure de mots. Exemple de texte dit justifié, en bloc ou en encore en paragraphe réparti, avec au besoin la coupure de mots.

2.6. Tramage et optimisation du rendu

Les glyphes dans la majorité des polices modernes sont définis en terme d'équations qui définissent les contours de ces glyphes⁽²³⁾. Or, en fin de compte, ces contours devront apparaître sous la forme de pixels noircis ou non. Si on n'avait à rendre une police qu'à une seule force⁽²⁴⁾ et à une seule résolution d'écran ou d'imprimante, il n'y aurait nul besoin de définir les glyphes sous la forme de contours, une image en mode point suffirait. L'utilisation de contours permet à la police de s'adapter à différentes résolutions et forces de corps.

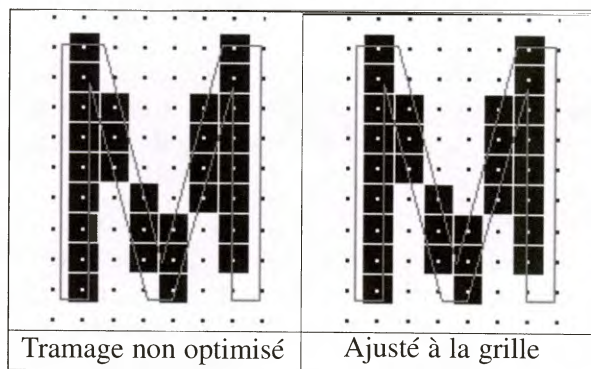


Figure 4. - Optimisation du rendu

(23) - D'autres sont définis directement en termes de pixels, on appelle ces polices des polices « bitmap » ou en mode « point ».

(24) - Taille du caractère en points : « un corps de force 6 », « du 6 » de façon elliptique.

Un trameur se charge du passage des contours vectoriels à une trame de carrés équidistants disposés en une grille. La tâche du trameur est donc de noircir les bons carrés pour épouser au mieux le contour de la lettre à rendre. Cette tâche n'est toutefois pas aisée, car il suffit parfois d'une petite différence de contour ou une troncature correcte, mais malencontreuse, pour qu'un mauvais carré soit noirci. Ainsi, même si le contour est parfait l'apparence tramée peut être illisible. Ce problème est plus fréquent, ou du moins plus visible, aux petites résolutions, car les carrés sont plus gros. Pour y pallier, on a prévu différentes méthodes qui permettent d'optimiser le rendu et d'ajuster le résultat du tramage. Dans le cas des polices TrueType et de leurs successeurs, les polices OpenType, cette optimisation se fait à l'aide d'instructions supplémentaires. Celles-ci permettent de modifier la forme des contours pour l'adapter à la grille de tramage donnée, c'est-à-dire pour une résolution et une force données. Ce nuancement des contours permet de guider l'algorithme de tramage et l'aider à noircir les bons carrés. L'optimisation précise d'une police TrueType pour en assurer un rendu de qualité à l'écran est un véritable travail de bénédictin.

3. Les polices de glyphes⁽²⁵⁾

3.1. Historique

En 1985, Adobe révolutionne la production et la publication de documents électroniques sur microordinateurs en introduisant le langage de description de page PostScript. Adobe frappe un autre grand coup quand il signe un accord avec la société Linotype, chef de file américain dans le domaine des polices, pour numériser et convertir son énorme bibliothèque de polices en format PostScript. Grâce à l'utilisation commune de Postscript par les tireuses d'épreuves (l'imprimante laser) et au système imageur (l'imprimante de qualité), on pouvait garantir la cohérence du résultat. Bien que la définition de l'imprimante laser n'était que de 300 points par pouce (ppp) et celle du système imageur était de 1270 ppp, la géométrie de la page était identique. On pouvait donc être assuré que ce qu'on imprimait sur son imprimante laser serait reproduit par l'imprimante à haute définition. C'était une énorme innovation.

Ce succès ne fut pas sans inquiéter les deux meneurs de file de la micro-informatique de l'époque - Apple et Microsoft - qui se mirent à la recherche d'un autre système de rendu de polices pour écran. De ces efforts naquit TrueType qu'Apple sortit en 1990 et Microsoft un an plus tard.

(25) - Nous avons délibérément évité le terme traditionnel de « police de caractères » car ces fontes ne contiennent justement pas des caractères, mais des glyphes - les œils en typographie traditionnelle française - associés à des caractères ou des suites de caractères.

Pendant toutes les années 90, les systèmes d'exploitation de Microsoft ne proposaient que des polices TrueType. Cependant Adobe et Microsoft se réunirent pour mettre au point un format de police unifié. Le résultat, désormais connu sous le nom d'OpenType, est un format multi-plateforme qui peut contenir à la fois des contours TrueType et PostScript ainsi que de nouvelles fonctionnalités de typographie de pointe. Avec Windows 2000 et Windows XP, Microsoft offre un trameur PostScript intégré, ceci signifie que ces deux versions de Windows prennent en charge de manière native OpenType (que les contours des glyphes soient décrits sous forme PostScript ou TrueType). OpenType semble promis à un bel avenir et sa prise en charge par les applications (car comme on le verra une partie de la prise en charge est laissée aux applications) se répand de plus en plus, c'est pourquoi dans le reste de cet article nous nous concentrerons sur ce format de police.

3.2. Une police, un ensemble de tables

Mais, sous le capot, quel est le format que prend une police ? Il s'agit d'un ensemble de tables où chaque table a une fonction différente. Certaines tables sont standard et parfois communes à TrueType (l'ancien format compris par toutes les applications Windows) d'autres introduits par OpenType. Notons enfin que certaines tables peuvent être définies par un fondeur sans accord préalable ou permission avec des tiers, on les dit privées.

Nom de la table ⁽²⁶⁾	Prise en charge
cmap	True Type et OpenType
glyf	True Type et OpenType
hmtx	True Type et OpenType
CCF	Post Script et OpenType
GSUB	Open Type
GPOS	Open Type
...	...

Tableau 1. - *Tables OpenType*

Rappelons aussi que les polices sont passives : elles ne font par elles-mêmes rien. Elles définissent des informations. Un programme les interprète, il s'agit du moteur de rendu.

(26) - Notons que les noms des tables en TrueType sont en minuscules alors que les noms de tables OpenType sont en majuscules.

3.2.1 La table « cmap »

Cette table obligatoire permet de passer de l'espace des caractères (par exemple Unicode) à l'espace des glyphes. Notez qu'à un même glyphe peuvent correspondre plusieurs caractères. C'est typiquement le cas de ces trois caractères dans les polices qui les connaissent :

U+002D	TRAIT D'UNION-SIGNE MOINS
U+00AD	TRAIT D'UNION VIRTUEL ⁽²⁷⁾
U+2011	TRAIT D'UNION INSÉCABLE

Ces caractères ont, en règle générale, le même œil⁽²⁸⁾ dans l'écriture latine et se référeront donc habituellement au même glyphe dans une police; ils ne se distinguent que par leur propriété (de sécabilité) et c'est pourquoi ces trois caractères sont codés séparément dans Unicode. Il existe, comme on le verra plus en détails plus tard, également des cas où un glyphe ne correspond à aucun caractère Unicode.

3.2.2. La table « glyf »

La seconde table obligatoire, en l'absence d'une table CCF (cf. 3.2.3), héritée de TrueType est la table glyf. Il s'agit souvent de la table la plus volumineuse, elle contient les contours de glyphes. Chaque entrée de la table correspond à la définition d'un glyphe. Le tableau 2 ci-dessous illustre schématiquement le contenu de la table glyf.

N° de glyphe <i>x</i>	Rectangle englobant (coins opposés)
	Contour 1
	Contour 2
	Instructions de nuancement
N° de glyphe <i>x+1</i>	Rectangle englobant (coins opposés)
	Contour 1
	Contour 2
	Contour 3
	Contour 4
	Instructions de nuancement
N° de glyphe <i>x+2</i>	Rectangle englobant (coins opposés)
	Contour 1
	Contour 2
	Instructions de nuancement
	...

Tableau 2. - La table « glyf »

(27) - Le trait d'union virtuel n'a nécessairement d'œil (de dessin), en effet depuis Unicode 4.0 ce caractère est de catégorie générale Cf (« Autre caractère de formatage »).

(28) - L'œil est le dessin du glyphe, on dit au pluriel des œils.

Chaque entrée commence par la définition du rectangle englobant le glyphe, il s'agit d'une paire de coordonnées qui correspondent aux coins opposés du plus petit rectangle qui englobe le glyphe. Vient ensuite une série de contours fermés sous la forme d'une série de points qui définissent la forme du glyphe. Parmi les autres données qu'on retrouve pour chaque glyphe, il faut signaler les instructions de nuancement⁽²⁹⁾ - il s'agit d'un véritable code assembleur - qui permettent d'optimiser le rendu à faible définition en modifiant légèrement les contours à une force et une définition données.

Une entrée simplifiée pour une barre oblique « / » peut donc se représenter en XML, comme ci-dessous. Ces données structurées se trouvent codées dans la police sous forme binaire⁽³⁰⁾.

```
<Glyphe xMin="2" yMin="-30" xMax="574" yMax="1423">
  <contour>
    <pt x="85" y="-30" sur="1"/>
    <pt x="2" y="-30" sur="1"/>
    <pt x="495" y="1423" sur="1"/>
    <pt x="574" y="1423" sur="1"/>
  </contour>
  <instructions>
    <!-- code compilé se trouve ici -->
  </instructions>
</Glyphe>
```

La valeur de l'attribut sur vaut « 1 » quand le point définit un point de contour, « 0 » quand il définit un de commande (ou de contrôle)⁽³¹⁾. Ce contour correspond au dessin illustré à la figure 5 ci-dessous.

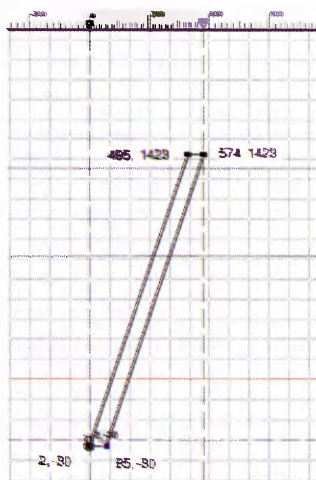


Figure 5. - Contour d'un glyphe

(29) - On dit parfois «hinting» même en français bien que, stricto sensu, ce terme ne s'applique qu'aux polices postscript.

(30) - Cf. page 697 de Yannis Haralambous, ouvrage cité dans la bibliographie.

(31) - Voir l'article de Youssef Aït Ouguengay dans ces mêmes actes.

3.2.3. La table « CCF »

Si la table glyf contient des contours TrueType, La table CCF contient des contours PostScript. En effet, les contours peuvent être exprimés indifféremment en format TrueType ou en PostScript. Chaque mention de la table glyf dans ce texte désigne la table glyf ou la table CFF. Le seul endroit où la technologie importe est dans la section 2.6 : le nuancement est explicitement la responsabilité du créateur de la police pour TrueType, alors qu'en PostScript, le créateur fournit des contraintes que le trameur doit respecter (le trameur doit faire plus de travail).

3.2.3. La table « hmtx »

Cette table obligatoire est l'une des plus simples. Il s'agit de la table des « métriques horizontales » de la police. Elle contient pour chaque glyphe, sa chasse et son approche gauche. La distance horizontale entre la position actuelle (avant l'insertion du glyphe) et l'extrême gauche du glyphe est son approche gauche. La distance horizontale parcourue après l'insertion du glyphe est sa chasse. Ainsi, un glyphe en surplomb aura-t-il un chasse nulle et probablement une approche gauche négative s'il doit suivre le glyphe surimprimé de la sorte alors qu'une espace aura une grande chasse bien que ce glyphe n'ait pas d'œil.

3.2.3. La table « GSUB »

Une des nouvelles tables introduites par OpenType est la table GSUB. Cette table définit la substitution d'un ou plusieurs glyphes par un ou plusieurs autres glyphes. Plusieurs types de substitutions sont possibles :

Simple	a → A
Décomposition	ffi → f f i
Sélection de variante	& → &, &, &, &, &, &, &
Ligature	f f i → ffi
	ل ا → لا
Contextuelle	d a n s → <i>dans</i>

Ainsi, on trouve des substitutions qui permettent de remplacer un glyphe par un autre, de décomposer une ligature en ses glyphes constituants, de sélectionner une variante, de former des ligatures à partir de deux glyphes (comme la ligature arabe *lam alif*) ou encore de modifier un glyphe selon son contexte. On peut se demander ce qui distingue la sélection de variante de la substitution simple. Après tout, on substitue un glyphe par un autre dans les deux cas. La différence réside dans le fait que pour la sélection de variante le logiciel permet à l'utilisateur de choisir la

variante qu'il préfère alors que pour la substitution simple aucune interaction avec l'utilisateur n'est nécessaire. Dans le cas de dans , les lettres sont par défaut pourvues d'un queue et dépourvues d'une attaque (*dans*). Grâce à une substitution contextuelle, la terminaison du s final est supprimée et le d initial prend un petit trait appelé attaque.

On voit qu'à l'aide de ces substitutions il est tout à fait possible de définir des glyphes qui ne correspondent directement à aucun caractère Unicode particulier, mais bien à une suite de caractères Unicode.

3.2.4. La table « GPOS »

La table GPOS définit le déplacement de glyphes par rapport à leur position habituelle, à d'autres glyphes (par exemple une paire de glyphes pour mettre en œuvre le crénage) ou pour placer des diacritiques sur des lettres de base ou encore sur d'autres diacritiques.

Le déplacement d'un seul glyphe permet de mettre ce glyphe en indice, en exposant ou de mieux centrer des parenthèses quand on compose en petites capitales (cf. ci-dessous). Remarquez comme les parenthèses du dernier (BACH) en petites capitales, mieux centrées que son voisin, laissent une bien meilleure impression.

(BACH) (BACH) (BACH)

Grâce au déplacement de paires de glyphes, on peut non seulement mettre en œuvre un crénage traditionnel horizontal (cf. ci-dessous), mais également déplacer chacun des deux glyphes séparément dans toutes les directions.

AVION → AVION

La table GPOS permet de préciser pour chaque diacritique un point de fixation (d'une certaine classe) qui viendra s'attacher à un point de fixation de même nom défini sur le glyphe de base. Chaque glyphe de base possède autant de points de fixation qu'il n'y a de classe de diacritiques. Ainsi si l'on prend un O, comme dans l'illustration ci-dessous, on définit un ensemble de points de fixation dont un que l'on nommera haut_centre. Tous les diacritiques qui peuvent s'attacher à ce point définissent un point de fixation de classe haut_centre.

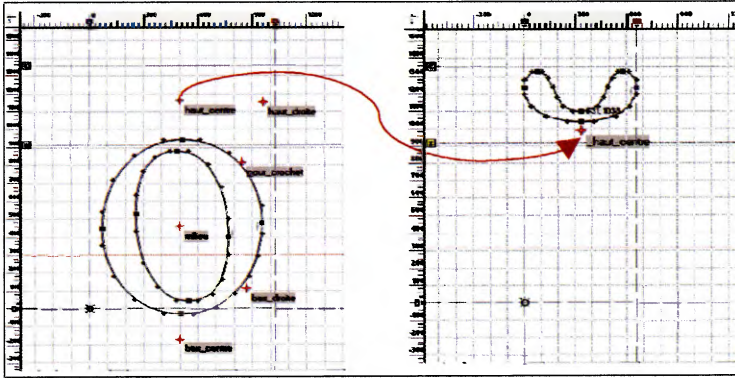
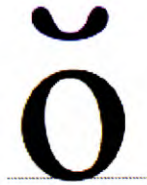


Figure 6. - *Points de fixation*

Dans l'exemple ci-dessus, nous avons illustré le cas d'un signe diacritique « brève » que l'on veut adjoindre à un O. Un moteur de rendu OpenType placera le brève correctement au-dessus du O en utilisant les points de fixation.



Plusieurs langues (le vietnamien) et notations (comme l'alphabet phonétique international) utilisent de multiples signes diacritiques. La table GPOS permet de définir ce genre de positionnement relatif.

a A E

Figure 7. - *Exemple de lettres vietnamiennes à accents multiples*

3.2.5. GSUB ou GPOS ?

En théorie, il est tout à fait possible de se passer de GPOS pour placer ou déplacer un diacritique. En effet, on peut obtenir le même résultat en utilisant des substitutions de plusieurs glyphes (la lettre de base et un ou plusieurs diacritiques) vers un nouveau glyphe précomposé.

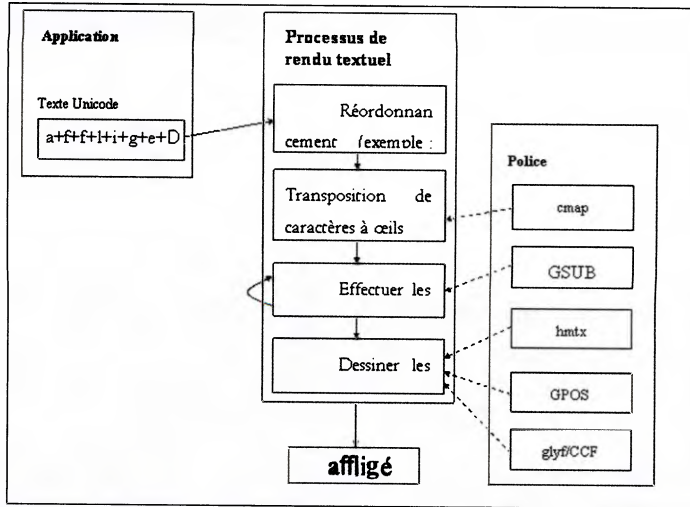


Figure 8. - Étapes internes du processus de rendu

Chaque solution comporte un avantage : les substitutions sont plus rapides car aucun calcul de positionnement n'est nécessaire. Toutefois, pour des écritures fortement diacritiquées comme l'arabe ou certaines variantes latines, l'utilisation de GPOS est nettement plus économique en terme de développement de la police et de taille de la police car on définit dans ce cas nettement moins de glyphs. En principe, toutes les lettres de base peuvent être ornées d'un ou plusieurs diacritiques génériques [U+0300..U+036F], on conçoit facilement que la création de glyphs précomposés pour toutes ces combinaisons est une tâche impossible à réaliser avec des substitutions.

4. Processus de rendu

Maintenant que nous avons introduit les principales tables OpenType, intégrons-les au processus de rendu. La figure 8 ci-après illustre un modèle de rendu textuel typique. Dans cet exemple, le texte-source Unicode est en forme normalisée KD⁽³²⁾, le résultat affiché pour sa part utilise la ligature « _ » et un é précomposé. Le processus de rendu commence d'abord par exécuter l'algorithme bidirectionnel d'Unicode⁽³³⁾ dans l'espace des caractères. Il effectue ensuite le passage des caractères aux œils grâce à la cmap. Pour chaque passage directionnel⁽³⁴⁾, il remplace les glyphs selon les règles définies par la table GSUB et, le cas échéant, des

(32) - Voir l'article du prof. Zenkour dans ces mêmes actes.

(33) - Unicode ne prescrit pas la mise en œuvre de cet algorithme, mais le résultat de sa mise en œuvre.

(34) - C'est-à-dire dans une suite de caractères de même directionnalité (droite-à-gauche, gauche-à-droite).

paramètres passés par l'application⁽³⁵⁾. Ces remplacements se font de manière itérative : une ligature issue d'une substitution peut participer à une substitution ultérieure. Ensuite, une fois l'ordre et l'identité des œils déterminés, on calcule leur position relative précise à l'aide de la table hmtx et on raffine celle-ci grâce à la table GPOS. Enfin, le système d'exploitation utilise les tables glyf ou CCF et certaines autres tables pour dessiner les œils.

La prise en charge des fonctionnalités originales d'OpenType, comme les substitutions, nécessite une modification des applications. C'est ce qui explique qu'il ne suffit pas de créer une police OpenType pour que MS Word affiche les ligatures qu'on aura pu ajouter à ces polices. En fait, aucun progiciel de Microsoft ne prend actuellement en charge les ligatures latines ! Microsoft n'utilise la formation de ligatures GSUB que dans les langues non latines où elles s'avèrent essentielles. Toutefois d'autres produits, comme InDesign d'Adobe, permettent la formation de ligatures latines OpenType pour les écritures latines.

5. Un moteur de rendu Uniscribe

Uniscribe est une DLL⁽³⁶⁾ qui fait partie du système d'exploitation Windows. Il s'agit d'un moteur de rendu conçu pour prendre en charge les fonctionnalités des polices OpenType sur Windows. Uniscribe est une partie intégrale de Windows 2000 et Windows XP. La sortie d'Uniscribe s'est accompagnée de la sortie de nouvelles versions d'Internet Explorer (la version 5) et d'applications faisant partie d'Office 2000 conçues pour tirer parti des fonctionnalités offertes par Uniscribe.

Uniscribe est en réalité un ensemble de DLL, chacune de celles-ci, appelée moteur de façonnage par Microsoft, est chargée d'une écriture⁽³⁷⁾. Chaque moteur de façonnage applique un sous-ensemble de fonctionnalités OpenType dans un ordre lié à l'écriture qu'il prend en charge. La première version d'Uniscribe ne prenait en charge que trois écritures : la dévanâgarî, le tamoul et l'arabe. Depuis lors, plusieurs versions améliorées d'Uniscribe sont sorties avec Windows XP et Office XP, elles ont notamment introduit la prise en charge du thaï, du syriaque et de la composition des accents latins. Toutefois, de nombreuses écritures ne sont toujours pas prises en charge correctement, particulièrement les ligatures latines. Pour ce qui a trait aux ligatures touarègues, celles-ci ne seront certainement pas prises en charge avant la sortie de la prochaine version de Windows, au nom de code « Longhorn ».

(35) - Dans le cas d'OpenType, les substitutions sont réalisées par le client et non le système d'exploitation. Le client a donc toute latitude pour sélectionner les substitutions désirées. En revanche, dans le cas du système ATSUI sur Macintosh, tout le processus est effectué de manière transparente par le système d'exploitation.

(36) - DLL est l'abréviation anglaise de « bibliothèque à lien dynamique ». Une DLL est un ensemble de petits programmes qui peuvent être appelés par un programme plus important pour effectuer une tâche ou une série de tâches particulières. Le principal avantage des DLL tient au fait qu'elles ne sont chargées en mémoire que lorsqu'on les appelle, libérant ainsi la mémoire pour d'autres processus.

(37) - En général, car il existe aussi des moteurs de façonnage responsables de l'affichage des chiffres.

La figure 9 illustre les principales opérations quand une application prévue pour travailler avec Uniscribe désire afficher une chaîne de caractères arabes (avec potentiellement du texte latin). L'application divise d'abord le texte en passages de même style (par exemple de même police, couleur et force). L'application appelle Uniscribe pour diviser ces passages en « items », c'est-à-dire des fragments de même écriture et directionnalité. Cette découpe permet de découvrir le moteur de façonnage qui sera chargé de rendre chaque segment.

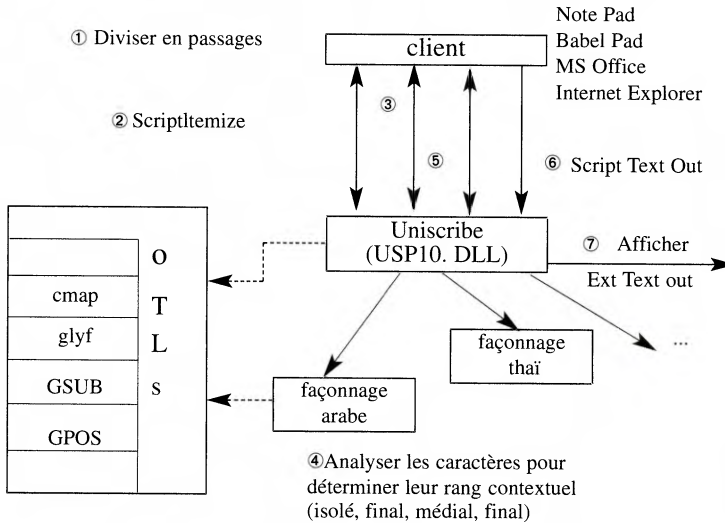


Figure 9. - Uniscribe

En ③, l'application appelle la fonction ScriptShape d'Uniscribe, pour chaque passage, en lui fournissant les caractères à afficher et le résultat de ScriptItemize. La fonction ScriptShape rend un vecteur de glyphes qui correspond aux caractères fournis en entrée. Pour déterminer ce vecteur de glyphes, le moteur de façonnage, arabe en l'occurrence, doit analyser les caractères fournis en entrée pour déterminer leur rang contextuel④, c'est-à-dire si un caractère donné est un caractère initial, médial, terminal ou isolé. Ensuite, avec cette connaissance, le moteur arabe effectue les substitutions (GSUB) nécessaires. Le moteur de façonnage peut également effectuer d'autres substitutions qui n'impliquent pas le rang contextuel des caractères. Nous reviendrons plus en détails sur ce sujet. Notons que, pour accéder aux polices, Uniscribe et les applications peuvent profiter de la bibliothèque d'OTLS qui facilite l'accès aux différentes tables OpenType.

En ⑤, l'application appelle la méthode ScriptPlace avec le résultat de ④ pour calculer la chasse⁽³⁸⁾ et les coordonnées des glyphes. Enfin, l'application appelle ScriptTextOut ⑥ pour afficher les glyphes d'un passage en fournissant les glyphes,

(38) - La largeur des glyphes.

leurs approches et coordonnées. ScriptTextOut appelle la fonction système d'exploitation ExtTextOut de manière appropriée et le passage arabe s'affiche. Notons enfin que l'application peut appeler d'autres fonctions Uniscribe pendant ce processus comme ScriptGetCMap qui rend les numéros de glyphes associés à une chaîne de caractères pour une police donnée. Grâce à cette fonction on peut, par exemple, vérifier si une police permet d'afficher une chaîne de caractères, en d'autres mots qu'aucun caractère n'a pas de glyphe. Si certains caractères ne sont pas pris en charge l'application pourra par exemple décider de choisir une autre police et réessayer la même opération avec cette nouvelle police.

6. Fonctionnalités et règles OpenType

6.1. Hiérarchie d'accès

On appelle « règle » une substitution (définie dans la table GSUB) ou un positionnement (défini dans la table GPOS). Le moteur de rendu filtre les substitutions et positionnements définis dans une police pour n'appliquer que ceux qui correspondent à l'écriture et à la langue utilisées. Même les fonctionnalités qui ne sont pas par nature liées à une écriture ou une langue sont structurées selon la hiérarchie suivante :

Écriture > Langue > Fonctionnalité⁽³⁹⁾ > Groupe⁽⁴⁰⁾ > Règle

Une langue est toujours associée à une écriture, elle précise une convention graphique particulière à une écriture pour la langue en question. Beaucoup de langues écrites à l'aide d'une même écriture partagent les mêmes conventions graphiques, c'est pourquoi OpenType définit également une « langue » commune ou implicite représentée par la valeur dflt. En règle générale, les fonctionnalités rangées sous dflt suffisent à représenter la quasi-totalité des langues qui utilisent cette écriture. Le développeur de police pourra modifier ce comportement typographique implicite pour une écriture en précisant des langues pour lesquelles un traitement différent doit être effectué. C'est à l'application-client de fournir la langue en vigueur pour un passage donné.

Dans l'exemple ci-dessous, on a ajouté une hiérarchie « turque » à l'écriture latine pour distinguer le type de ligatures permises en turc et exclure ainsi certaines ligatures permises pour d'autres langues utilisant l'écriture latine. On se rappellera que le turc distingue le *i* et le *ı* (i sans point), on ne peut donc escamoter le point du *i* comme on le fait habituellement en latin pour les ligatures *fi* et \square .

(39) - C'est ce que les spécifications d'OpenType appellent en anglais des « features ».

(40) - C'est ce que les spécifications d'OpenType appellent en anglais des « lookups ».

Latin <latn>
 Implicite <dflt>
 Ligatures habituelles <liga> (fonctionnalité)
 Toutes les ligatures-f (groupe GSUB)
 f f i -> ffi
 f f l -> ffl
 f f -> ff
 f i -> fi
 f l -> fl
 Turc <TRK>
 Ligatures habituelles <liga> (fonctionnalité)
 Les ligatures ff et fl (groupe GSUB)
 f f l -> ffl
 f f -> ff
 f l -> fl

Le même principe peut être utilisé pour distinguer les lettres serbes italiques des lettres cyrilliques italiques habituelles ou les formes précises des caractères CJC unifiés dans Unicode. Le composant linguistique permet en quelque sorte de désunifier les différentes formes linguistiques d'un même caractère qu'Unicode unifie à dessein par écriture. Il est important de noter que ce système ne permet pas de définir des exceptions qui seraient soustraites à la liste des substitutions implicites à l'écriture : il faut préciser toutes les fonctionnalités à effectuer pour une langue particulière, aucune des fonctionnalités implicites (dflt) ne s'appliquera à cette langue.

Il nous reste à introduire deux autres éléments de la hiérarchie : les fonctionnalités et les groupes. Les fonctionnalités correspondent à un ensemble de règles typographiques qui définissent la manière d'utiliser des glyphes pour une langue donnée. Ainsi, une police arabe pourra-t-elle définir une fonctionnalité qui précise les règles de substitution des glyphes initiaux alors qu'une police kanji pourra définir une fonctionnalité qui permet de positionner verticalement des glyphes. Une fonctionnalité peut définir des substitutions de glyphes, des positionnements de glyphes ou les deux. Comme ces fonctionnalités doivent être interprétées par un moteur de rendu (ou de façonnage pour Microsoft), il est préférable de n'utiliser que des fonctionnalités prédéfinies⁽⁴¹⁾ pour assurer une interopérabilité maximale. Enfin, un groupe rassemble une série de règles, c'est-à-dire une série de substitutions (GSUB) ou de positionnements (GPOS). Un groupe peut être utilisé par plusieurs fonctionnalités. La répartition des règles parmi les groupes est laissée au développeur.

L'exemple, tronqué ci-dessous, illustre quelques fonctionnalités et groupes associés à l'écriture arabe. Trois fonctionnalités sont brièvement illustrées : la

(41) - Yanniss Haralambous énumère ces fonctionnalités dans l'annexe D.10 de son ouvrage cité dans la bibliographie qui conclut cet article.

composition et décomposition de glyphes (*ccmp*), les formes initiales (*init*) et les ligatures obligatoires (*rli*g), d'autres fonctionnalités sont bien sûr permises, les plus fréquentes sont celles associées aux autres aspects contextuels des lettres arabes (*isol*, *medi*, *fin*a), elles sont en fait obligatoires pour Uniscribe.

Alors que la sémantique des fonctionnalités enregistrées est définie et publique, les règles précises qui constituent la mise en œuvre de ces fonctionnalités dépendent du fondeur. L'ordre des fonctionnalités est également défini pour une écriture donnée, la fonctionnalité *ccmp* est toujours la première exécutée quelle que soit l'écriture. En d'autres mots, on effectue les règles précisées par le fondeur dans la fonctionnalité *ccmp* avant de considérer toute autre fonctionnalité.

Arabe <arab>
 Implicite <dflt>
 Composition et décomposition <ccmp> (fonctionnalité)
 décomposer (groupe GSUB)
 ا → | + ء (décomposer alif hamza)
 (autres substitutions)

 composer (groupe GSUB)
 ء + ء → hamza_fathatan
 (autres substitutions)

 Formes initiales <init> (fonctionnalité)
 lam -> lam.init
 çad -> çad.init
 sîn -> sîn.init
 (autres substitutions)

 Ligatures obligatoires <rli>g (fonctionnalité)
 lam alif -> lam_alif

Bien que chaque fonctionnalité OpenType puisse en théorie⁽⁴²⁾ être associée à des langues différentes et puisse permettre de traiter différemment les glyphes de ces langues, certaines fonctionnalités prédéfinies - et donc prises en compte par de nombreuses applications - ont été explicitement conçues pour prendre en compte le niveau linguistique introduit dans la hiérarchie d'accès aux fonctionnalités. Une de ces fonctionnalités se nomme « Formes locales » (*locl*), elle permet d'associer des variantes de glyphes à une langue particulière. Un exemple typique est celui du bulgare qui écrit quelques-unes de ses lettres cyrilliques différemment des lettres cyrilliques habituelles. Une police bulgare pourra décider que les formes bulgares sont implicites et déclarer des variantes de glyphes pour le russe. Une police russe fera sans doute l'inverse.

(42) - En pratique ce n'est pas toujours le cas, certaines fonctionnalités ne devraient pas être utilisées avec certaines écritures. C'est le cas, par exemple, de la fonctionnalité « *cpsp* » - espacement des textes écrits en capitales - qui ne devrait pas être utilisé avec les écritures cursives comme l'arabe.

6.2. Intégrer le tout

Tout cela est bien beau. Mais comment tous ces différents concepts et fonctions s'intègrent-ils en pratique ? Tout commence par la détection de l'écriture choisie. Dans le cas d'Uniscribe, c'est le rôle de la méthode ScriptItemize. Ensuite, le moteur de rendu détermine la langue utilisée pour chaque segment analysé par ScriptItemize, celle-ci peut être précisée par du balisage (l'attribut `xml:lang` en XML par exemple) ou, en l'absence d'un tel balisage, correspondre à la langue du profil de l'utilisateur actuel.

À partir de maintenant le moteur de rendu peut commencer à effectuer les fonctionnalités définies dans la police choisie. Toutefois, certaines des règles peuvent nécessiter l'intervention de l'utilisateur (pour lui permettre par exemple de choisir la formation de ligatures optionnelles dlig comme les 2 et 1 en latin). Il est important de noter que l'ordre d'application des fonctionnalités est fixe pour une écriture donnée. Dans le cas de l'arabe, l'ordre standard⁽⁴³⁾ est précisé dans le tableau 3 ci-après. Certaines des fonctionnalités énumérées ne s'appliquent qu'à certaines langues qui utilisent l'écriture arabe.

Fonctionnalité	Description	Table	Obligatoire
Formes linguistiques			
ccmp	Composition et décomposition	GSUB	
isol	Forme isolée du glyphe	GSUB	
fina	Forme finale du glyphe	GSUB	X
medi	Forme médiale du glyphe	GSUB	X
init	Forme initiale du glyphe	GSUB	X
r lig	Ligature obligatoire	GSUB	X
calt	Alternatives de jonction	GSUB	
Formes typographiques			
liga	Ligatures habituelles	GSUB	
dlig	Ligatures optionnelles	GSUB	
cswh	Lettres à paraphe contextuels	GSUB	
mset	Positionnement de diacritiques à l'aide de substitution	GSUB	
Positionnements			
curs	Positionnement cursif	GPOS	
kern	Crénage par paires	GPOS	
mark	Positionnement de diacritiques par rapport au glyphe de base	GPOS	
mkmk	Positionnement de diacritiques par rapport à un autre diacritique	GPOS	

Table 3. - *Ordre d'exécution des fonctionnalités par le moteur de façonnage arabe*

(43) - Précisé ici : <<http://www.microsoft.com/typography/otfntdev/arabicot/features.htm>>.

6.3. Rôle du moteur de rendu

Prenons un extrait assez typique de la fonctionnalité `init` qui fournit la forme initiale de glyphes arabes :

`uni062C ٢ → uni062C.init(44) ٢`

`uni0628 ٨ → uni0628.init ٨`

La première ligne de cet extrait se lit de la façon suivante : quand le moteur de façonnage trouve un djîm non contextualisé, le remplacer par sa forme initiale. Mais voilà, rien n'indique dans la police quand il faut appliquer cette fonctionnalité ! C'est le moteur de façonnage (cf. ④ de la figure 8) qui détermine quand il invoquera les substitutions liées à une fonctionnalité particulière. Pour ce faire, il doit pouvoir découper un texte en « mots » pour l'écriture arabe et identifier les caractères qui se trouvent en positions initiales.

6.4. Ordre d'exécution des fonctionnalités et des groupes

D'emblée, rappelons que le moteur de rendu effectuera d'abord toutes les substitutions (GSUB) avant de passer au traitement des positionnements (GPOS).

L'ordre dans lequel les fonctionnalités sont exécutées dépend du moteur de rendu. Dans le cas des moteurs de façonnage de Microsoft, il arrive que certaines fonctionnalités GSUB soient traitées séquentiellement alors que d'autres sont effectuées simultanément. En règle générale, les moteurs de façonnage distinguent entre les comportements typographiques essentiels des écritures et les raffinements typographiques. Les fonctionnalités essentielles s'effectuent avant les autres. Dans le cas de l'écriture latine, la fonctionnalité `ccmp` s'appliquera avant `liga` ou `dlig`, parce que `ccmp` (par exemple, la composition de caractères par l'adjonction de diacritiques) est considéré comme essentiel alors que la formation des ligatures est considérée comme du peaufinage typographique.

Pour les écritures dites complexes (par exemple la dévanâgarî, l'arabe, le khmer ou le thaï), les fonctionnalités sont habituellement exécutées séquentiellement alors que, pour les écritures prétendument simples (par exemple, latine ou cyrillique) elles sont habituellement exécutées simultanément.

L'exécution simultanée de plusieurs fonctionnalités signifie que les groupes de règles respectifs de ces fonctionnalités seront exécutés dans l'ordre de leur définition dans la liste générale des groupes de règles. Cette liste énumère tous les groupes de règles de toutes les fonctionnalités.

(44) - Nous adoptons ici pour le nom des glyphes la convention PostScript et celle du langage de description de fonctionnalités OpenType d'Adobe. Voir l'article sur la police Hapax Berbère de ces mêmes actes pour plus de détails.

Pour clarifier les choses, imaginons une police qui définit trois fonctionnalités (A, B et C) et leurs groupes de règles :

```
FonctA
  GroupeA1
  GroupeA2
  GroupeX
FonctB
  GroupeB1
  GroupeB2
  GroupeX
FonctC
  GroupeC1
  GroupeC2
  GroupeC3
  GroupeX
```

Le groupe X ci-dessus est un groupe partagé par plusieurs fonctionnalités. Comme nous le disions plus haut, il existe une liste de tous les groupes indépendante de la liste des fonctionnalités. L'ordre des règles dans cette liste est indépendant de l'ordre des fonctionnalités. Supposons que la liste des groupes est ordonnée de la façon suivante :

```
GroupeA1
GroupeB1
GroupeC1
GroupeA2
GroupeB2
GroupeC2
GroupeC3
GroupeX
```

Maintenant, si le moteur de façonnage exécute séquentiellement les fonctionnalités A et C, tous les groupes associés à FonctA seront d'abord traités avant que tous les groupes associés à FonctC le soient. Ce qui donne dans l'ordre :

```
GroupeA1
GroupeA2
GroupeX
GroupeC1
GroupeC2
GroupeC3
GroupeX
```

Si les fonctionnalités A et C s'exécutent simultanément, les groupes s'exécuteront alors dans l'ordre suivant :

```
GroupeA1
GroupeC1
GroupeA2
GroupeC2
GroupeC3
GroupeX
```

6.5. Ordre d'exécution des règles

Il est important de signaler que le moteur de rendu applique d'abord un groupe de règles à tous les glyphes d'un passage avant de passer au groupe de règles suivant. Pour chacun des glyphes du passage, on passe en revue toutes les règles du groupe jusqu'à ce qu'une règle s'applique. Si c'est le cas, on exécute la transformation décrite par cette règle et on passe au glyphe qui suit le contexte d'application de la règle (représenté à la gauche des flèches), sinon on laisse le glyphe en question intact et le moteur de rendu passe au glyphe suivant.

Ce qui donne en pseudocode :

```
pour chaque groupe dans les fonctionnalités sélectionnées faire
  glyphe ← premier glyphe du passage;
  tant que il reste des glyphes faire
    règle_exécutée ← faux;
    règle ← première règle du groupe;
    tant que il reste des règles et pas règle_exécutée faire
      si condition d'exécution de règle est remplie alors
        exécuter règle;
        glyphe ← glyphe suivant le contexte d'exécution;
        règle_exécutée ← vrai;
      sinon
        règle ← prochaine règle du groupe;
    si pas règle_exécutée alors
      glyphe ← prochain glyphe du passage;
```

Ainsi, si on remplace « f f » par « f_f » le glyphe suivant sera celui qui suit « f f » et il n'est pas possible de remplacer « f_f » dans cette itération. Pour contourner ce comportement, il suffit de définir un autre groupe (car on itère sur tous les glyphes pour un groupe), le premier groupe produira le « f_f », le second pourra modifier le « f_f ».

Soit le passage « ...raffiné... » Pour lequel on désire remplacer le « ffi » par une ligature. On pourrait a priori penser que deux règles de substitution simples comme celles ci-dessous feraient l'affaire.

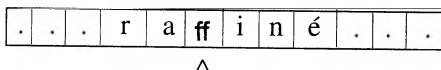
```
Latin <latn>
  Implicite <dflt>
    Ligature <liga> (fonctionnalité)
      ligatures_ff_ff (groupe)
        f f → ff
        ff i → ffi
```

Imaginons que le moteur de rendu considère le glyphe correspondant au premier « f » :

.	.	.	r	a	f	f	i	n	é	.	.	.
---	---	---	---	---	---	---	---	---	---	---	---	---

^

Il trouve une règle qui s'applique (« f f » → « □ »). Après la substitution, le glyphe courant devient le i, le glyphe qui suit le contexte d'exécution de la règle, à savoir « f f ». Schématiquement, on a donc :



Mais voilà la seconde règle de ce groupe ne s'exécutera pas, car la première règle d'un groupe, dont la condition d'exécution est remplie, gagne et met fin à la boucle. Le résultat final ne contiendra donc que la ligature «□». Pour pallier ce problème, il suffit d'introduire un deuxième groupe dans la même fonctionnalité :

```

Latin <latn>
  Implicite <dfit>
    Ligature <liga> (fonctionnalité)
      ligature_ff (groupe)
        f f → ff
      ligature_ffi (groupe)
        ff i → ffi
  
```

On aurait évidemment pu, dans notre cas, n'avoir qu'une seule substitution qui remplaçait « f f i » par \square , si on ne désirait effectuer que ce type de ligature et ne pas traiter la formation des autres ligatures de cette famille (f f → ff, f i → fi). Quand on veut traiter cette famille de ligatures dans un même groupe, il est important de préciser en premier lieu dans un groupe les substitutions les plus longues. En effet, si on définit d'abord dans un même groupe une substitution plus courte comme dans les règles ci-dessous,

```

f f → ff
f f i → ffi
  
```

il est facile de vérifier que l'on n'obtiendra jamais la ligature ffi puisque les f f seront d'abord changés en ligature ff et que la condition d'exécution de la règle suivante ne sera jamais vraie, puisqu'il ne restera plus de f f dans le texte d'origine.

6.6. Exécution parallèle et simultanée des fonctionnalités et règles

Revenons un instant sur l'ordre d'exécution des fonctionnalités et des règles dans OpenType. Si on considère le pseudocode présenté ci-dessus comme une méthode `appliquerRègles()`, membre d'une classe passage qui contient le passage à rendre, méthode à laquelle on passera en arguments les fonctionnalités à appliquer, on peut alors représenter l'ordre d'exécution des fonctionnalités et de leurs règles respectives de la façon suivante pour les écritures dites simples :

```

passage.appliquerRègles({ccmp, rlig, ...}) # application simultanée des
# fonctionnalités GSUB ccmp, rlig,
...
...
passage.appliquerRègles({kern, mark, mkmk, ...}) # simultanée GPOS
  
```

Et pour les écritures complexes :

```
passage.appliquerRègles ({ccmp}) # GSUB séquentiel  
passage.appliquerRègles ({rlig}) # GSUB séquentiel  
...  
passage.appliquerRègles ({kern}) # GPOS séquentiel  
passage.appliquerRègles ({mark}) # GPOS séquentiel
```

Maintenant, cette distinction entre écritures simples et complexes n'est peut-être pas opportune car toutes les écritures - même la latine quand on considère une mise en œuvre perfectionnée - sont complexes, c'est la prise en charge par le moteur de façonnage correspondant qui peut être simple ou complexe. On peut se demander si cette différence n'est donc pas gratuite et ne complexifie pas la description et le fonctionnement des moteurs OpenType.

7. Conclusion

Nous avons vu dans cet article comment les caractères d'Unicode sont transformés en glyphes pour être finalement rendus à l'écran ou sur tout autre dispositif comme une imprimante. Dans cette brève introduction aux technologies de polices de glyphes nous avons voulu fournir un aperçu de ces différents processus et bien faire comprendre les implications liées à des techniques aussi pointues qu'OpenType. En particulier, il ne suffit pas d'écrire quelques substitutions dans une police pour qu'elles soient prises en compte. En effet, les fonctionnalités sont définies par les fabricants de moteur de rendu (p.ex. Uniscribe sur Windows), le concepteur de police devra donc respecter les noms de ces fonctionnalités s'il veut utiliser ce moteur. Il devra également prendre en compte l'ordre défini dans lequel ces fonctionnalités sont exécutées pour une écriture donnée.

Enfin, on ne peut que vivement recommander la lecture de l'excellent livre de Yannis Haralambous mentionné ci-dessous à tous ceux qui désirent comprendre dans le détail les différentes techniques liées aux polices de glyphes, y compris celles du Macintosh (AAT) ou de TeX (Metafont).

Ressources

- ❖ Fontes et codages, par Y. Haralambous, O'Reilly France, 2004, ISBN 2-84177-273-X
- ❖ Unicode en français: <<http://hapax.qc.ca>>
- ❖ Listes de diffusion et forum de discussion
 - ❖ Typo : <<https://www.irisa.fr/wws/info/typographie>>
 - ❖ Unicode Afrique : <<http://fr.groups.yahoo.com/group/Unicode-Afrique/>>
 - ❖ Unicode fr : <[nntp://fr.comp.normes.unicode](mailto://fr.comp.normes.unicode)>
- ❖ FontForge (logiciel libre de création de polices) :
<<http://fontforge.sourceforge.net/>> (version française existe)
- ❖ Babelmap (outil de visualisation des caractères) :
<http://hapax.qc.ca/BabelMap_fr.html>

إلى الكتب المطبوعة. وبهذا كان السباعي الصوت الوحيد الذي أجهر رأيه في معارضة المطبعة، وإن كان صوته جاء متأخراً حتى العقد الثامن من القرن التاسع عشر، بعد أن خرجت المطبعة من إدارة المخزن إلى إدارة الخواص.

ويمكن أن نلاحظ على هذه المطبوعات الاستهلاكية، التي فاق مجموع نسخها ألف وخمسمائة نسخة، والتي حظيت بالأولوية في الطبع، المبالغة في إتقانها تصحيحاً وطبعاً وورقاً. يقول عنها ابن زيدان: «هذه الكتب طبعت طبعاً متقناً في غاية الجودة يخيل للرأي أنه خط يد أكبر فنان خطاط، وطالما حاول الطابعون محاكاته فلم يكن يحاكيه طابع حتى ما طبع بالمطبعة الحجرية المصرية في العصر الحاضر والعيان أكبر شاهد وأوضح برهان»⁽⁹⁹⁾.

وعلى كل فإن دخول هذا الفن الجديد للكتابة مهد السبيل للتوجه نحو طرق جديدة في المعرفة لم تعد تعتمد على الذاكرة فقط بل أصبحت تركز على التفكير والفهم، وقلّت شيئاً فشيئاً من هيمنة العلماء على ميدان المعرفة.

وخلاصة القول فالطباعة الحجرية على الرغم من ضعف منشوراتها ومحدودية موضوعاتها، لم تكن مجرد آلة (محايدة) بل كانت لها أبعاد وانعكاسات اجتماعية واقتصادية وثقافية. حيث ساهمت في توفير الكتب مع رخص ثمنها بالمقارنة مع المخطوطات، فلم يعد العلم مقتصراً على النخبة فقط، بل ساعدت على توسيع قاعدة المتعلمين، وانتقال المعرفة وذيوع العلم، ولهذا سماها المغاربة بألة (نشر العلم).

(99) - ابن زيدان، النهضة العلمية، مرجع سابق، ورقة 72.

وهنا لابد من الوقوف عند النقطة المتعلقة بموقف العلماء من تبني المخزن لألة الطباعة، بصفتهم أعلى سلطة فكرية بالبلاد .

لقد عرف المغرب تقليداً ثابتاً يتمثل في استفتاء السلطان للعلماء بشأن جميع مبادراته، حيث سلك معهم أسلوب التشاور، حتى يستفيد من مواقفهم ويزكي بالتالي أعماله، بصفتهم أصحاب الحل والعقد، وكان العلماء بالنسبة للسلطين دعامة من دعائم شرعيتهم، فقد كانوا «الفئة الوحيدة التي كان السلطان يرى ضرورة تبرير أعماله أمامها»⁽⁹⁷⁾، وكانوا بصفة عامة يحظون بتقدير الرعية، الذين ينظرون إليهم بنوع من التبجيل والتقديس، لذلك كان المخزن يعيرهم العناية والاعتبار .

ويظهر أن قرار السلطان بتبني المطبعة منذ وصولها إلى ميناء الصويرة وتشغيلها بمدينة مكناس، صادف قبولاً لدى معظم العلماء، فلم يظهر حينها أي احتجاج أو اعتراض يذكر، بعكس ما كنا نجد دائماً أصوات المعارضين والمؤيدين، واختلاف فتاويهم عند إقدام المخزن على إدخال أي تجديد أو تغيير، خصوصاً وأن آلة الطباعة تدخل ضمن التجديد، ولها اتصال مباشر بميدان العلم، وهذا ما يؤكد بأن العلماء أدركوا أهمية الطباعة في نشر العلم والمعرفة، ودورها في تكثير الكتب، لذا حاولوا الاستفادة من تقنياتها، فشرعوا في مساندة السلطان في مشروعه المطبعي. ويظهر ذلك واضحاً من خلال كتابات كل من محمد العربي المشرقي، وعبد السلام اللجائي، ومحمد الروندة، التي كانت كلها تمجيداً وتعظيماً لعمل السلطان، وإقراراً بمزايا التقنية الجديدة للكتابة . وربما قبول العلماء للمطبعة، وهذا التأييد والاستحسان من طرفهم، يرجع لكون اختيار الروداني لمطبعة حجرية، سوف لا يحدث قطيعة مع الماضي، وبالتالي سوف لا يشكل أي تهديد لفن الخط المغربي التقليدي، فلم يعتبروه تجديداً، وإنما هو تكريس للتقليد .

والصوت الوحيد الذي عارض نشر الكتب بواسطة المطبعة، تمثل في شخص الفقيه محمد ابن إبراهيم السباعي المراكشي (ت 1332 هـ / 1914 م)، وذلك من خلال رسالة مدح فيها القلم، وحض على الكتب الخطية والاعتناء بها، لكونها فيها البركة وعليها الاعتماد، كما حذر من المطبعة لأنها تمتهن الكتاب وتسلبه قدسيته، كما أنها سبب في تقليل الهمم وعدم حفظ العلم ونسيانه⁽⁹⁸⁾ . وربما كان السباعي يرى أن المطبعة ستزاحمه، وتفقده مكانته، بانصراف طلبته عنه

(97) - Laroui (A), Les Origines sociales et culturelles du Natrionalisme marocain Maspero, 1977 , p. 98.

(98) - عبد الحفيظ الفاسي، رياض الجنة، المطبعة الوطنية، الرباط، 1350 هـ / 1931 م، ج 1، ص. 57.

المعلومات يقول الأزرق أخبره بها الطيب المصري، دون أن يشير إلى الإهداء والتعويضات كما جاء في بعض الروايات، كما أنه تحاشى ذكر الحيازة أو المصادرة كما جاء عند المختار السوسي، وإن كان حديثه يوحي بذلك.

ويضيف فوزي عبد الرزاق بأن مصادرة المطبعة كان أمراً متوقفاً لوجود كل من الصفار والعمراوي في حاشية السلطان، حيث سبق لهما أن اطلعا على أسرار الطباعة بفرنسا وشاهدا استعمالها تحت إشراف الدولة، لذا أوعزا إلى السلطان بأنه أحق بامتلاك هذه الآلة حتى تكون تحت مراقبة وإشراف الدولة.

فكل من شهادة الطيب الأزرق، وإشارة المختار السوسي، تثبتان مصادرة المخزن لمطبعة القاضي الروداني، هذا الأخير الذي توفي في نفس السنة ببلده تارودانت، ربما دون أن يشاهد أول ثمرات إنتاج مطبعته، وربما فشل مشروعه المطبعي، كان وراء ما أصابه من إحباط واكتئاب أدى إلى وفاته بعد وقت وجيز. ومما جاء في رثاء الشيخ أحمد الجشتيمي⁽⁹⁴⁾ للقاضي الروداني قوله:

نـة" بحر علم لافظاً أنفس الدر	إمام الهدى التلمي من كان في "ردا"
يشاء على ما شاء عن قهره يجري	(وبعد) فحكم الله في خلقه كما
بحمل أخ كالطود جبراً إلى القبر	وإن جل ما يعرفو كما جل ما جرى
رضا الطيب المشهور كالروض بالزهر ⁽⁹⁵⁾	إمام الهدى والعلم والحلم سيدي الـ

وباختصار فهذه الشهادات جلتها تؤكد حقيقة واحدة، وهي أن الطيب الروداني، نقش اسمه في سجل التاريخ المغربي بوصفه صاحب مبادرة رائدة، تتجلى في إدخال آلة الطباعة إلى المغرب. وتتفق جل الإشارات السابقة، إلى استقرار المطبعة أول الأمر بمكناس، حيث تم طبع أول كتاب وهو "الشمائيل المحمدية" لأبي عيسى الترمذي، ثم استقرت بعدها بمدينة فاس. وقد جاء في وثيقة الروندة قول السلطانص: «إن الناس أخبروني أن الأليق بالمطبعة هو فاس لكثرة علمائها وكتبها»⁽⁹⁶⁾، أي أن المطبعة وظفت لخدمة العلماء، ولتوفير ما يحتاجونه من مؤلفات.

(94) - انظر ترجمته عند:

- المختار السوسي، سوس العالمية، مؤسسة بنشرة للطباعة والنشر، الدار البيضاء، 1404 هـ/ 1984 م، ص. 205.

(95) - انظر النص الكامل للقصيد عند:

- المتوكل عمر الساحلي، المعهد الإسلامي بتارودانت والمدارس العلمية العتيقة بسوس، دار النشر المغربية، الدار البيضاء،

1985، ج 1، صص. 319-320.

(96) - محمد عبد السلام الروندة، مرجع سابق.

بالعقد، وهذا يعني بالنسبة إليه أنه أصبح يشتغل لحساب الدولة التي حلت في ظروف نجلها، محل السيد الطيب»⁽⁸⁹⁾.

فغياش لم يستطع الحسم في ظروف انتقال المطبعة من الروداني إلى المخزن. ويمكننا أن نضع احتمالاً آخر يرمي إلى أن الروداني عند وصوله بالمطبعة إلى ميناء الصويرة، واجه صعوبات مع موظفي المرسى، فأشار عليه البعض بإهدائها إلى السلطان، فحملها صحبة المعلم المصري إلى مكناس، حيث كان يوجد السلطان حينئذ، وقدمها هدية له، وهذا ما يقترب من رأي المنوني الذي يقولص: «إن القاضي الروداني كان في أول الأمر يستخدم المطبعة في اسمه، غير أنه لا يلبث أن يقدمها هدية للسلطان "محمد الرابع" وقد وقع ذلك بمجرد رجوعه إلى المغرب»⁽⁹⁰⁾.

وفي اتجاه آخر نجد المختار السوسي يشير إلى أن القاضي الروداني «اشترى في حجته المطبعة الحجرية الفاسية الأولى من مصر يديرها لنفسه ثم حازها منه سنة 1281 هـ، السلطان سيدي محمد بن عبد الرحمان فنقلت من الصويرة إلى مكناسة»⁽⁹¹⁾.

وكلمة حيازة هنا تعني مصادرة، بمعنى أن المخزن صادر المطبعة باسم السلطان. وفي نفس السياق يتحدث فوزي عبد الرزاق عن مصادرة المخزن للمطبعة⁽⁹²⁾، معتمداً في ذلك على مضمون وثيقة مستمدة من كناشة محمد بن عبد السلام الرنودة (ت 1365 هـ / 1945 م)⁽⁹³⁾. وهي عبارة عن حديث أجراه القاضي الرنودة مع الطيب الأزرق أشهر طابع مغربي، استهله بقولهاص: «بيوم الجمعة 4 رمضان 1336 هـ، أخبرني الأشيب البركة السيد الطيب الأزرق الفاسي، وكان يباشر المطبعة بفاس حيث أنشأت زمان السلطان سيدي محمد بن عبد الرحمان العلوي». فالحديث بين الرجلين كان يدور حول كيفية دخول المطبعة إلى المغرب، تحدث خلاله الطيب الأزرق عن وصول آلة الطباعة إلى ثغر الصويرة وبصحبته طابع مصري، ثم أشار إلى إخبار أمين المرسى المدعو الفرنساوي القباج، قائد المنطقة عبد الله وبه السوسي بأمرها. فكاتب هذا الأخير السلطان يعلمه بأمر المطبعة، ويستفسره في نفس الوقت عن الإجراء الواجب اتخاذه في موضوع هذه الآلة الغريبة. فأصدر السلطان أوامره بإحضار المطبعة ومعها الطابع المصري إلى مكناس. وهذه

(89) - نفسه، ص. 134.

(90) - المنوني، مرجع سابق، ص. 206.

(91) - المختار السوسي، مرجع سابق، ص. 120.

(92) - فوزي عبد الرزاق، مرجع سابق، ص. 141.

(93) - الوثيقة من كناشة العلامة محمد بن عبد السلام الرنودة قاضي الرباط، ثم وزير العدلية والمعارف الإسلامية في عهدي مولاي يوسف ومحمد الخامس. وتوجد الوثيقة في حوزة حفيده الصديق الرنودة الكاتب العام للأمانة العامة لرابطة علماء المغرب، وقد سلمني صورة منها.

من مصر، بينما المطبعة السلكية تتطلب تجهيزات وآلات باهظة الثمن تستورد من أوروبا التي كان المغاربة يمتقونها آنذاك.

يظهر أن هذه الأسباب جميعها كانت حاضرة في ذهن الطيب الروداني عند عزمه على اقتناء المطبعة، فعرف مزايا المطبعة الحجرية، وأدرك أنها التقنية الأكثر ملاءمة لذوق المغاربة، لكونها سوف لا تحدث قطيعة مع الماضي بل ستضفي عليه فوائد جمّة. لذا أقدم على شرائها، وتعاقد مع طابع مصري يدعي محمد إبراهيم القيانى لمرافقته إلى المغرب، لتشغيل المطبعة ببلده تارودانت سنة كاملة بشروط مفصلة في العقد.

فهذا العقد يلقي أضواء كاشفة على نوايا الروداني، وعلى المكان الذي كان يرغب أن يشغل فيه مطبعته. فكما ينص العقد، يلتزم الطابع المصري أن يأتي برفقة الروداني إلى مدينته "رودانة" ويشتغل عنده في المطبعة مدة سنة كاملة، تبتدئ من ربيع الأول 1281 هـ/ يونيو 1864 م، وتنتهي صفر الخير 1282 هـ/ يوليو 1865 م. ومقابل ذلك يتحمل الروداني القيام بجميع لوازم القيانى من مأكّل ومشرب وملبس حسب رغبته، مع أجرة شهرية بمقدار مائتي قرش مصري. كما نص العقد على أنه قابل للتجديد عند انتهاء السنة.

فبنود العقد تظهر بأن الروداني اشترى المطبعة لنفسه، وكان يرغب بتشغيلها ببلده تارودانت، بدليل تعهده بتحمل كل نفقاتها ونفقات الطابع، ولاشك أنه كان يعلم قبل توقيع العقد ما تحتاجه المطبعة من لوازم، وما يحتاجه تشغيلها من نفقات للمساعدين وغيرهم. ولاشك أن قدراته المادية كانت قادرة على استيعاب كل هذه النفقات.

إن نص العقد المبرم بين الرجلين، تعهدا فيه بوضوح بالتزامات فرد نحو فرد، فلم ترد فيه مطلقاً كلمة السلطان أو المخزن، فكيف يفسر انتقال المطبعة من ملكية الروداني إلى المخزن؟ وبالتالي كيف تم تحويلها من مشروع فردي إلى مشروع مخزني رسمي؟

هذه النقطة يكتنفها الكثير من الغموض، فبعض الذين تحدثوا عن المطبعة، أشاروا إلى أن الروداني عند وصوله إلى ميناء الصويرة (القريب من بلده تارودانت) أهدى المطبعة للسلطان محمد الرابع، الذي كان يقيم حينها بمدينة مكناس⁽⁸⁷⁾. حيث يشير عياش إلى أن الروداني ربما اشترى المطبعة لحساب السلطان بصفته موظفاً تابعاً للمخزن المغربي⁽⁸⁸⁾. ثم يستدرك قائلاً: «إن أمناء الصويرة تكفلوا بالقيانى في المدينة في الوقت الذي كان ما يزال العمل فيه جارياً

(87) - ورد هذا عند اللجائي، وعند المنوني، مرجع سابق.

(88) - عياش، مرجع سابق، ص. 133.

بالكتب من الشرق خسروا بها لعدم إقبال الناس عليها ليس لغلاء أثمانها وإنما لعدم معرفة الأهلين قراءتها. وليس من اختلاف بين اللغتين الشامية والمغربية إلا أن حرف الفاء عند المغاربة ينقط من تحت وللقاف نقطة واحدة. وأما شكل الحروف الخطية فيختلف والمغربي أشبه بالكوفي، وأهالي المغرب لا يستطيعون أن يقرأوا الكتابة الشامية»⁽⁸³⁾.

وهكذا نرى أن الروداني في اختياره للطباعة الحجرية حافظ للمغاربة على تراثهم الذي كان يكتسي في نظرهم طابعاً مقدسياً، والمتمثل في الخط المغربي التقليدي، لأن المطبوعات الحجرية تشبه في مظهرها المخطوطات تماماً. فالقراء المتعودون على قراءة المخطوطات لن يجدوا أي صعوبة في قراءة مطبوع حجري.

ب . السبب الاجتماعي: إن الروداني في اختياره للطباعة الحجرية، حافظ على

وجود طبقة مهمة داخل المجتمع المغربي، ونعني بها طبقة النساخ أو الوراقين، الذين كانوا يعيشون من نسخ الكتب. فالأمر هنا مرتبط بعدد الناسخين ومكانتهم الاجتماعية وقدرتهم على التصدي لآلة الطباعة، ويبدو أن عددهم هام ولهم وزن في المجتمع المغربي. ففي العصر العلوي ازدهرت هذه الحرفة، فأنشأ المولى سليمان ديواناً للوراقة⁽⁸⁴⁾، وازدادت مكانتهم في عهد محمد الرابع، وبلغوا الشأو البعيد أيام مولاي الحسن، هذا الازدهار كان له انعكاس على مدخول الوراقين من النساخة، فالحاج إدريس بن إدريس العمراوي اكتسب من مردود هذه المهنة عقاراً⁽⁸⁵⁾، وعن قيمة هذه الحرفة في المجتمع المغربي يذكر أبو حامد الفاسي: «أنها من أحسن الحرف والأشغال، لما فيها من نشر العلم وتخليده، وقد احترف بها كثير من المقتدى بهم»⁽⁸⁶⁾. ومن الواضح أن خوف هؤلاء من انهيار مهنتهم، هو ما يفسر نفورهم من المطبعة، وهو ما يوضح أيضاً سبب اختيار الروداني للمطبعة الحجرية لأنها تعتمد على النسخ، وبالتالي ستحافظ لأصحاب هذه الحرفة على مصدر رزقهم.

ج . السبب الاقتصادي: فالمطبعة الحجرية غير مكلفة كثيراً مثل غيرها، فهي لا

تتطلب تجهيزاً كثيراً، ولا تستدعي إنشاء مؤسسة كبيرة. فهي تعتمد على ثلاثة أفراد فقط (الطابع والناسخ والمصحح) بل يمكن لشخص واحد القيام بجميع الأعمال من نسخ وطباعة وتصحيح وتوزيع. كذلك بعض المواد اللازمة للمطبعة الحجرية يمكن صنعها بالمغرب أو جلبها

(83) - أسعد كرم، مرجع سابق، ص. 135.

(84) - محمد المنوني، تاريخ الوراقة المغربية، مطبعة النجاح الجديدة، الدار البيضاء، 1991، ص. 170.

(85) - محمد غريط، فواصل الجمال، المطبعة الجديدة، فاس 1347 هـ / 1927 م، ص. 142.

(86) - المنوني، المرجع السابق، ص. 12.

قبل معرفة الأسباب لابد من التعرف على طريقة الطبع الحجري أو الليثوغرافي. فكلمة ليثوغرافي Lithography تعود إلى أصل إغريقي، وهي مكونة من مقطعين حجر Litho وكتابة Graph، أي الكتابة على الحجر. اخترع هذه الطريقة الألماني ألويز سنيفلدر Aloise Senefelder سنة 1796 م. والحجر المستخدم فيها مركب من الكلس والطفل والرمل، ينحت جيداً ويصقل بالرمل الناعم وينظف جيداً. أما تقنية الطباعة بالحجر حسب الطريقة التي كانت متبعة في المغرب فهي كالآتي:

كان الناسخ يكتب على الورقة كتابة مستقيمة غير مقلوبة، بعد هذا توضع اللوحة التي يراد نقل الكتابة إليها في مكبس بعد أن يحمى، ثم تبسط الورقة عليه بطريقة عكسية، بحيث يقع وجهها المكتوب على وجه اللوحة الحجرية، ويضغطان مراراً كثيرة، فتلتصق الورقة باللوحة، ثم يرطب ظهر الورقة، وتدار اللوحة وتضغط مراراً كثيرة أيضاً، وترطب مرة أخرى وتفرك بالأنامل لكي يسهل نزعها عن اللوحة، فتتزع عنها تاركة الكتابة عليها، ثم يصب على اللوحة قليل من الصمغ، وتبل خرقة بقليل من حبر الطباعة ويمسح بها فيلصق الحبر حيث كانت الكتابة. وحينما تبرد جيداً يصب عليها مزيج يسير من الحامض ومذوب الصمغ مرة أو أكثر، حتى إذا تشفت توضع في المطبعة، وتبلل بالماء ثم الحبر، ثم يطبع الورق عليها حسب العدد المطلوب، ويعاد تبليها بالماء وتحبيرها قبل طبع كل ورقة⁽⁸²⁾.

وهكذا نرى بأن طريقة الطبع على الحجر لم تكن يسيرة بل كانت معقدة شيئاً ما. فلماذا اختارها الروداني وفضلها على المطبعة التيبوغرافية؟

يمكن أن نحصر ذلك في عدة أسباب وهي:

أ . السبب الديني والثقافي : فالروداني بحكم تكوينه الديني، رأى في الطباعة الحجرية، أكثر ملاءمة لذوق العلماء التقليديين، لأنها تمتلك الخصوصيات التي تسمح بإصدار كتب مشابهة تماماً للنماذج المخطوطة، لكونها تحافظ على أصالة الخط المغربي ومميزاته الخاصة، ككتابة الفاء بنقطة في الأسفل، والقاف بنقطة واحدة أعلى، ولهذا السبب كان المغاربة يفضلون المخطوط على المطبوع المجلوب من الشرق. ونستشهد هنا بشهادة أحد الكتاب الشرقيين: «ولما كانت المطابع معدومة من البلاد كانت صناعة النسخ رائجة والكتب الخطية هي المعول عليها حتى أنه يسهل على المطالعين قراءتها أكثر من قراءة الكتب المطبوعة والذين تاجروا

(82) - مقال الليثوغرافيا أو طبع الحجر، مجلة المقتطف، الجزء الرابع من السنة السادسة، شتبر 1881، ص 193 - 194 .

فإذا فهمنا شخصية وأخلاق الروداني، يمكننا إدخال جلبيه لآلة المطبعة، ضمن أعماله الخيرية والعلمية التي أراد بها نشر العلم في منطقتة سوس.

ويظهر أن اقتتاه المطبعة لم يأت صدفة، بل دليل المبالغ المالية التي حملها معه إلى الحج من شراء المطبعة، ودفع مستحقات نقل الحمولة من ميناء الإسكندرية إلى ميناء الصويرة، بالإضافة إلى ما أنقده للطبيع المصري لحل مشاكله المادية، كل هذا يظهر أن الروداني قبل ذهابه للحج كان عازماً على شراء المطبعة، فربما سمع عنها من أحد الحجاج، أو اطلع على إنتاجها من خلال الكتب التي كانت تدخل إلى المغرب عن طريق الحجاج، خصوصاً العلماء منهم⁽⁷⁸⁾، أو ربما علم بخبر المطبعة الفرنسية بالجزائر أثناء مزاولته القضاء بوجدة قبل أن يتولى قضاء تارودانت⁽⁷⁹⁾.

تفيدنا جميع الروايات بأن القاضي محمد الطيب الروداني، قصد الحج سنة 1280 هـ/ 1864 م، وعند رجوعه مر بمصر - كدأب المغاربة - واشترى مطبعة حجرية جلبها إلى المغرب، كما استقدم معه طبيباً مصرياً يسمى محمد القيانى ليشتغل بها⁽⁸⁰⁾.

وليست لدينا أي وثيقة أو إشارة إلى الثمن الذي اشترى به الروداني المطبعة، ولا الجهة التي اقتناها منها، وإن كنا نعتقد أنه اشتراها من مطبعة بولاق، التي كان بها يومئذ قسم للطباعة الحجرية، ولكون الطبيع المصري كان يشتغل بها قبل تعاقد مع الروداني، وربما يكون وكيله ووسيطه في ذلك، المطبعي محمد هاشم المغربي، الذي كان يحظى بامتياز ومعاملة خاصة، إذ سمح له بفتح مطبعة بحارة برجوان بالقاهرة سنة 1861 م، دون استئذان ودون احترام للشروط المنصوص عليها في القانون الخاص بالمطابع بمصر⁽⁸¹⁾.

على كل هذه تبقى استنتاجات في غياب أية معلومات عن اتصال الروداني بالطباع المغربي بمصر. فكل الوثائق تبتدئ بالعقد المبرم بين الروداني ومحمد القيانى بتاريخ 14 ربيع الأول 1281 هـ/ 11 يونيو 1864 م. هذا العقد يظهر بوضوح تاريخ دخول المطبعة الحجرية إلى المغرب. وهنا نتساءل لماذا وقع اختيار الروداني على مطبعة حجرية، تقنياتها معقدة جداً بالقياس مع المطبعة التبوغرافية، أو ما تسمى في المغرب بالسلكية؟

(78) - سبق الإشارة إلى ذلك عند الحديث عن رحلة الفيغاني، فربما يكون الروداني قد اطلع على هذه الرحلة، أو التقى بصاحبها، حيث لا يفصل بين رحلته ورحلة الفيغاني سوى ست سنوات.

(79) - المنوني، مرجع سابق، ص. 206، الهامش رقم 1.

(80) - انظر ذلك في:

. كناشة بليمي، مرجع سابق؛ - المنوني، مرجع سابق، ص. 206؛ - اللجائي، مرجع سابق، ورقة 233؛ - فوزي عبد الرزاق، مرجع سابق، ص. 138.

(81) - أبو الفتوح رضوان، تاريخ مطبعة بولاق، مرجع سابق، ص. 387 - 389، أورد نص الامتياز الذي حظي به محمد هاشم المغربي من طرف الخديوي سعيد باشا.

وفي العصر العلوي يقول المختار السوسيس: «نبغ منهم ثلاثة من القضاة علا شأنهم، وخفق أزماناً بندهم منهمص: محمد بن أحمد القاضي التملي (والد الطيب)، تولى القضاء في تارودانت قبل 1255 هـ. وقد نفي مرة إلى وجدة هو والقائد حمو الروداني الأندوزالي في عهد مولاي عبد الرحمان، وفي عهد أيام القائد (بومهدي). وكانت وفاة القاضي التملي قبل 1276 هـ، بعد أن رجع إلى خطة القضاء بتارودانت من منفاه بوجدة.

والقاضي الثاني هو الطيب بن محمد ابنه (صاحب المطبعة)، كان يدرس ويدوم على ذلك وهو أدمت أهله خلقاً وألينهم عريكة، أخذ من فاس، ثم كان ينوب عن أبيه، ثم تولى القضاء بعده. توفي سنة 1282 هـ، وهو الذي اشترى في حجته المطبعة الحجرية الفاسية الأولى من مصر يريدتها لنفسه، ثم حازها منه السلطان سيدي محمد بن عبد الرحمان سنة 1281 هـ، فنقلت من الصويرة إلى مكناسة. فأول ما طبع فيها "الشمائل" للترمذي، ثم حولت إلى فاس. وهذه همة عظيمة نادرة له رحمه الله»⁽⁷⁵⁾.

كما عرض نص الاتفاقية التي أبرمت بين الطيب الروداني والطبيب المصري. وختم حديثه بقوله: «فكان لتارودانت وقاضيهما السابق في إحداث المطبعة في المغرب وذلك فضل الله يؤتيه من يشاء»⁽⁷⁶⁾.

يظهر من حديث المختار السوسي أن صاحب المطبعة ينحدر من أسرة عريكة، جل أفرادها من العلماء والقضاة بمنطقة سوس، فوالده وجدّه كانا من العلماء المحافظين المتشبهين بالتعاليم الإسلامية، عرفا بصرامتهما في قول الحق وتطبيقه، مما أدى إلى اصطدامهما أحياناً برجال المخزن وهذا ما يفسر النفي الذي تعرض له والد الطيب كما أشار إلى ذلك المختار السوسي. أما الطيب فكما قال عنه السوسي كان أدمت أهله خلقاً، وألينهم عريكة، أي أنه أقل تعصباً من أبيه وجدّه، وأكثرهم تبحراً في العلم. فقد كان عالماً ومدرساً لمختلف التخصصات الدينية، ذا أخلاق حميدة، ساهم بالعديد من أعمال الخير والإحسان ببلده «حيث بذل المجهود في حفر عين من الماء العذب، وأجره بتارودانت، وعم النفع بها في المساجد والحمامات والمطاهر والسقايات»⁽⁷⁷⁾.

(75) - المختار السوسي، خلال جزولة، مطبعة المهدية، تطوان (د.ت)، ج 4، صص. 120 . 121 .

(76) - نفسه، ص. 203.

(77) - البشير بن محمد بوحدو، بكرة الاقتضاض، في بغية الانقضاض، مخ. ضمن مجموع بالخزانة العامة بالرباط، تحت رقم 97 ج.

ولا ندري لم هذا التقصير من طرف المؤرخين المعاصرين، إلا إذا كانت الأحداث الخطيرة التي كان يعيشها المغرب حينئذ طغت على اهتماماتهم، أو ربما كان تأثير المطبعة محدوداً حينها. لذا لم تستأثر باهتمام المؤرخين. وعلى كل فإن الوثائق المتعلقة بالمرحلة الأولى من دخول الطباعة إلى المغرب، لم تظهر بصورة واضحة إلا في العقد السابع من القرن العشرين، حيث ظهرت بعض الدراسات في الموضوع معتمدة على وثائق ومستندات مخزنية استمدت من كناشة بليميني⁽⁷⁰⁾.

فعلى هذه الكناشة اعتمد عبد الوهاب بنمنصور في نشر وثائقه حول تاريخ المطبعة⁽⁷¹⁾، ومنها استمد كل من الأستاذ المنوني⁽⁷²⁾، وجرمان عياش⁽⁷³⁾ معلوماتهما حول الطباعة.

فهذه الوثائق أزال الغموض عن المسألة، وأثبتت بأن دخول المطبعة إلى المغرب، لم يكن عملاً مخزنيًا، وإنما تم بمبادرة فردية على يد القاضي محمد الطيب الروداني. فمن هو الروداني هذا صاحب المطبعة؟ وما سبب اختياره لمطبعة حجرية، وكيف انتقلت المطبعة من ملكيته إلى يد المخزن؟

لقد قصرت كتب التراجم في الحديث عن هذه الشخصية المهمة التي كان لها فضل إدخال المطبعة إلى المغرب، فعدا بعض الإشارات الصغيرة هنا وهناك، يكاد المختار السوسي أن يكون المرجع الوحيد لهذه الشخصية.

يرجع أصل الطيب الروداني إلى مدينة تارودانت، واسمه الكامل هوص: محمد الطيب بن محمد بن أحمد السوسي التملي الروداني، فالتلميون أصلهم من تيزخت قرية من قبائل تسكن حوض أمكن الكبير بالأطلس الصغير شمالي مركز تافراوت، وقد نبغ من التمليين في العصر السعدي عدد وافر من الأدباء والفقهاء والقواد تولوا مناصب سامية في دولة الشرفاء سواء في حاضرتهم الأولى المحمدية (تارودانت)، أو في بلاط مراكش، أو فاس دار مقام ولي العهد، وكذا في مختلف الأقاليم التي امتد إليها نفوذ السعديين حتى ما وراء الصحراء من بلاد السودان⁽⁷⁴⁾.

(70) - كناشة الطيب بليميني بوعشرين، مرجع سابق.

(71) - عبد الوهاب بنمنصور، مجلة الوثائق، المجموعة الثانية، 1396 هـ / 1976 م، من ص. 417 إلى ص. 437. وهي تتعلق بإنشاء صناعة الطباعة، وتعليم الطبايعين، وشراء المطابع ولوازمها.

(72) - المنوني، مظاهر بقطة المغرب الحديث، مرجع سابق، صص. 205 - 232.

(73) - جرمان عياش، دراسات في تاريخ المغرب، مرجع سابق، صص. 121 - 143.

(74) - محمد حجي، معلمة المغرب، العدد الثامن، سنة 1416 هـ / 1985 م، صص. 2556 - 2557.

الخماسي⁽⁶³⁾ إلى هذا الحدث المهم في عهد محمد الرابع. نفس الشيء نجد المؤرخ الناصري، سكت أيضاً عن ذكر المطبعة في مؤلفه "الاستقصا"، وهو الذي لم يحجم عن ذكر لا كبيرة أو صغيرة في مؤلفه هذا. ولا ندري هل سكوتها يدل على تحفظهما إزاء تكنولوجيا الطباعة، أم أن اهتمامهما كان منكباً على تدوين المواضيع البارزة، والتي لها تأثير على مسار التاريخ المغربي، مما يظهر معه أن دخول الطباعة إلى المغرب، لم يكن ذلك الحدث الكبير الذي أثار كل اهتمام المؤرخين.

ونجد إشارات للمطبعة في كتابات بعض المؤرخين المعاصرين لدخولها للمغرب، كمحمد العربي المشرفي (ت 1313 هـ / 1895 م)، الذي قال عنها ص: «وفي آخر السبعين اخترع إمامنا المؤيد بالله مطبعة، وأول ما طبع بها شمائل جده (المنسوبة للترمذي)⁽⁶⁴⁾. أما عبد السلام اللجائي (ت 1332 هـ / 1913 م) فأشار إلى الموضوع بقوله ص: «وجاء الأمير سيدي محمد هذه المطبعة التي تطبع فيها الكتب، هدية أهداها له خديمه الفقيه العلامة سيدي الطيب الروداني، قاضي تارودانت من سوس، لما حج ورجع من الحج، وجاء أيضاً بشيخ كبير عارف بكيفية الطبع فقبل سيدنا منه الهدية وأكرم الشيخ المذكور، وأجرى عليه النفقة»⁽⁶⁵⁾.

وإذا كانت الروايات المعاصرة لم تُولِ الأمر كثير اهتمام، حيث لم تقدم لنا وثائق أو معلومات دقيقة عن دخول المطبعة إلى المغرب، فإن المؤرخ ابن زيدان اكتفى في كتابه "النهضة العلمية"⁽⁶⁶⁾ بنقل رواية اللجائي، وفي كتابيه "الإتحاف"⁽⁶⁷⁾ و"الدرر الفاخرة"⁽⁶⁸⁾، ذكر فقط أسماء الكتب الأولى التي طبعت بفاس⁽⁶⁹⁾.

ويظهر أن ابن زيدان الذي أرخ للدولة العلوية ولمدينة مكناس بالضبط، وعاش داخل القصر الملكي، مطلعاً على الوثائق المخزنية، لم يكن على علم باشتغال المطبعة بمكناس قبل انتقالها إلى فاس، كما أنه لم يستشهد في كتاباته بأية وثيقة تتعلق بجلب المطبعة، رغم أن مؤلفاته بالذات كانت دائماً مشحونة بالوثائق.

(63) - شاء لهذا الكتاب أن يطبع بالمطبعة الحجرية بفاس.

(64) - محمد العربي المشرفي، نزهة الأبصار لذوي المعرفة والاستبصار، مخ. خ. ح تحت رقم 5616، ورقة 334.

(65) - عبد السلام اللجائي، المفاخر العلية والدرر السننية في الدولة الحسنية العلوية، مخ. خ. ح تحت رقم 460، ورقة 233.

(66) - عبد الرحمان بن زيدان، النهضة العلمية على عهد الدولة العلوية، مخ. خ. ح تحت رقم 11772، هامش الورقة 73.

(67) - ابن زيدان، الإتحاف، ج ٣، ط. 1931، ص. 568.

(68) - ابن زيدان، الدرر الفاخرة، المطبعة الاقتصادية بالرباط، 1356 هـ / 1937 م، ص. 92.

(69) - نفسه.

الطباعة الحجرية كأول اتصال مباشر للمغاربة بتكنولوجيا الطباعة

إن الكتابات الأجنبية التي أشارت إلى موضوع الطباعة متضاربة، وتعتمد جلها على الروايات الشفوية. فكل من دلفان (Delphin)⁽⁵⁷⁾، وبيريتي pérétié⁽⁵⁸⁾، يذكran بأن المطبعة الحجرية وصلت من القاهرة على يد شخص تركي، فاشتراها السلطان محمد الرابع وأنقذ هذا التركي. أما روجي لوطورنو Roger Le Tourneau، فقد اعتمد في مصادره على ما جاء عند دلفان وبيريتي وأوبين⁽⁵⁹⁾، بالإضافة إلى المعلومات الشفوية التي زوده بها أحد الطباع الفاسيين وهو محمد بردلة. ومما جاء في كتابته عن المطبعة ص: «ولم تنشأ مطبعة حجرية بفاس إلا في عهد سيدي محمد بن عبد الرحمان أي في النصف الثاني من القرن التاسع عشر، ويبدو أن الفكرة جاءت من حجاج أشادوا للسلطان بالوسيلة التي شاهدوا استعمالها في الشرق، كما يبدو أن المخزن منح تسهيلات لأول طابع، وهو الحاج الطالب الأزرق الذي أقام مطبعته بدرب الحمام قرب سوق الجوطية، بعدما استحضر من الشرق جهازاً وتقنيّاً تركياً»⁽⁶⁰⁾.

أما مياج Miège فقد أتى بنفس رواية لوطورنو، محيلاً عليه، زاعماً بوجود تقني تركي وبأن الأزرق هو أول من أدخل مطبعة إلى المغرب⁽⁶¹⁾.

وكما نرى فإن هذه الكتابات كلها اعتمدت على الرواية الشفوية، لذا لم تشكل معطيات أكيدة، حيث وقعت في العديد من الأخطاء، فالطبع مصري وليس تركياً كما تؤكد ذلك الوثائق المخزنية⁽⁶²⁾ ما أن اسمه الطيب الأزرق وليس الطالب الأزرق، كما ذكر لوطورنو ومياج، وليس هو صاحب أول مطبعة دخلت إلى المغرب، ولا أول طابع، بل سوف لا يظهر اسمه في ميدان الطباعة إلا في المرحلة الثانية من تاريخ الطباعة الحجرية.

أما المغاربة المعاصرون لدخول الطباعة إلى المغرب، والذين أرخوا لعصر السلطان سيدي محمد بن عبد الرحمان، فبعضهم سكت عن ذكر موضوع المطبعة، كمحمد أكنسوس (ت 1294 هـ / 1877 م) الذي كان كاتباً ومؤرخاً للسلطان. ومع هذا فلم يشر في كتابه "الجيش العرمرم

(57) - Delphin (G), Fès, son université et l'enseignement supérieur Musulman, Paris, 1889, p. 86.

(58) - Peretie (A), Les Madrasas de Fès, Archives Marocaine, N° XVIII, 1912, p. 363.

(59) - Aubin (E), Le Maroc d'aujourd'hui, Paris, A. Colin, 1904, p. 280.

(60) - روجي لوطورنو Roger le Tourneau، فاس قبل الحماية، تعريب محمد حجي ومحمد الأخضر، منشورات الجمعية المغربية للتأليف والترجمة والنشر، مطبعة دار الغرب الإسلامي، بيروت، لبنان، ج 2، ص. 681.

(61) - Miège(J.L), Le Maroc et l'Europe, op.cit, T.III,P.118

(62) - كناشة بليمي، مخ. ح. تحت رقم 10933.

ورد كذلك ذكر المطبعة في الرحلة الحجازية للغيثاني سنة 1858⁽⁵⁴⁾ حيث أعطى وصفاً دقيقاً لمطبعة بولاق بمصر، ذاكراً لأئحة الكتب التي اقتناها من هناك. وهذا ما يثبت أن الكتب المطبوعة بالمشرق كانت رائجة بين العلماء المغاربة. وإذا كانت كل من ملاحظة الصفار والغيثاني، اكتفت بوصف المطبعة والإعجاب بها، فإن السفير إدريس العمراوي، تجاوز ذلك إلى الاقتراح والطلب من السلطان اقتناء آلة الطباعة.

سجل العمراوي في رحلته التي عنونها بـ"تحفة الملك العزيز بمملكة باريز"⁽⁵⁵⁾، جميع مشاهداته بالعاصمة باريس، فوصف بدقة دار الطباعة التي خصص لها ما يزيد عن أربع صفحات في كتابه. وصف آلاتها وحروفها وطريقة سبكها.

لم يكتف العمراوي بوصف المطبعة وذكر محاسنها فقط - كالصفار والغيثاني - بل تجاوز ذلك إلى الطلب من السلطان بتزويد المغرب بآلة الطباعة، مظهراً مزاياها ومنافعها.

بقوله: «ونطلب الله بوجود مولانا أمير المؤمنين أن يكمل محاسن مغربنا بمثل هذه المطبعة ويجعل في ميزان حسناته هذه المنفعة....»⁽⁵⁶⁾.

فأهمية هذه الإشارات السابقة والتي ظهرت من خلال الرحلات سواء الحجازية أو السفارية إلى أوروبا، تؤكد أن المغاربة، خصوصاً السلاطين والعلماء كانوا على علم بفن الطباعة وتقنياتها، وعلى وعي بأهميتها في نشر العلم، وربما فكرة استيراد المطبعة كانت واردة لدى السلطان محمد الرابع لكنها أجلت بسبب انشغاله بالأحداث التي كان يعيشها المغرب، إثر هزيمة تطوان، وما خلفته من استنزاف لخزينة الدولة. ومما يؤكد اهتمام المخزن بفن الطباعة، أنه بمجرد ما وصلت آلة الطباعة إلى ميناء الصويرة، أمر بنقلها إلى مكناس، ثم إلى فاس، وأصبح يتحمل مسؤولية الإشراف على تسيير شؤونها. فكيف وصلت تكنولوجية الطباعة إلى المغرب؟

أجمعت جل الروايات على أن دخول الطباعة إلى المغرب لم يتم على يد الدولة، وإنما حدث بمبادرة فردية، وذلك سنة 1281 هـ/ 1864 م، وأنها كانت من النوع الحجري. في حين نجد بعض الاختلاف حول كيفية دخولها، وطريقة وصولها إلى يد المخزن.

(54) - محمد الغيثاني، الرحلة الحجازية، مخ. خ. ع. رقم ج 98.

(55) - إدريس العمراوي، تحفة الملك العزيز.... نشر زكي مبارك هذه الرحلة مع ترجمة النص إلى الفرنسية والمقدمة والتعليق، مؤسسة التغليف والطباعة للشمال، طنجة، 1989.

(56) - نفسه، ص. 55.

تطلع المغاربة لفن الكتابة الجديد

وردت العديد من الإشارات، التي تظهر بأن المغاربة كانوا منذ وقت مبكر على اتصال بتقنية الطباعة، ولبعضهم تطلعات ورغبات نحو اقتناء هذا الفن الجديد للكتابة. أول هذه الإشارات وردت أواخر القرن السابع عشر، من طرف السفير عبد الله ابن عائشة، عندما أرسله السلطان مولاي إسماعيل في سفارة إلى لويس الرابع عشر فزار المطبعة العربية هناك، والتي كانت في أوج نشاطها، فأهداه الفرنسيون بطاقة زيارة مطبوع عليها اسمه بحروف عربية جميلة، حملها معه إلى المغرب، وهو يتعجب لهذا الابتكار الأوربي الهائل. ومن المؤكد أنه أطلع مولاي إسماعيل على تلك البطاقة، وحدثه عن المطبعة العربية بباريس، وعدد له مزاياها وأهميتها في نشر الكتب. لكن هذه الفكرة ظلت مجرد إعجاب بدون أي أثر أو أية نتيجة.

إشارة أخرى عن اتصال المغاربة بالطباعة، وردت نهاية القرن الثامن عشر، حيث نشرت صحيفة فرنسية بالقاهرة سنة 1799 م اسمها *Courrier de l'Egypte*، خبر زيارة الشيخ محمد الفاسي لمطبعة الحملة الفرنسية بالقاهرة، ضمن مجموعة العلماء الذين وجهت لهم حكومة نابليون الدعوة لمشاهدة آلة الطباعة ذات الحروف العربية والفرنسية التي أنشأها الفرنسيون حديثاً بالقاهرة. وذكرت الصحيفة أن الشيخ الفاسي أبدى دهشته وإعجابه الشديد بها، معلقاً بأنه شاهد مطبعة الاستانة من قبل، لكنه لم يجد فيها عمالاً في مهارة ودقة عمال مطبعة القاهرة⁽⁵³⁾.

وردت إشارات أخرى إلى آلة الطباعة في كتب الرحلات سواء الحجازية، أو التي ألفها سفراء أو مرافقوهم إلى أوروبا. أولها "رحلة الصفار إلى فرنسا 1845 - 1846".

قدم الصفار في رحلته وصفاً دقيقاً لمختلف مظاهر الحضارة الفرنسية، مبدئياً إعجابه وانبهاره بالاختراعات التكنولوجية، من بينها المطبعة الملكية التي حظيت باهتمامه الشديد، فقدم لها وصفاً دقيقاً بكل الجزئيات والتفاصيل، فوصف دار الطباعة، وآلاتها الحجرية والتبيوغرافية، ومراحل الطبع، مبدئياً شدة إعجابه بها، معتبراً إياها من أعاجيب الصنائع.

(53) - ورد هذا الخبر عند:

. المنوني، مظاهر يقظة المغرب الحديث، ص. 205.

. إبراهيم عبده، تاريخ الطباعة...، ص. 41.

. فوزي عبد الرزاق، تاريخ الطباعة بالمغرب، ص. 129.

عزّز المؤلف بحثه بعرض قائمة الكتب المطبوعة بفاس بعناية شموئيل نديفوت (Shemuel Nedivot)⁽⁴⁹⁾ وابنه إسحاق، اللذين تمكنا من إصدار عشرة كتب في الفترة المتراوحة ما بين 1515 و1520 م⁽⁵⁰⁾، حدد عنوان تسعة منها فقط، ورتبها ترتيباً كرونولوجياً، ذاكراً تاريخ نشرها، ومكان تواجدها، وعدد نسخها.

ويظهر من اللائحة أن جلّ محتويات هذه المطبوعات كان يتعلق بكتب دينية، خمسة منها تتعلق بتلمود بابلي⁽⁵¹⁾، وأحدها شرح للطقوس الدينية والتقويم العبريين من إنجاز داوود بن يوسف أبودرهام David Joseph Abuderham، دُيّل بعبارة: (تمت طباعته في شهر Kislev من عام 5277 (نوفمبر 1516) في مدينة فاس على يد شموئيل نديفوت وابنه إسحاق).

إن يوسف تدغي باعتماده على معاينة النسخ الموجودة في القدس، وفي نيويورك، وعلى المعلومات الواردة في العديد من المصادر، جعل من مسألة وجود مطبعة عبرية بفاس خلال القرن السادس عشر حقيقة واقعة، معتبراً فاس مهد الطباعة العبرية في إفريقيا. في حين لا نجد أية إشارة إلى مطبعة القرن السادس عشر في المصادر المغربية، ولا حتى في الدراسات الحديثة التي اهتمت بتاريخ الطباعة بالمغرب، فكلها جعلت من سنة 1864 م، البداية الفعلية للطباعة بالمغرب.

ربما لم يتعامل المغاربة المسلمون مع المطابع العبرية، لموقفهم التقليدي من الأقليات الغير المسلمة. فالمغربي المسلم وفي أحيان كثيرة كان يعتبر اليهودي المغربي أجنبياً⁽⁵²⁾ يحترز في التعامل معه، خصوصاً إذا كان الأمر يتعلق بالثقافة الدينية بسبب اعتبارها ثقافة دخيلة، وبما أن جل منتجات المطابع كما رأينا ذو طابع ديني يهودي، لذا لم يظهر لها أي أثر في المصادر المغربية باعتبارها تخص أقلية غير مسلمة.

(49) - ذكر تدغي أن شامويل نديفوت، يهودي من أصل برتغالي، كان قد تمرن على الطباعة بلشبونة على يد ألبازار طوليدانو، وربما هذا كان سبب الخلط عند بعضهم، الذين ذكروا أن طوليدانو هو صاحب أول مطبعة بفاس.

(50) - أثناء مناقشتي مع المؤلف حول هذه اللائحة، أكد لي معاينته لبعض النسخ، من ضمنها المؤلفات التلمودية المحفوظة في مكتبة Jewish Theological Seminary بنيويورك، وكذا نسخة تامة من تفسير أبو درهم بالخزانة الجامعية بالقدس، بالإضافة إلى نسخ لدى الخواص.

(51) - يوجد تلمودان، تلمود بابلي، وتلمود بروشليمي. والتلمود هو كتاب الإرث الشفوي اليهودي الذي خلفته الأجيال، وكان من الأصل تفاسير على التوراة تناقلتها الأجيال شفويّاً، وهو بالعبرية والآرامية، ويتكون من قسمين هما المشنه والجماري. وموضوعه: أ. التشايع والقوانين اليهودية. ب. الأخبار والأساطير التي تجمعت على مدى القرون فأصبحت جزءاً من تاريخ بني إسرائيل. انظر: أحمد شحلان، ندوة الإصلاح والمجتمع المغربي، مرجع سابق، ص 208.

(52) - Simon Levy, Essais d'histoire et de civilisation Judeo-Marocaines, centre Tarik Ben Ziad, 2001, p. 3.

بعضهم فكرة تزويد البلاد بها. وتذهب بعض الدراسات إلى أبعد من ذلك، لتجعل المغرب أول دولة عربية عرفت تقنية الطباعة، وتؤكد أن بدايتها تعود إلى القرن السادس عشر على يد المهاجرين اليهود.

فاس مهد الطباعة العبرية بإفريقيا

تتضارب الآراء حول وجود الطباعة في المغرب في وقت مبكر من القرن السادس عشر، حيث أثارت مقولة إدخال اليهود المهاجرين المطبعة، جدلاً ما بين مؤيد ومعارض لها. فقد ورد في الموسوعة اليهودية، خبر جلب المطبعة إلى المغرب على يد اليهود الفارين من إسبانيا والبرتغال، بعد سقوط الأندلس سنة 1492 م، حيث استقر العديد منهم بمدينة فاس، وهناك استمروا في إنتاج مطبوعاتهم⁽⁴⁵⁾.

وفي هذا السياق، يقدم لنا يوسف تدغي⁽⁴⁶⁾ (Joseph Tedghi) دراسة مفصلة في مؤلفه "الكتاب والمطبعة العبرية بفاس"⁽⁴⁷⁾، وهو أهم مؤلف تناول الموضوع بتفصيل، حيث خصص الشق الثاني منه لتاريخ المطبعة العبرية بفاس في القرن السادس عشر، وربط بدايتها بتهجير اليهود من إسبانيا سنة 1492 م، والبرتغال سنة 1497 م، حيث أغلقت المطابع، وانتقل أصحابها إلى أماكن أخرى من حوض البحر المتوسط فراراً بدينهم، فاستوطن العديد منهم بالمغرب، واستقر معظمهم بفاس حاملين معهم مطابعهم. وهناك استأنفوا طبع كتبهم مستفيدين من جو التسامح الديني السائد داخل المغرب، حيث كان المرينيون قد سمحوا لليهود المطرودين من الأندلس بالاستقرار بالمغرب، مما وفر لهم الجو المناسب للعمل الثقافي والعلمي برعاية الدولة. ويضيف تدغي بأن اليهودي "أليعازر طوليدانو Eliezer Toledano" وهو إسباني الأصل، ومن أشهر الطابعين اليهود، عاش في طليطلة، ثم استقر بالبرتغال، وبها سير مطبعة ما بين 1489 - 1492 م، وكان ضمن المهاجرين إلى فاس، حاملاً معه جزءاً من تجهيز مطبعة لشبونة⁽⁴⁸⁾.

(45) - Encyclopedia Judaica, Jérusalem, 1975, T. 11, colonne 303.

(46) - نسبة إلى تدغة، وهو من مواليد مدينة فاس، يقيم حالياً بباريس، أستاذ اللغة العبرية بالمعهد الوطني للغات والحضارات الشرقية INALCO.

(47) - الكتاب في الأصل باللغة العبرية، نشر سنة 1994 بالقدس، من طرف معهد ابن صفى للبحث في موضوع الطوائف اليهودية الشرقية والجامعة العبرية، يقع في 208 ص. وقد تم تعريبه بصور ورسوم من بينها صور للمطبوعات العبرية بالمغرب خلال القرن السادس عشر، وكذلك مطبوعات القرن العشرين.

في شهر أبريل من سنة 1998، التقيت بالمؤلف بباريس، حيث سلمني جزءاً من الكتاب، يحتوي على الجزء المتعلق بالمطبعة العبرية بفاس، مترجماً إلى الفرنسية تحت عنوان:

«1492, L'expulsion des Juifs d'Espagne».

(48) - لم يتأكد المؤلف من حقيقة وصول أليعازر طوليدانو إلى المغرب، نفس المرجع، ص. 255.

المغاربة و فن الكتابة الجديد: الطباعة الحجرية (المليثوغرافيا) نموذجا

لطيفة الكندوز

أجمعت كل الدراسات والأبحاث التي أشارت إلى تاريخ الطباعة بالمغرب، على أن المغاربة لم يستعملوا هذا الفن الجديد للكتابة إلا في العقد السادس من القرن التاسع عشر، وذلك بعد مرور أربعة قرون على اكتشافه بأوروبا؛ وأن المغرب كان آخر دول الشمال الإفريقي، ومن أواخر الدول الإسلامية إقبالاً على التقنية الجديدة للكتابة، رغم كونه من أقربها إلى أوروبا. وهذا ما يدفعنا إلى التساؤل عن الأسباب التي كانت وراء ذلك التأخير. أهي أسباب سياسية ترجع إلى سياسة العزلة والاحتراز التي لجأ إليها المغرب لحماية أرضه من الأطماع الأجنبية، خصوصاً بعد احتلال الجزائر سنة 1830م؟ أم هي أسباب اقتصادية، ناتجة عن الوضعية العامة للاقتصاد المغربي، وما عرفه من هزات خلال القرن التاسع عشر، خصوصاً بعد هزيمة إيسلي سنة 1844، وحرب تطوان سنة 1859 - 1860م؟ أم هي أسباب اجتماعية، ترتبط بالتوجه العام السائد داخل المجتمع المغربي، المبني على تمسك المغاربة الشديد بالمحافظة على التقاليد الإسلامية، ورفضهم لأي تغيير وتجديد أو تفتح على أوروبا وحضارتها؟ أم أن هذه الأسباب جميعها كانت وراء تأخير دخول الطباعة إلى المغرب؟

فاحتراز المغاربة من كل ما هو أوروبي ومواجهتهم للضغوط الأجنبية وتعاملهم بحذر مع البنيات الجديدة التي حاولت أوروبا زرعها في بلادهم خوفاً أن تكون سبباً لتسرب الأوربيين للمغرب دفع بهم إلى سلك سياسة العزلة. وكذلك تشبث المغاربة بتقاليدهم في الكتابة خصوصاً فن الخط المغربي الذي كان يعتبر مقدساً لديهم، حيث كانوا في غنى عن منتجات الطباعة. ومن أسباب رفض المغاربة لفن الطباعة في رأيهم نجاسة المواد التي صُنعت منها آلات الطباعة، فلا يليق بلغة القرآن وكلام الله ولغة الجنة أن تُكتب بهذه المواد.

الاتصال الأولي للمغاربة بفن الطباعة

إذا كانت جل الدراسات قد حددت سنة 1864م، كتاريخ لدخول الطباعة إلى المغرب، فهذا لا يعني أن المغاربة لم يكن لهم علم حتى هذا التاريخ بتقنية الطباعة، ولم يكونوا على اتصال بها، بل هناك العديد من الإشارات التي تؤكد أن المغاربة كانوا على علم بفن الطباعة، وتولدت لدى

TYPOGRAPHIE, HISTORIQUE ET PRATIQUE.

Christian Paput

*Maître-Graveur et Conservateur
au Cabinet des Poinçons de l'Imprimerie nationale*

Les besoins de manuscrits, sans cesse plus importants depuis le XIIe siècle, ont engendré au XVe siècle l'invention de la typographie selon Gutenberg, puis son développement jusqu'à nos jours.

Il est de coutume de retenir Gutenberg comme étant l'inventeur du caractère mobile et métallique d'impression.

Bref rappel sur la typographie au plomb : le poinçon est une tige d'acier dont une extrémité polie est gravée en relief et à l'envers. Ce poinçon est enfoncé en force et à froid dans un bloc de cuivre qui devient la matrice ou moule. Le plomb est ensuite fondu dans la matrice et constitue le caractère d'imprimerie qui sera assemblé par le compositeur typographe puis imprimé.

Tout d'abord rudimentaire à ses débuts, la typographie d'aujourd'hui est l'héritière de 500 ans de traditions.

Dès les débuts du XVIe siècle, François 1er comprend l'importance qu'il y a à posséder un outil efficace permettant d'imprimer donc de communiquer et diffuser ses idées, sa pensée, moyen d'affirmer et de renforcer son pouvoir.

C'est en 1540 que commence l'histoire des collections de l'Imprimerie nationale.

Puis c'est sous Richelieu, en 1640, qu'est créé l'établissement Royal qui s'équipe, par achat, d'une typographie spécifique : le Garamont. C'est ce caractère qui inaugure la série des caractères exclusifs de l'État français.

La première véritable typographie créée pour l'imprimerie Royale sera gravée à partir de 1692 et mise en œuvre en 1702. Il s'agit du caractère Grandjean dit Romain du roi. Ce caractère inspiré des dessins de la commission Jaugeon, réunion de scientifiques et de savants est le caractère qui symbolise le règne de Louis XIV.

Puis au fil des siècles, le patrimoine typographique sera constitué en fonction des besoins, des modes, des réformes et des désirs des dirigeants français. Il faut noter les grands caractères français : le Luce, caractère de l'époque de Louis XV, le Didot est celui de Napoléon, le Marcellin Legrand est le caractère de Charles X, le

Jaugeon caractère du début du XXe siècle et le Gauthier de la fin du XXe, notamment. À noter que le caractère Gauthier est le seul caractère à avoir été adapté à toutes les technologies de l'imprimerie à savoir la fonte traditionnelle, la fonte mécanisée (monotype) et le numérique.

À côté de ces grands caractères latins, l'Imprimerie nationale possède toute une collection de caractères dits orientaux : japonais, grecs, arabes, chinois, géorgiens, etc., plus de 60 écritures y sont présentes.

C'est le travail incessant des graveurs attachés à l'Imprimerie nationale et plus particulièrement au Cabinet des poinçons qui a permis de constituer ce patrimoine unique classé Monument historique ;Le Cabinet des poinçons est le lieu de conservation de l'Imprimerie Nationale pour ses pièces gravées ainsi que l'atelier des graveurs qui y travaillent. Les différents travaux menés au Cabinet des poinçons.

D'une part, pour les besoins de la composition plomb des volumes d'une collection (la Salamandre) éditée par l'IN qui sont toujours composés manuellement.

Dans le même domaine, un atelier du livre d'art et de l'estampe est là pour répondre à des besoins extérieurs en matière de lithographie, taille-douce ou encore phototypie. Cet atelier accompagne fréquemment ses travaux graphiques d'une impression typographique.

C'est pourquoi, il est de la mission du Cabinet des poinçons de conserver les poinçons typographiques en état parfait de manière à répondre aux besoins de fourniture de caractères.

Dans la pratique, l'atelier de composition commande les caractères plomb à l'atelier de fonderie. Celui-ci fond selon les besoins dans les matrices de sa typothèque (lieu de conservation des matrices) lorsque l'état de celles-ci le permet. Sinon la demande remonte la filière de fabrication jusqu'au Cabinet des Poinçons qui fournit le poinçon original permettant de refrapper une matrice. Si le poinçon ne peut supporter la frappe, il est alors regravé à l'identique en utilisant les méthodes ancestrales que nos anciens nous ont léguées. Une partie création est également présente au Cabinet des poinçons. Il s'agit, en dehors de polices spécifiques en cours de gravure, d'éléments ponctuellement présents en typographie décorative.

Nous disposons pour mener à bien ces diverses tâches d'une vaste bibliothèque, historique et technique.

Mais une question se pose. Pourquoi travailler à l'ancienne, avec les techniques d'autrefois, alors que l'on peut penser que les technologies actuelles sont plus rapides et plus précises ? Il s'agit pour nous de témoigner d'un niveau général de qualité et de savoir-faire dans tous les domaines du livre et de l'imprimé. Certains éléments sont spécifiques à l'imprimerie « au plomb » : le foulage, le papier,

l'engraissement des caractères à l'impression, etc. Pour qu'un certain niveau de qualité soit respecté, il est nécessaire que toutes ces particularités soient prises en compte par le créateur contemporain ainsi que par tous les partenaires de la chaîne graphique.

Pour conclure, je ne résiste pas au plaisir de vous montrer d'autres pièces relatives au livre et à l'imprimé que sont les fers-à-dorer, les taille-douces, les bois d'affiches, les bois d'illustration. Ce sont 500 000 pièces, toutes classées Monuments Historiques.

DE L'HISTOIRE, DE L'ÉCRITURE ET DE LA TYPOGRAPHIE LA NAISSANCE DE L'ÉCRITURE

Christian PAPUT

L'écriture est une invention récente face à l'histoire de l'humanité : l'homme parle depuis environ 100 000 ans et n'écrit que depuis 5 000 ans.

Que ce soit au moment de l'apprentissage de l'écriture chez les enfants, ou plus tard à l'époque de l'adolescence lorsque l'on essaye de personnaliser son écriture, certaines questions viennent naturellement à l'esprit.

Pourquoi un «A» s'écrit-il de cette manière ? Pourquoi telle lettre possède-t-elle cette forme et pas une autre ? Pourquoi avons-nous des majuscules et des minuscules ?

Ces questions trouvent petit-à-petit leurs réponses à qui cherche un peu, mais la grande question essentielle demeure la naissance de l'écriture et pour nous, peuples occidentaux, son évolution vers l'écriture alphabétique.

Pourquoi un jour l'homme s'est-il trouvé «équipé» de cet «outil-écriture», indispensable, rigoureux et beau ?

Nos «cultures», artistiques, technologiques ou sociales sont enregistrées grâce à lui.

Il est difficile, voire impossible, d'imaginer organiser un monde développé sans écriture, et pourtant nous savons que celle-ci n'a pas toujours existé.

Beaucoup de peuples ont cru qu'elle était d'essence divine, plus raisonnablement, elle est sans doute née de l'impossibilité de mémoriser un grand nombre d'informations. Il s'agit alors de trouver un système standardisé pour conserver des informations orales, puis que ce même système permette de restituer les informations par le langage.

Il faut pour bien appréhender l'arrivée de l'écriture, prendre en compte l'invention de l'élevage, de l'agriculture et de leurs besoins. Il est également intéressant d'examiner les deux thèses principales qui encadrent la venue de l'écriture dans l'humanité.

Selon la thèse de la «nouvelle archéologie» (développée dans les années 70 aux États-Unis par Lewis Binford), qui explique les grands événements de la préhistoire

par l'écologie et l'économie, les hommes se seraient mis à cultiver la terre et à pratiquer l'élevage parce que les ressources «sauvages» étaient épuisées. Pour Jacques Cauvin (chercheur à l'institut de préhistoire orientale et directeur de recherche au CNRS), cette explication ne tient pas, car certaines fouilles ont montré que l'agriculture n'est pas née dans un contexte de pénurie. Son apparition serait plutôt la conséquence d'une profonde transformation des mentalités.

Elle fut en effet précédée par un changement d'attitude de l'homme vis-à-vis de l'Univers.

À cette époque, ses préoccupations métaphysiques trouvent une nouvelle orientation, qui se traduit par une «révolution des symboles». Pour la première fois, on trouve des figures de femme et de taureau sur les parois des grottes, qui susciteront plus tard, et jusqu'à l'époque classique, un véritable culte dans tout l'Orient méditerranéen. La femme représente la déesse mère, et le taureau, fils de la déesse mère, traduit la force masculine, la guerre, l'orage, la sexualité.

Si l'homme se met brusquement à adorer des divinités, c'est qu'il a désormais conscience d'être dominé par des forces qui lui sont supérieures, impliquant sa propre «finitude», explique Jacques Cauvin. Pour compenser, il se met à travailler avec une frénésie nouvelle et cherche à «maîtriser le monde». Cette maîtrise commence par l'«invention» de l'agriculture puis de l'élevage, c'est-à-dire la «production de subsistance». C'est le moment où apparaît la pictographie symbolique (voir les quatre grandes étapes) qui correspond à une étape importante de l'invention de l'écriture.

D'après Émilie Masson (chercheur au CNRS, spécialiste de l'histoire de l'écriture et des civilisations du Proche-Orient ancien), l'écriture suit l'évolution de la société qui l'a créée.

Les premières ébauches d'un système graphique apparaissent lorsqu'une civilisation atteint le stade économique et intellectuel où le besoin se fait sentir de fixer, de manière plus ou moins durable, des faits concrets, inhérents à la vie quotidienne, ainsi que des données plus abstraites relevant des premiers concepts et des premières croyances. C'est pourquoi l'écriture, en tant que phénomène, ne résulte pas d'une création unique dans le temps et dans l'espace.

Le plus souvent, il s'agit d'inventions spontanées, pour la plupart indépendantes, qui se produisent dans différents points du globe, à des dates variées, et qui connaissent des aboutissements inégaux. Elles ont un point de départ sensiblement identique, en général des signes primitifs, puis elles évoluent selon des processus analogues.

L'apparition de l'écriture coïncide avec une organisation sociale certaine, dont la conséquence est la fondation des villes.

On peut distinguer quatre étapes cruciales dans cette évolution.

□ La première est celle des signes primitifs, marques isolées tracées spontanément, sous forme d'encoches, de nœuds, de représentations linéaires, parfois même d'esquisses annonçant les premiers pictogrammes. Ces signes sont révélateurs d'un peuple de chasseurs. On les trouve dans les grottes ornées du paléolithique, mais aussi, encore aujourd'hui, chez les Amérindiens d'Amérique du Sud et les Aborigènes d'Australie.

□ La seconde étape, celle de la pictographie symbolique, voit apparaître des schémas plus complexes, qui cherchent à traduire des messages. Par exemple, l'action de marcher est exprimée par deux pieds en mouvement de marche, l'action de donner est concrétisée par une main tendue, etc. Ils apparaissent (schémas) au moment où l'agriculture se met en place. «Inventée» au Proche-Orient vers 9000 av. J-C., l'agriculture a gagné ensuite la vallée du Danube vers 5500 av. J-C., puis l'Europe occidentale vers 5300 av. J-C. Son avènement fait ressentir le besoin de noter, à l'aide d'un répertoire bien établi, des règles administratives, ou de transmettre un savoir-faire.

Les plus anciens et les plus élaborés de ces graphismes symboliques sont ceux qui viennent d'être découverts et datés de 9000 ans av. J-C. (Les graphismes du bassin central et oriental du Danube sont datés de 4000 ans et ceux des parois du Mont Bégo (Vallée des merveilles) de 2000 ans av. J-C.).

□ La troisième étape vers l'écriture est celle de la «révolution urbaine» (organisation sociale plus hiérarchisée et économie en plein développement).

À Sumer (Mésopotamie) et dans les premières dynasties égyptiennes, apparaissent les premières écritures idéographiques : l'écriture cunéiforme des Sumériens et les hiéroglyphes des Égyptiens.

□ La quatrième et dernière phase voit la naissance de l'écriture alphabétique. Le plus ancien alphabet connu est celui des Phéniciens, daté de 1300 av. J-C. C'est l'alphabet ougaritique, mis au jour à Ougarit (aujourd'hui Ras Shamra), sur la côte syrienne.

Une autre analyse du processus concernant la naissance des écritures, nous permet de dégager les évolutions suivantes :

Les écritures synthétiques : notation à l'aide de cailloux, de cordelettes, de rebus, battements de tambours etc... le nombre des signes est limité.

Les écritures analytiques : présentent un progrès sensible par la décomposition des signes symboliques. Chaque signe servant éventuellement à un mot : chinois, égyptien, sumérien. Le principal défaut de ce type d'écriture est la mémoire immense indispensable à un message élaboré.

Les écritures phonétiques : c'est la notation par signes de sons constituant le mot. Ces écritures sont syllabiques, consonantiques ou alphabétiques, (peu d'écritures sont syllabiques, un bon exemple est l'écriture syrienne du IIe millénaire av., le persépolitain, le cyprote).

Les premières notations graphiques sont insuffisantes pour assurer un véritable système d'écriture permettant d'affirmer la transcription d'idées précises, mais seulement la notation d'idées simples. Ce sont les écritures pictographiques. Celles-ci incapables de noter la pensée abstraite vont évoluer vers des écritures idéographiques plus précises. C'est dans le pays de Sumer et presque en même temps en Égypte, sur le bord oriental de la méditerranée qu'est née l'écriture vers le IVe millénaire avant notre ère. À partir d'un répertoire de signes et de symboles, ces précurseurs ont élaboré un système graphique pour fixer les messages. La société d'alors, principalement fondée sur l'agriculture et l'élevage, a senti la nécessité d'enregistrer et d'administrer la production et les échanges commerciaux. L'outil indispensable à cette tâche sera l'écriture. Une nouvelle classe va se créer : les lettrés. Ils seront scribes, juristes, écrivains et leur nouveau genre d'activité intellectuelle va se développer, fondé sur l'observation, le raisonnement, l'analyse autant que sur la déduction. Le fait de savoir lire et écrire était considéré comme un grand privilège.

Les écritures analytiques du début n'offrent pas la possibilité de transcription de la pensée écrite, traduite en parole. Et ce sont des éléments phonétiques ajoutés à ces écritures qui vont mener à l'écriture phonographique, tout d'abord syllabique, puis la précision augmentant avec les besoins liés aux langues parlées, le découpage phonographique sera de plus en plus précis pour parvenir au système alphabétique.

Les cunéiformes

L'écriture naît à Sumer en Mésopotamie (meso = entre et potamos = fleuve) sur le territoire situé entre les deux fleuves que sont le Tigre et l'Euphrate. En 3300 avant Jésus-Christ les premiers signes sumériens apparaissent à Uruk, en Mésopotamie, aujourd'hui Warka en Iraq.

De cette situation géographique découle le support de cette écriture en tablettes d'argile humide et souple. On comprend également que l'outil choisi pour y graver des signes soit le roseau. Ces divers matériaux induisent partiellement les graphismes de ces écritures, qu'elles soient pictographiques ou cunéiformes. Précurseurs de ces écritures, les calculis sont des sortes de jetons ou de pièces d'argile enfermés dans des bulles de terre. L'enregistrement ou le comptage des propriétés est ainsi devenu possible et fiable. Pour noter que l'on possède trois moutons, on enferme trois jetons dans une bulle d'argile. Puis sur cette enveloppe extérieure on effectue trois repères symbolisant le contenu. La notation du nombre de moutons est faite et vérifiable par le bris de la bulle après séchage. Seul

problème, il faut encore savoir s'il s'agit de moutons, de maisons ou de tout autre objet. On sait donc dénombrer mais pour écrire il faudra encore quelques étapes complémentaires.

Tracés, griffés sur des tablettes d'argile cuite, ce sont au début, des éléments figuratifs qui permettent les notations concrètes : animaux, plantes, etc. C'est le plus ancien système d'écriture que nous aient laissé les documents. Ce principe descriptif n'est pas lié à une langue en particulier et ne permet pas les notations abstraites ou théoriques. La représentation des objets, avec ces pictogrammes va se préciser et petit à petit se synthétiser en signes cunéiformes c'est-à-dire en forme de clous. L'origine du terme cunéiforme est à trouver dans le mot latin *cuneus* : clou. L'écriture cunéiforme proprement dite apparaît. Elle comprend environ 300 à 900 signes, gravée avant cuisson, sur ces tablettes d'argile humide à l'aide de bâtonnets taillés.

Diverses langues utilisèrent des signes cunéiformes comparables, le vieux perse, l'élamite, l'akkadien. C'est sans doute parce que ces signes et ce système d'écriture étaient au service d'une quinzaine de langues et de dialectes dans des régions diverses et éloignées, qu'ils donneront naissance aux différentes écritures alphabétiques du monde.

C'est un jeune professeur du Lycée de Göttingen (Allemagne), Georg Friedrich Grotefend (1775-1813) qui le premier en 1802 a été en mesure de présenter les premières traductions d'écritures cunéiformes. Henri Geswicke Rawlinson en paracheva le déchiffrement en 1846.

Le sens de lecture de ces écritures s'effectue de droite à gauche, en colonne, puis de gauche à droite et en ligne. Instrument de prestige, cette écriture est réservée à une élite.

Seuls trente à quarante signes sont utilisés pour certaines adaptations alphabétiques de cette écriture.

Les Hiéroglyphes

Les hiéroglyphes Égyptiens apparaissent presque en même temps que les cunéiformes, aux alentours de 3200 avant notre ère. Il s'agit d'un système idéographique et phonographique primitivement élaboré pour des besoins cadastraux. Ces signes au nombre d'environ 700 dès les débuts se multiplièrent pour parvenir à 5000 entre le II^e avant et le IV^e après (période gréco-romaine).

C'est le Dieu Thot, substitut de Rê, qui en Égypte est le Dieu inventeur de l'écriture. Il est à la fois le «maître des paroles divines» et le «scribe des Dieux».

En Égypte, les «hiéroglyphes», «signes sacrés» ou «qui renferment la parole des dieux» peuvent exprimer un objet, une idée ou un son.

À l'origine, chaque signe a pu reproduire l'objet ou l'acte qu'il représentait, mais il est certain que les signes acquièrent de très bonne heure une valeur phonétique qui se superposa à la valeur idéographique sans la remplacer, ces deux éléments apparaissant combinés dès la plus ancienne date où cette écriture nous est connue.

L'apparition de l'écriture hiéroglyphique, qui est l'écriture égyptienne la plus ancienne et la plus caractéristique, a coïncidé avec l'unification de l'Égypte vers 3000 avant J. C. Son usage s'est maintenu jusqu'au III^e siècle après J. C.

Les hiéroglyphes furent gravés en creux ou en relief sur matière dure (pierre, bois ou métal) et employés comme ornements; ils constituent l'écriture égyptienne monumentale.

Ils se présentent sous la forme de dessins, de caractère artistique remarquable, disposés en ligne horizontale ou verticale.

À l'horizontale, ils peuvent être écrits dans les deux sens (boustrophédon) et comptent près de 7000 signes. C'est l'orientation des signes hiéroglyphiques qui fixe le sens de lecture des textes. La lecture s'effectue en allant à l'encontre des profils des signes animés (hommes ou animaux). Ils sont généralement réservés aux inscriptions officielles comme le pensaient les Grecs bien que cette écriture ait également servi aux usages profanes.

Néanmoins, deux écritures cursives, s'écrivant et se lisant de droite à gauche, ont été développées à partir de l'écriture hiéroglyphique :

à partir d'une époque qui a dû commencer environ en 2500 avant J.-C., le système hiéroglyphique a été doublé d'une cursive, utilisée surtout pour écrire sur papyrus, cuir, étoffe et plus rarement dans les inscriptions gravées. Les signes en ont été abrégés en des tracés où la figuration dessinée disparaît. Le nom de hiératique ou «sacerdotal», attribué par les Grecs à ce genre d'écriture au moment où on l'employait pour des usages solennels, lui est demeuré.

Cette écriture est rapide et cursive pour les besoins des religieux, de l'administration, du commerce, des textes littéraires, ou scientifiques. Son usage se maintiendra jusqu'au Ve siècle après J. C., elle se pratique sur le papyrus avec un pinceau et l'encre.

Au VI^e siècle avant J.-C., l'écriture hiératique devait se transformer en une cursive d'emploi courant réservée aux usages de la vie pratique : ce fut l'écriture démotique ou «populaire», qui notait un état évolué de la langue, tandis que se perpétuait la transcription de l'ancien égyptien classique en hiéroglyphes primitifs. Elle sert à noter les textes de la vie courante et se trace également sur papyrus à l'encre.

Elle a perdu son aspect figuratif et a gagné des ligatures et des abréviations.

L'écriture démotique se compose de signes syllabiques et donnera sept signes à l'alphabet copte, langue indigène de l'Égypte après la conquête arabe (641).

C'est à Rosette, ville d'Égypte, que l'officier du génie Bouchard fit, en 1799, la découverte de la fameuse pierre, appelée par la suite «pierre de Rosette». Il s'agit plus exactement du fragment d'une stèle gravée en hiéroglyphe, en démotique et grec d'un décret de Ptolémée V qui allait permettre de trouver la clef de l'écriture sacrée égyptienne.

En 1822, l'égyptologue français J. F. Champollion (1790-1832) parvint, le premier, à déchiffrer les caractères hiéroglyphiques et démotiques inscrits sur la pierre de Rosette fragment de stèle datant de 196 avant J. C.

(À 16 ans Champollion maîtrisait déjà 6 langues orientales, en plus du grec et du latin. professeur d'histoire au Lycée de Grenoble, à 19 ans, il est fasciné par le mystère des hiéroglyphes, dont les Français ont rapporté des copies manuscrites au retour de la campagne d'Égypte.

De hautes connaissances sur l'Égypte ancienne et le copte, dernier état de l'égyptien au IIIe siècle de notre ère, ont permis à François Champollion de réaliser, en 1822, le déchiffrement des hiéroglyphes, grâce à l'étude poussée d'un document découvert lors de l'expédition française de 1799 en Égypte : la pierre de Rosette, à l'inscription trilingue (hiéroglyphique, démotique, grecque).

Le Phénicien

Le mot vient phénicien vient d'un mot grec : «phoinikes» qui signifie pourpre ou rouge, comme le pigment qui était une des productions de la phénicie. Cette contrée s'étendant le long de la Méditerranée, depuis le mont Carmel au Sud jusque vers l'embouchure de l'Oronte au Nord était peuplée dès le début du IIIe millénaire de sémites cananéens.

Les origines de l'écriture Phénicienne sont obscures et résultent probablement des écritures du Moyen-Orient, égyptiennes, cunéiformes, égéennes, protosinaïtiques, etc... Les Phéniciens la répandent à travers la Méditerranée. Les premières inscriptions connues en alphabet à consonnes s'échelonnent entre les XIVe et XIe siècles av. J.C.

La conception claire de cet alphabet est due vraisemblablement à l'influence de l'Égypte d'une manière générale mais l'alphabet d'Ougarit précède le phénicien.

La découverte de l'alphabet s'est faite par la capacité à décomposer la parole en sons simples, représentés chacun par un seul signe. Cette notation, déjà connue des Égyptiens sera utilisée par les phéniciens.

L'articulation complète de cette langue à partir de l'écriture consonantique ne peut s'imaginer que grâce à un système vocalique déterminé par le contexte.

Les phéniciens écrivaient et lisaient de droite à gauche.

L'alphabet phénicien est à la source de tous les alphabets actuels. L'inscription la plus ancienne en phénicien est une inscription gravée sur le tombeau du roi Akhiram à Byblos (aujourd'hui Djebaïl au Liban) date du XIe-Xe siècle avant J-C.

Le phénicien semble avoir résisté plus longtemps que les autres langues sémitiques à l'expansion de l'araméen.

Le commerce, activité essentielle des cités développées de Phénicie telles, Byblos, Arados, Sidon et Tyr, pousse à la fondation de colonies dont Carthage est la plus remarquable et par lesquelles la langue se perpétuera quelque temps supplémentaires.

Le phénicien reste en usage en Phénicie jusqu'au IIe siècle de notre ère.

Composé de vingt-deux consonnes, aux formes nettes et précises, l'alphabet phénicien du début excluait toute confusion ; les déformations de caractères appurent seulement lors de l'expansion de cette écriture. Ce système simple d'utilisation comme d'apprentissage est en théorie très démocratique.

La légende prétend que c'est Cadmos, qui parti de Tyr (ville du Liban au Sud de Beyrouth) à la recherche de sa sœur Europe, enlevée par Zeus, aurait fondé Thèbes (ville grecque) et rapporté aux grecs, les lettres phéniciennes. L'alphabet phénicien fut diffusé à partir de Tyr, vers l'Ouest, la Grèce et Rome, mais aussi vers l'Asie centrale et l'Inde devenant, par là également, l'ancêtre de presque tous les alphabets qui notent les langues sémitiques et la plupart des langues indo-européennes.

L'alphabet est un système conventionnel abstrait : il n'y a pas de lien entre le sens du texte et sa représentation graphique. Ce qui en fait un outil universel permettant de noter n'importe quelle langue.

L'alphabet a été inventé selon le système acrophonique. En isolant chaque son (consonne) et en représentant chacun d'entre-eux par le dessin d'un objet dont le nom commence par ce son. Par exemple, une maison se dit Beth dans la plupart des langues sémitiques. Le signe B représente le plan approximatif d'une maison. Le nom de ce signe sera donc Beth et se prononcera B. La lettre B est née.

À Ougarit, cité antique de la côte syrienne, on a retrouvé des milliers de tablettes en écriture cunéiforme datant du XIIIe siècle avant J-C. et déjà, on constate qu'une partie d'entre-elles sont en écriture alphabétique de 30 signes.

L'alphabet phénicien a donné naissance d'une part à l'alphabet grec, lui-même à l'origine de l'alphabet cyrillique (europe orientale). C'est également du Phénicien que vient l'alphabet araméen, lui-même à la source de l'hébreu. L'arabe et les écritures de l'Inde nous viennent également du Phénicien.

Le phénicien classique des IVe et Ve siècles avant J-C. se trouve représenté par l'inscription du sarcophage d'Ech mounazar, roi de Sidon.

Le Grec

Les Grecs ou Hellènes du premier millénaire avant J-C. essayèrent plusieurs écritures, crétoise et mycénienne, avant d'adopter l'alphabet phénicien.

Les grecs avaient besoin d'un alphabet adapté à la transcription de leur langue, c'est pourquoi, en inventant les voyelles qu'ils fondirent dans l'alphabet phénicien, les grecs, vers 800 avant J-C., vont promouvoir l'alphabet moderne de 24 lettres avec consonnes et voyelles. Ils ont emprunté l'alphabet phénicien en attribuant à certains signes inutiles pour eux, une valeur phonétique dont ils avaient l'utilité. Par exemple, la consonne Aleph, première lettre de l'alphabet phénicien, devint la voyelle Alpha tout en conservant son graphisme général.

Vers le Ve siècle avant J-C., on assiste à la construction conceptuelle de lettres inscrites dans des formes géométriques. Les grecs les ont simplifiées et géométrisées dans le pays même qui est considéré comme le berceau de la géométrie.

Le sens d'inscription de cette écriture s'effectue primitivement de droite à gauche, puis en boustrophédon, puis à partir du Ve siècle avant, il s'oriente de gauche à droite, ce qui ne se fait pas sans incidence sur la forme des lettres. En effet les lettres tracées ou ouvertes vers la gauche pour une lecture de droite à gauche, seront pour la plupart tournées d'un quart à un demi-tour pour s'inscrire dans un dynamisme orienté de gauche à droite.

Les mots sont collés les uns aux autres n'assurant pas une compréhension très claire du texte. Enfin l'accentuation sera utilisée pour restituer la langue grecque.

Il est enfin possible, grâce à la valeur phonétique de chaque lettre, de prononcer une phrase écrite en alphabet grec avec consonnes et voyelles, même si l'on ne connaît pas la langue. Dès l'adaptation des écritures alphabétiques à d'autres langues européennes que celles auxquelles elles sont destinées, on constate l'élimination des sources acrophoniques et figuratives. La forme des lettres est elle-même la représentation de la parole.

L'alphabet grec est à la source des alphabets coptes et cyrilliques notamment. Il donne naissance à l'alphabet étrusque qui engendrera le latin.

Vers le Ve Siècle av. J.C. la construction conceptuelle des lettres inscrites dans des formes géométriques, ajoute à la perfection de cet alphabet et sont en totale harmonie avec les idées et les philosophies d'alors. Dès cette époque, les inscriptions lapidaires fleurissent sur les monuments. Ces inscriptions lapidaires servent de d'ossature générale applications courantes de l'écriture. La cursivité due à la main, et les raccourcis obligatoires du tracé dynamisent et assouplissent cette capitale géométrique.

La diffusion des œuvres littéraires se faisait principalement de manière orale, mais aussi sous forme écrite, à l'aide d'un stylet sur des tablettes enduites de cire.

C'est grâce aux conquêtes d'Alexandre le Grand que le Grec se répand en particulier dans tout l'empire Byzantin.

Le Latin

Sous cette appellation, nous pouvons ranger actuellement toutes les écritures et typographies qui participent de notre univers écrit «occidental» depuis l'alphabet latin de l'empire romain. Dès la fixation de l'alphabet Caroline sous Charlemagne qui, en son temps, détermina l'écriture et la langue (le latin) pratiquées dans son empire, jusqu'à Gutenberg où la typographie gothique fleurit, en passant par l'époque de la renaissance où l'on reprit les modèles anciens, jusqu'à nos jours, on voit la forme des lettres typographiques évoluer en fonction de la culture des peuples et aussi parfois aussi de leurs modes. «L'écriture - et surtout aujourd'hui l'écriture typographique - est la forme dans laquelle s'incarne l'esprit du temps».

Les témoignages les plus anciens que nous possédions de l'écriture latine remontent au VIIe et VIe siècle avant notre ère.

C'est au cours du VIIe siècle avant J-C., par l'intermédiaire des Étrusques (alphabet de 26 lettres), que les romains vont hériter de l'alphabet grec archaïque pour créer leur alphabet latin.

Entre 80 et 30 ans av. J.C. la langue et l'écriture romaine se stabilisent et l'alphabet romain est très proche de celui des Grecs.

Au IIIe siècle avant J-C. cet alphabet ne comprenait que 19 lettres. Les lettres changèrent de nom, par exemple, l'alpha devint le «a», le bêta le «b», etc. Certaines lettres qui n'avaient pas été utilisées par les romains, complétèrent cet alphabet par la suite. Des signes, dont certains nous sont restés, furent ajoutés pour noter les «renforts» de la langue. C'est le cas des accents, de la cédille, du tilde, de l'esperluète, du é cédillé, etc.

La puissance de l'empire romain imposa l'alphabet latin sur un vaste territoire.

L'emprunt des capitales lapidaires grecques par les romains, se fait avec quelques modifications. En effet la séparation nette des mots est rendue nécessaire pour assurer la compréhension et la mémorisation des textes que les romains mettent au service de leur pouvoir.

La capitale romaine est une lettre de structure statique dont l'utilisation première est monumentale. Les différentes utilisations que les romains faisaient de l'écriture dès le 1er siècle et sans doute bien avant, ont influé sur le style de ces écritures.

La dimension des lettres est adaptée à la distance de lecture et se doit d'assurer la puissance de l'empire.

Pour la première fois, on voit apparaître le souci de l'épaisseur des traits en gravure lapidaire liée à la profondeur de gravure et à l'éclairage solaire.

À l'image de l'écriture grecque, l'écriture monumentale latine est gravée avec des traits d'égale épaisseur jusqu'au Ier et IIe siècle après J-C. Les lettres étaient assez brutes et peu raffinées. Ensuite, on emploiera les pleins et les déliés de l'écriture manuscrite.

Vers le IIe siècle avant J-C. on voit apparaître l'alignement précis du texte et les terminaisons incisées en amorce d'empattements. C'est vers la fin du 1er siècle avant J-C. que sont réellement mis en place les empattements. Tout d'abord fonctionnels pour terminer proprement les verticales de chaque lettre et pour les aligner, les empattements prennent également une fonction décorative.

Ces capitales ont essentiellement évoluées jusqu'à nos jours avec leur emploi à des fins utilitaires liés aux échanges économiques et culturels.

L'écriture livresque, destinée à la notation d'importants volumes de textes, comme pour l'écriture grecque, évolue formellement avec l'accélération de son tracé.

Au début de notre ère, IIe et IIIe siècle, le parchemin (feuille) supplante le papyrus (rouleau) et génère le livre sous sa forme actuelle (codex).

La forme des lettres qui composent l'alphabet latin va évoluer avec les époques et le type d'utilisation que l'on en fera.

La dimension de la lettre diminue et la règle principale est la clarté et la lisibilité.

La quadrata, est l'aboutissement tardif (IVe siècle) de la capitale romaine lapidaire monumentale. C'est une écriture manuscrite soignée.

Ces écritures, tracées autour d'éléments géométriques tels le cercle et le carré, peuvent se mémoriser grâce à cet élément graphique :

La Quadrata

Un tracé plus cursif et plus rapide de ces écritures, capitale romaine et quadrata, va introduire une écriture à structure étroite et verticalisée : la Rustica. Issue des capitales romaines peintes, on la trouve en abondance sur les ruines de Pompéi. C'est le premier emploi démocratique de l'écriture. L'angle, très vertical, de tenue du calame utile à son tracé, induit des déliés verticaux et des pleins plutôt horizontaux. Elle est tracée rapidement et son usage est éphémère. Elle s'étroitesse petit à petit jusqu'au Ve siècle.

On peut la mémoriser grâce à cet élément graphique :

La Rustica

Les lettrés, écrivant souvent le grec avec l'alphabet latin ou le latin avec l'alphabet grec, ont certainement influencé le tracé des écritures manuscrites par l'adoption d'éléments graphiques de l'écriture onciale, présents dans les textes grecs.

L'écriture Onciale apparaît au III^e siècle et dès le IV^e siècle elle sera réservée aux ouvrages précieux et la plupart du temps religieux jusqu'au VIII^e siècle. Écriture sacrée, utilisée dans les monastères, cette écriture s'installe pour très longtemps. De tracé très arrondi, les rapports qu'elle entretient avec l'architecture sont flagrants.

Elle résistera bien aux invasions barbares, car protégée dans les monastères, elle comporte des capitales et des minuscules mélangées.

On peut la mémoriser grâce à cet élément graphique :

L'Onciale

Dans la période pré-carolingienne, à partir des minuscules cursives romaines tracées sur différents supports : les murs, le papyrus ou encore les tablettes de cire, une grande diversité d'écritures est employée, avec la sensibilité des différents pays qui les utilisent.

Du I^{er} au VI^e siècle, différentes écritures vont se croiser et générer des schémas multiples. La minuscule primitive c'est-à-dire la demi-onciale s'était déformée et avait dégénéré en griffonnages, souvent illisibles. À partir du III^e siècle, les liaisons entre caractères se multiplient et, à l'intérieur de la lettre, la rapidité de l'écriture se marque par un plus grand arrondissement des angles.

On sait que l'empire romain entretient des relations avec l'Irlande par exemple. La fondation en 432 de l'église Irlandaise par Saint Patrick, premier évêque et patron de l'Irlande, en est une preuve évidente. Les manuscrits irlandais font apparaître une écriture minuscule dont les verticales courtes sont souvent courbes et dotées de terminaisons triangulaires qui confèrent à cette écriture un style tout particulier. Agrémentée de capitales de structure latine évoquant à l'évidence les écritures runiques, cette écriture est appelée semi-onciale irlandaise.

On peut la mémoriser grâce à cet élément graphique :

La semi-onciale irlandaise

Pendant que l'empire romain se désagrège, les écritures européennes, toutes à peu près à la même époque, vont se libérer des règles de tracé et céder à diverses

fantaisies dues aux inflexions de la pensée et à l'accélération de la notation, allant jusqu'à l'illisible, sous les mérovingiens du Ve au VIIIe siècle.

LA CAROLINE

Fort de l'appui de l'Église et des circonstances qui en font le seul héritier de la couronne, le prestige de Charlemagne est immense.

On lui attribue, sans doute à juste titre, l'unification de son empire. Mais en ce qui concerne l'aspect culturel, on peut être assuré que c'est au moins Alcuin (écolâtre d'origine anglo-saxonne) dont il s'était attaché les services, que revient l'introduction des arts libéraux, c'est-à-dire le trivium (grammaire, rhétorique, dialectique), et le quadrivium (arithmétique, musique, géométrie, astronomie).

Charlemagne aidé d'Alcuin entreprend donc de résoudre pour toute l'Europe un problème d'unification dont la méthode peut être énoncé d'une façon très simple : «Une seule langue pour tous les clercs de l'Europe : le latin; une seule écriture : la caroline». Cette décision datée de 789 tend, à dissiper l'ignorance, à reprendre le terrain perdu par le latin et à faire régner l'ordre et la clarté.

La minuscule carolingienne issue des modèles des grands scriptoria : Autun, Corbie, Luxeuil, Lyon, Tour, donne une forme quasi définitive aux a, g, n, r, écrites jusque-là de façon très anarchique. Elle est le tardif aboutissement de l'évolution des écritures cursives. Les graphismes nationaux, dissemblables et devenus illisibles furent remplacés par cette nouvelle écriture, dont les lettres sont de petit module, bien séparées les unes des autres, avec un espace suffisant entre les mots. Son dessin s'imposera, beaucoup plus tard, au XVe siècle, comme «bas-de-casse» typographique. La caroline a été un instrument d'unification de la pensée Occidentale chrétienne.

La construction de ces caractères en traits courts et fragmentés rappelle la façon d'écrire des romains, à petits coups de poinçon sur de la cire, mais avec toute la rondeur nécessaire à l'élaboration de ces modèles.

La consécration officielle de la caroline n'allait pas entraîner automatiquement la disparition des capitales romaine, rustica ou onciale, qui serviront aux initiales de couleur rouge brique ou or (ce travail revient au Rubricator).

Les livres étaient écrits sur parchemin mais, pour les écritures courantes, on achetait du papyrus en rouleaux. On écrit à partir de ces périodes avec des stylets de cuivre rouge, d'or, d'argent ou de plomb. Les «mines en plomb» gainées de cuir, peuvent être considérées comme les portemines d'alors.

S'il ne fallait conserver qu'une seule lettre pour symboliser et mémoriser l'alphabet complet en minuscule caroline le «a» serait certainement choisi :

La Caroline

BIBLIOGRAPHIE SUCCINTE

- BLACKWELL (Lewis). - Typo du XXe siècle. - Paris, Éditions Flammarion, 1994.
- CALVET (Louis-Jean). - Histoire de l'Écriture. - Pluriel-Hachette littérature, 1998.
- COHEN (Marcel). - La Grande invention de l'écriture et de son évolution. - Paris, Imprimerie nationale/Librairie C. Klincksieck, 1958. 3 vol. (Textes, documentation et index.)
- DRUET (Roger) et GRÉGOIRE (Herman). - La Civilisation de l'écriture. - Paris, Desain et Tolra/ A. Fayard, 1976. Rééd. 1980. 312 p., ill. n. et c.
- FRIEDL (Friedrich), OTT (Nicolas), STEIN (Bernard). - Typo. - Könemann, 1998.
- JACKSON (Donald). - Histoire de l'écriture. - Paris, Denoël, 1982. 176 p., ill. (The Story of writing, 1981.)
- JEAN (Georges). - L'Écriture, mémoire des hommes. - Paris, Gallimard, 1987. 224 p., ill. (Coll. Découvertes.)
- L'Aventure des Écritures. - Bibliothèque nationale de France, 1997. - 223 p., ill.
- MALLON (Jean). - De l'écriture. Recueil d'études publiées de 1937 à 1981. Liminaire de Jean Glénisson. - Paris, C.N.R.S., 1982. 31,5 cm, 367 p., ill.
- MANDEL (Ladislas). - Écriture, miroir des hommes, Reillanne, Atelier Perrousseau Éditeur, 240 p., ill.
- MÉDIAVILLA (Claude). - Calligraphie. Du signe calligraphié à la peinture abstraite. Préfaces de Pierre Restany et Gérard Xuriguera. - Paris, Imprimerie nationale, 1993. 336 p. ill. n. et c.
- NAISSANCE de l'écriture. Cunéiformes et hiéroglyphes. (Béatrice André-Leicknam et Christine Ziegler). - Paris, Réunion des musées nationaux, 1982. 24 cm, 384 p., ill.
- OUAKNIN (Marc-Alain). - Les Mystères de l'Alphabet. Paris, Éditions Assouline, 1997. 15,5x21,5, 382p., ill.
- PAPUT (Christian). - Vocabulaire des Arts Graphiques, de la Communication, de la PAO, etc. TVSO Éditions, 1997, 2002. 15x21, 288p.
- VENTRIS. Histoire de l'écriture. Paris Le Seuil, 1994.

La calligraphie Tifinaghe : une expression artistique en construction

Lahbib Fouad

Au cœur du débat animé autour de la graphie Tifinaghe et dans le cadre de l'origine et de l'évolution de cette graphie, existe-il des relations de parenté entre les symboles de l'art graphique traditionnel amazighe et la graphie tifinaghe ? Le symbolisme - en général - précède-il le graphisme au cours de l'évolution de la pensée humaine ? Quelles sont les grandes étapes de la genèse et de l'évolution de la graphie tifinaghe ?

Pour essayer de répondre en partie à ces questions, je me propose dans un premier temps, d'observer brièvement les grandes étapes de l'évolution de la production graphique chez l'enfant et d'analyser les similitudes que peut avoir cette évolution avec la genèse et l'évolution du graphisme produit par les premiers habitants de l'Afrique du Nord. Ensuite, dans un deuxième temps, passer en revue les grandes étapes de l'évolution et du perfectionnement de la graphie qui a engendré l'alphabet et la calligraphie Tifinaghe actuelles.

Sachant que la production graphique chez l'enfant passe par plusieurs étapes d'apprentissage et de persévérance, et que sa maîtrise évolue avec l'âge et selon les capacités psychomotrices de l'enfant, observons quelques graphismes produits par des enfants de différents âges.

Avant de maîtriser l'écriture, l'enfant commence, à l'école maternelle, par produire des traces graphiques libres et théoriquement sans aucun sens comme le montre ce gribouillage libre de Youssef (4ans, maternelle) :



L'enfant passe par la suite à une étape de gribouillage où il crée des signes et des graphismes sommaires auxquels il attribue plus ou moins des interprétations. Ce sont ses premiers pas vers le symbolisme et le dessin. Voici, selon Youssef (4ans, maternelle), le dessin de sa « maman » :



En général, à la fin de la maternelle, l'enfant s'aperçoit de la diversité des formes graphiques et parvient à en reproduire les plus simples. C'est souvent pendant cette période qu'apparaît la première tentative de l'écriture qu'il ne découvrira vraiment qu'en classes du primaire.

Entre cinq et sept ans, la forme des lettres semble être maîtrisée par l'enfant sans pour autant donner de l'importance ni à l'ordre, ni à la trajectoire des lettres. Les lettres recopiées et reproduites par l'enfant peuvent être inclinées, renversées, exagérées, etc. A cet âge, le système psychomoteur de l'enfant n'est pas encore perfectionné pour exécuter une tâche délicate comme l'écriture. L'hésitation et l'imprécision caractérisent le graphisme pendant cette phase de l'apprentissage⁽¹⁰⁰⁾. C'est le cas de Hassan, 5 ans, qui, en recopiant la lettre Z de tifinaghe, inverse le sens de l'arc supérieur de la lettre.



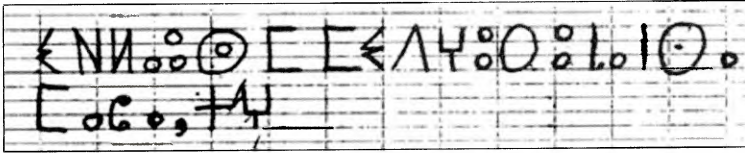
A un niveau scolaire plus avancé et après un long apprentissage, l'élève se dote d'une motricité plus élaborée qui met en œuvre simultanément la mémoire (la forme des lettres) et le geste qui reproduit ces graphismes. Plus tard, l'enfant perfectionne son geste, et l'écriture devient de plus en plus automatique. Le geste graphique se met alors spontanément au service de l'écriture.

Nous avons eu l'occasion d'animer plusieurs ateliers de Calligraphie Tifinaghe au profit des enfants de 7- à 14 ans et de donner des cours d'alphabétisation en

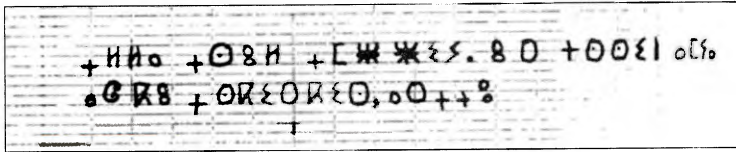
(100) Pendant le préscolaire, les difficultés psychomotrices peuvent décourager l'enfant non assisté et peuvent le pousser à éviter même la découverte du plaisir du graphisme et par conséquent celle de l'écriture.

Tifinaghe pour adultes. L'exercice consiste à recopier des mots, à écrire une phrase ou à produire librement un texte en utilisant la graphie Tifinaghe. Voici les extraits de quelques travaux :

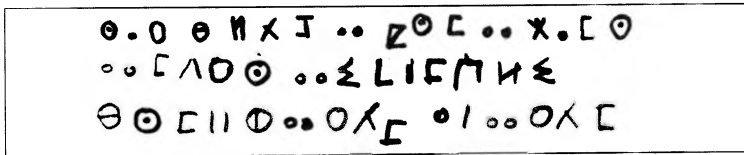
Meryam (7ans), en recopiant une phrase, ne s'aperçoit pas que les lettres « L » (L) du premier mot sont identiques, elle inverse la barre de liaison du premier. Pour elle, la lettre « S » (S) peut contenir un point comme elle peut contenir un petit rond :



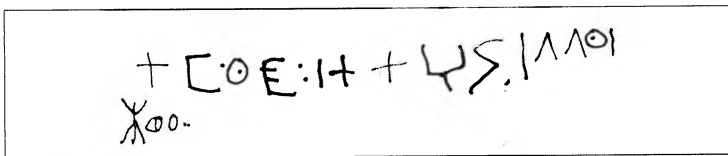
Pour Nisrine (9 ans) le « T » (T) est systématiquement marqué sous la ligne comme le signe « plus » (+) et les espaces entre les mots, comme pour Meryam sont souvent ignorés.



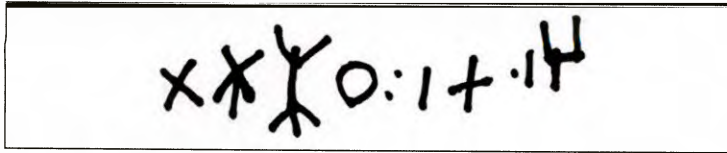
Les lettres sont parfois en indice, parfois en exposant et l'espace entre les mots sont systématiquement ignorés comme le montre cette production libre de Hassan (10 ans, auto-alphabétisé) :



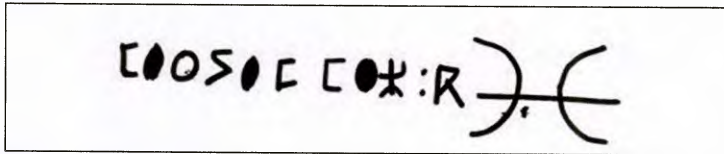
Zahra (13 ans) va même jusqu'à ajouter des barres supplémentaires à la lettre « ä » (D) :



Rqiya (personne âgée), analphabète, en recopiant les mots « taxzrunt nnv » (notre fibule) inverse la barre de la lettre « x » (X) et exagère la taille des lettres « z » (Z) et « v » (Gh) par rapport aux autres dans le même mot.



Pour improviser une solution personnelle de la distinction entre la lettre « a » (A) et la lettre « r » (R), Maryam (10 ans) en écrivant son nom, noircie les « a » (A) tout en lui gardant la même taille que « R » (R) :



De ce bref examen des travaux des élèves et des principales étapes de l'évolution du graphisme l'âge de l'enfant, on constate que :

- ❑ la trace graphique (le gribouillage) évolue en signes abstraits (les symboles) avant que l'enfant découvre le dessin d'une part et l'écriture de l'autre.
- ❑ l'activité graphique évolue avec l'âge et se développe selon le besoin de communication par l'écriture.
- ❑ l'épanouissement des capacités de la production graphique dépend de l'apprentissage.
- ❑ l'activité graphique met en œuvre plusieurs facteurs simultanément. Le plus important de ces facteurs est la mémorisation du geste graphique et la signification affectée à ce graphisme.
- ❑ quel que soit l'âge où commence l'initiation à l'écriture, il semble que le progrès dépend des efforts fournis pendant l'apprentissage.

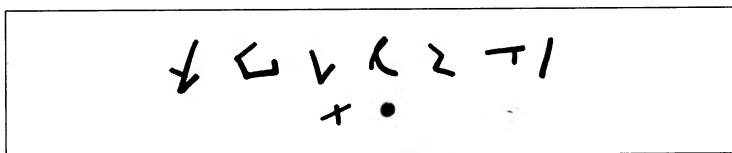
Remarquons au passage que la production graphique met en oeuvre plusieurs facteurs dont : le geste qui engendre la trace et le graphisme, l'action de la main, de l'outil (plume, pinceau...), le support (feuille, toile...), la matière (encre, peinture...), etc. Ces facteurs, essentiels en typographie traditionnelle semblent être sérieusement bousculés par le micro-ordinateur qui réduit à un « clic » sur le bouton de la souris la plupart de ces actions ! La création graphique semble abandonner le geste du corps, la réalité de la matière, la palpabilité de l'espace, la visibilité du support, etc., pour laisser la place à une création robotique, fictive et virtuelle.

Voyons à présent comment le graphisme a évolué dans une échelle plus grande du temps, celle de l'histoire de l'Homme.

En Afrique du Nord, les gravures et les peintures rupestres témoignent d'une activité graphique intense de l'homme préhistorique. La durée de vie d'une trace graphique dépend de la matière et du support sur lequel elle est réalisée. Un support - ou matière - peut être de mémoire courte comme le sable, la terre, la peau (tatouage)... ou de mémoire longue comme la pierre gravée, le papier, la peinture...

Dans plusieurs supports à mémoire longue, l'Homme préhistorique a laissé des traces graphiques. La plupart de ces empreintes graphiques sont peintes ou gravées sur la roche. Ces traces ont des formes souvent sommaires et non identifiées : parfois des croix, des lignes brisées, des losanges...⁽¹⁰¹⁾ Ce sont en général des traces qu'on peut trouver isolées ou accompagnées d'autres gravures.

Voici quelques exemples de traces graphiques relevées dans les sites rupestres de Tizi-n-Tirghist, et de Yagour (Haut Atlas, Maroc) d'après les photos de la Planche n° 1.



L'activité graphique de l'homme préhistorique aurait commencé par une production de traces libres non définies, qui pourraient être la base de départ d'un dessin ou d'un signe.

Plus tard, ces traces sommaires se précisent de plus en plus et évoluent en dessins plus ou moins élaborés. C'est ce qui caractérise en général les gravures et peintures rupestres.

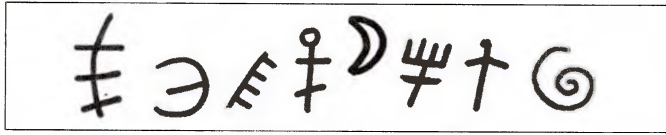
Plusieurs ouvrages ont été consacrés à l'art rupestre en Afrique du Nord. Cette activité artistique est abondante, riche et très élaborée. Elle caractérise des périodes plus avancées de l'histoire du graphisme nord-africain. Il s'agit de peintures d'abord archaïques, puis de plus en plus parfaites. Ces dessins représentent en général l'environnement naturel et la vie quotidienne de l'artiste.

Pendant cette étape, les artistes de la préhistoire semblent découvrir, à côté du « naturalisme », une autre expression graphique : « le symbolisme ».

Le symbole est un graphisme gravé seul ou reproduit plusieurs fois sur une même dalle ou sur plusieurs dalles éloignées. Ce sont des gravures énigmatiques ayant vraisemblablement un pouvoir magique et une fonction de communication.

(101) - Malika Hachid. Les Premier Berbères. Aix-en-Provence, Edisud - 2001

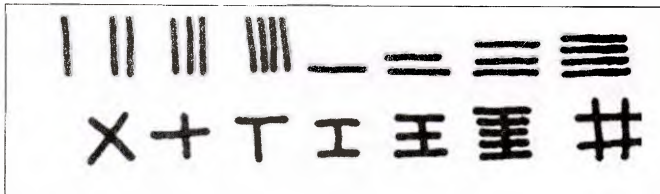
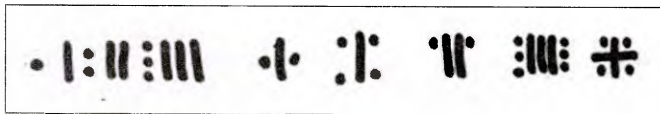
Voici quelques signes relevés dans les sites rupestres de Yagour, Tayanant, Tirghist (Haut Atlas) d'après les photos des planches 1, 2 et 3.



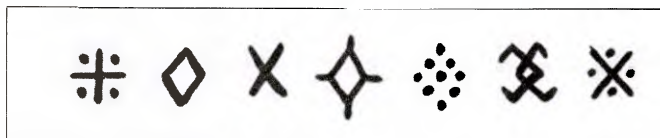
Sur certaines parois de Tizi-n-Tayanant, la spirale est gravée une seule fois, alors que le croissant est répété des dizaines de fois sur le même site. A Yagour (Haut Atlas de Marrakech), la roue « solaire » est répétée des dizaines de fois sur plusieurs dalles à des endroits différents.

Ces symboles pourraient être utilisés aussi par leurs auteurs pour marquer leurs bêtes et tatouer leurs corps. Ils seraient des marques d'appartenance (tribale, de possession d'objets ou d'animaux...) et vraisemblablement, ils auraient une fonction de communication.

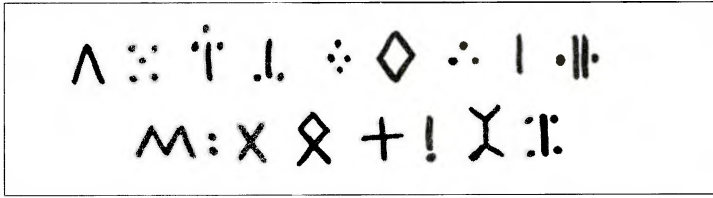
Les premiers tatouages auraient probablement la même fonction. Ce sont des signes simples basés sur un jeu de points et de traits ou des signes composés et plus ou moins élaborés :



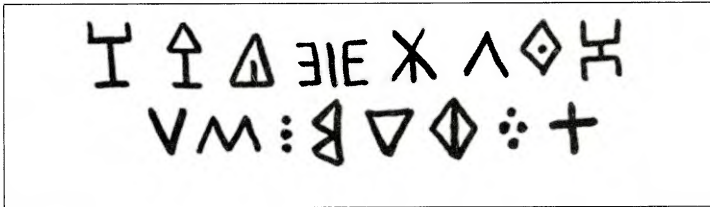
Le signe, en se développant de plus en plus, se diversifie et gagne d'autre support. Il sera utilisé en poterie, en bijouterie et autres artisanats. Voici à titre indicatif les motifs de tissage relevés sur un seul tapis du Haut Atlas (d'après la photo de la planche 4) :



Ces signes se retrouvent sous d'autres formes en architecture et en décoration... Les motifs relevés sur une seule façade d'un bâtiment à Errachidia (d'après la photo de la planche 5) montre la diversité de ces formes :

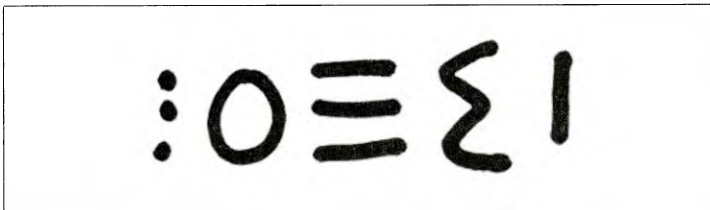


Les mêmes signes géométriques qui caractérisent la décoration artisanale se retrouvent en peinture murale, en décoration de plafonds... comme le montre ces motifs relevés à l'intérieur d'une maison de Kabylie sur le mur décoré d'une seule chambre (planche 5).



On constate que les signes et les symboles géométriques des arts traditionnels évoquent de très près les ébauches de signes alphabétiques utilisés aujourd'hui en écriture Tifinaghe.

Ces signes se multiplient en se diversifiant. Ils aboutissent au sommet de leur complication à des inscriptions. Un grand nombre de dalles nord-africaines sont gravées par ces écritures. Voici un fragment de l'inscription de Tinzouline / Zagora (d'après la photo de la planche 6) :



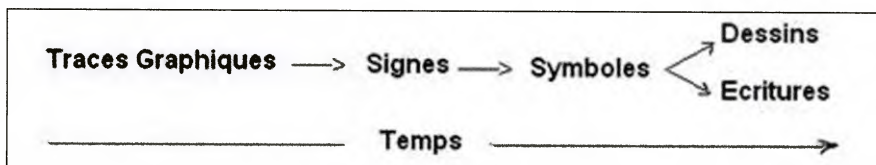
Ces écritures, utilisées depuis le néolithique jusqu'à nos jours, suivant des formes plus ou moins différentes selon les régions, constituent la base de l'alphabet Tifinaghe actuel (Voir les différents alphabets dans la planche 7).

Après cette brève présentation des principales étapes de l'évolution du graphisme en Afrique du Nord, on peut affirmer que sur une échelle du temps, l'échelle historique, la production graphique évolue en même temps que le développement intellectuel de l'homme et le progrès de ses moyens de production.

La trace graphique a évolué en dessin d'une part et en signes et symboles de l'autre. Les graphismes ayant évolué en signes pouvaient représenter des idées donc être utilisés à des fins de communication. Le signe a donc engendré, après une longue évolution, l'alphabet et l'écriture.

La trace graphique aurait donc évolué en dessin (lorsqu'elle représente une idée) et en écriture lorsqu'elle a une relation avec le signe abstrait.

En comparant l'évolution du graphisme dans le temps et le développement de l'activité graphique chez l'enfant, on peut constater que l'évolution de la production graphique à l'échelle humaine semble être récapitulée par les étapes de la maîtrise du graphisme chez l'enfant. Le schéma suivant montre les étapes de cette évolution:



Voyons à présent les grandes étapes de la construction de la graphie Tifinaghe actuelle.

Une grande partie de l'alphabet Tifinaghe dérive des inscriptions rupestres. Cette écriture, restée pendant longtemps exclusivement touaregue (Sahara), devient de plus en plus généralisée dans toute l'Afrique du Nord. Un bref historique de l'évolution de cet alphabet montre qu'elle a subi des transformations dans le temps et dans l'espace.

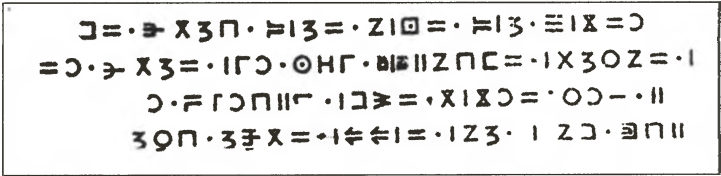
Dans les montagnes de Kabylie (voir planche 8), dans le massif du Hoggar, ainsi que dans le Yagour au Haut Atlas (Maroc), aux Canaries et dans toute l'Afrique du Nord, les plus anciennes inscriptions connues sont toutes libyque et libyco berbères⁽¹⁰²⁾. Malgré les différences entre l'alphabet saharien, occidental ou oriental..., le fond graphique reste pratiquement le même (cf. planche 7).

Le Libyque est un alphabet géométrique qui s'écrit en général verticalement. En observant les inscriptions de Anu-n-Elias au Niger (Planche 9), on comprend que le sens horizontal fait plutôt l'exception.

Ces inscriptions libyques ne se trouvent pas seulement gravées sur des dalles anonymes ; plusieurs dédicaces sous forme d'épitaphes ont été gravées sur les temples des Rois berbères. La plus célèbre est celle dédiée à Massinissa car elle est bilingue Punique-Libyque⁽¹⁰³⁾ :

(102) - Un inventaire des inscriptions rupestres du Maroc est brillamment présenté dans un ouvrage récemment publié. Tirra. Aux origines de l'écriture au Maroc. A. Skounti, A. Lemjidi, M. Nami. Publication de l'IRCAM, 2003.

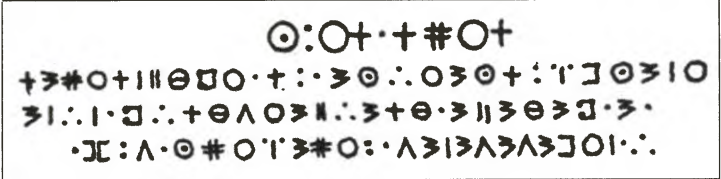
(103) - Les inscriptions libyques bilingues de l'Afrique du Nord. 1936, G. Marcy.



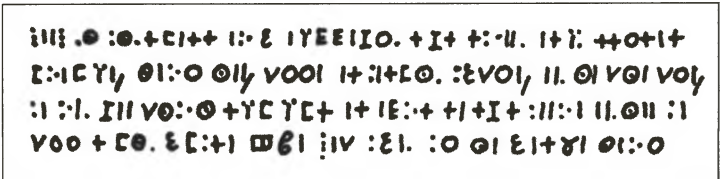
La nature des supports, les outils de gravure et la dureté des supports ont vraisemblablement privilégié le mode de gravure exigeant une forme géométrique des lettres. Cette forme géométrique caractérise l'alphabet Libyco-Berbère et le Tifinaghe en général, la forme géométrique étant plus facile à graver et fait ressortir toutes les caractéristiques de la lettre.

Cet alphabet a été perpétué par les femmes touaregues et enseigné aux enfants sur le sable. Celui-ci, support pourtant à mémoire courte, participait à la transmission de ce savoir ancestral d'une génération à une autre.

Une étape plus récente de l'évolution de la graphie Tifinaghe est celle des manuscrits. Cette étape est connue après l'introduction du papier au Sahara pendant l'époque coloniale. Des textes et des dictionnaires furent transcrits en graphie Tifinaghe Touaregue. L'Evangile a même été traduit et transcrit en cette graphie. Voici un passage traduit en Tamachek par la Société Nationale Biblique d'Ecosse.



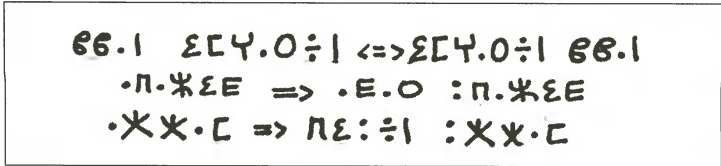
L'alphabet utilisé dans ce manuscrit est le Saharien ancien transcrit de droite à gauche. Le Petit Prince, célèbre conte d'Antoine de Saint-Exupéry fut également traduit en Tamachek et transcrit en Tifinaghe Touaregue :



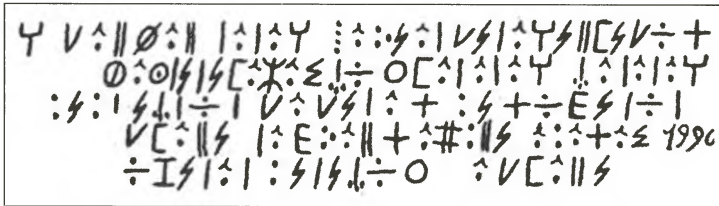
Les plus anciens manuscrits en Tifinaghe sont décidément touaregs. Ce peuple saharien a toujours utilisé cette graphie sur différents supports. Mais, cet alphabet ne fut vraiment diffusé qu'en 1964 par l'Associations Agraw imazighen appelé «Académie Berbère ». Cette Association a augmenté le nombre de graphèmes et phonèmes pour généraliser Tifinaghe à l'ensemble des dialectes amazighes. Estimant que les graphèmes composés de points et de traits étaient peu pratiques, l'ACB (Association Culturelle Berbère) les remplace par des graphèmes inspirés de

l'alphabet latin ou arabe (k, h, x, q, d ...) et y ajoute des sons jusqu'alors inconnus chez nombreux Imazighen (p, v...) : C'est les Néo-Tifinaghes. Agraw Imazighen fut le catalyseur de la diffusion de cet alphabet dans tous les pays de l'Afrique du Nord.

Voici l'extrait d'une leçon tirée du cahier d'initiation au Tifinaghe, revue « Imazighène » n° 65.

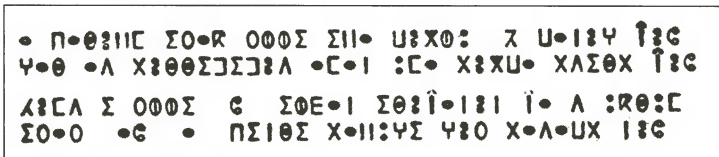


La diffusion de l'alphabet Tifinaghe Agraw Imazighen n'a pas empêché plusieurs formes locales de cet alphabet de se maintenir et de se développer. L'écriture diffère d'une région à une autre et quelque fois d'un auteur à un autre. On assiste à une hésitation entre le maintien de la tradition graphique et l'influence des réformes apportées par Agraw Imazighen. Ce système a subi plusieurs améliorations notamment pour la notation des voyelles. Un extrait d'un courrier personnel provenant d'un calligraphe du Niger en 1996, montre une utilisation simultanée de caractère Libyque et Néo-Tifinaghe.



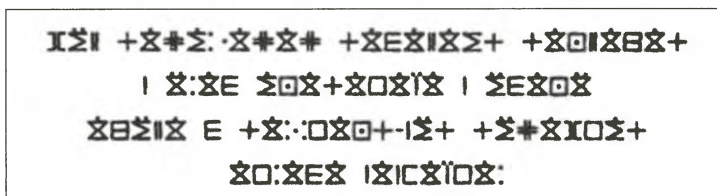
En analysant ces différents manuscrits, on s'aperçoit du caractère isolé des graphèmes de chaque mot. Les lettres ne sont jamais liées les unes aux autres. En plus, partout, la ponctuation et l'espace entre les mots ne sont toujours pas évidents.

Pendant les années 70, des intellectuels travaillant autour de la revue «Amazigh» décident de créer une machine pour écrire des textes en Tifinaghe. Ce fût la première machine à écrire en Tifinaghe. L'utilité de cette machine était limitée et de courte durée surtout après l'apparition des fontes informatiques.



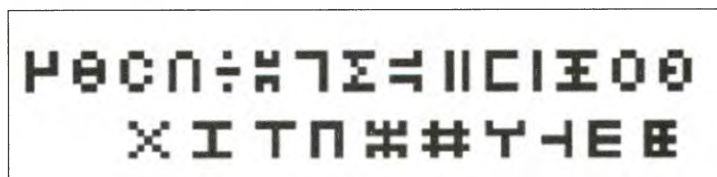
D'autres auteurs créent leurs propres machines à écrire en Tifinaghe. La police était évidemment différente d'une machine à une autre comme le montre cet extrait

du bulletin touareg Mns Ihgwugwun « le chameau bègue » n° 4, produit et diffusé par le Poète calligraphe Hawad.



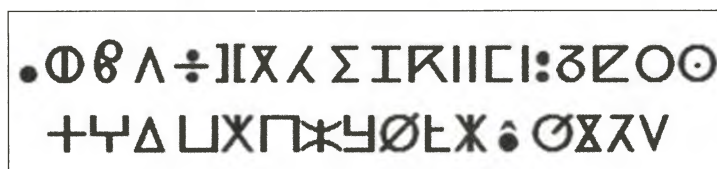
L'utilité de ces machines était limitée et de courte durée. Elles furent sévèrement concurrencées par l'apparition des fontes informatiques.

En 1989, Arzqi Nait Abedallah de l'Université Western Ontario met au point le premier logiciel de traitement de texte appelé « Awal Amazigh » conçu pour écrire en caractères Tifinaghe (ou latin). Le code produit par ce logiciel est un code ASCII très proche du code ASCII standard. La conception de la police Tifinaghe utilisée dans ce programme a largement puisé dans le répertoire de l'alphabet Libyque antique.



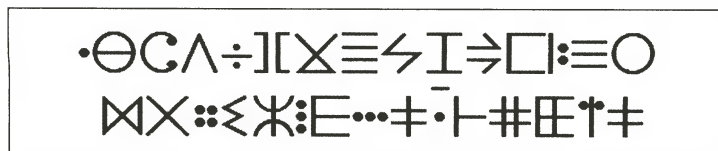
Bien que ce soit un programme très simplifié pour le traitement de texte, ce logiciel a le mérite d'être le seul jusqu'à aujourd'hui qui propose une édition et une écriture en Tifinaghe avec un environnement à menus entièrement en langue amazighe.

La généralisation de l'outil informatique fait apparaître une génération de concepteurs de polices de caractères pour micro-ordinateur. C'est la naissance des fontes informatiques. Les premières fontes créées sont basées sur le système diffusé par Agraw Imazighen. La police la plus populaire était « Afus deg Fus » :

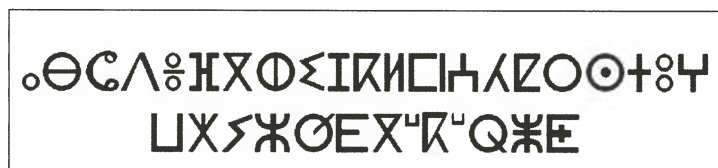


Plusieurs autres polices ont vu le jour, elles étaient conçues sur le même modèle de Afus deg Fus: Exemples : Tawiza, Itri, Yugerten... Les unes destinées pour les «PC» les autres pour les « Mac » Les concepteurs de polices Tifinaghes ont même créé des fontes élaborées pour des usages esthétiques (voir planche 10)

En 1994, l'INALCO (Institut National des Langues et Civilisations Orientales) tente de normaliser informatiquement l'écriture Tifinaghe. S. Chaker propose de remplacer certains graphèmes diffusés par Agraw Imazighen par des lettres puisées directement dans l'alphabet Libyque antique, ce système a été élaboré par Yanis Haralambous ⁽¹⁰⁴⁾ :



En 2002, l'IRCAM (Institut Rpyal de la Culture Amazighe) aménage et simplifie le système de notation proposé par Agraw Imazighen. Il standardise l'écriture Tifinaghe dans le système scolaire au Maroc. C'est la police Tifinagh-Ircam.



En 2004, la graphie Tifinaghe est identifiée numériquement selon le standard universel « UNICODE » Elle fût reconnue officiellement comme faisant partie du «Plan Multilingue de Base » par l'ISO (Organisation Internationale de Standardisation).

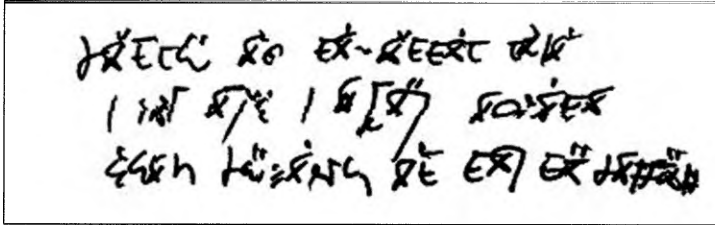
Une police de caractère « Hapax ber.ttf » est en élaboration où les caractères sahariens, touaregs et Néo-Tifinaghe sont répertoriés et codés dans une même police « pan-berbère ».

Le micro-ordinateur a simplifié l'écriture, mais il a confisqué le geste qui engendre la trace graphique. Certains artistes essayent de perpétuer la tradition manuscrite de Tifinaghe dans son volet artistique et affectif en cherchant à concevoir une écriture cursive. Ces artistes estiment que contrairement au traitement de texte sur micro-ordinateur, avec l'écriture cursive manuelle, on peut repérer le mouvement et la conduite de la main qui produit la trace graphique donc le comportement et le tempérament de son auteur. On peut ainsi relever les appuis, les relâchements, la rapidité, les hésitations... C'est ce qui caractérise le style du tracé et distingue « la griffe » de chacun, côté affectif bousculé totalement par le clavier des micro-ordinateurs.

(104) - Un système Tex Berbère. Etudes et Document Berbères, 11. 1994 : pp. 43-53.

Cependant, l'écriture Tifinaghe a toujours été non cursive comme le montre les divers manuscrits traditionnels cités plus haut. L'écriture cursive exige la mémoire de la main levée, le tracé unique, la liaison entre les lettres etc. Les utilisateurs préfèrent l'écriture scripte, plus facile à maîtriser et plus répandue dans les manuels de lecture. Le Tifinaghe cursif demeure de ce fait une pratique personnelle, intime et différente d'un artiste à un autre.

Hawad, le célèbre artiste et poète touareg, est l'un des précurseurs de l'écriture en Tifinaghe cursif :



D'autres artistes en herbe enrichissent l'expression artistique à base de Tifinaghe par des signes et figures puisées dans l'univers du symbolisme amazighe. (Voir Planches 11).

Nous avons eu l'occasion d'animer plusieurs ateliers de Calligraphie Tifinaghe au profit des enfants. Voici les travaux de certains d'entre eux qui ne manquent pas de talent (Voir Planche 12)

La Calligraphie Tifinaghe devient de plus en plus une expression artistique basée sur la maîtrise du style et de la trace des signes. La Calligraphie s'exprime par la forme et le geste des traits. Elle évoque le mouvement et la forme des êtres sans préciser leur nature. La calligraphie anime une harmonie entre le trait, le geste et l'espace.



Calligraphie de Yeschou

Bibliographie :

- Aghali-Zakara, M. *Alphabets Libyco-berbères et informatique*. La Lettre du RILB, N°3, 1997.
- Akpo, C. *Le berceau des écritures*. Jeune Afrique n° 1938, mars 1998.
- Camps, G. *Ecriture libyque*. Encyclopédie Berbère 17, 2565-2573.
- Chaker, S. *Textes en Linguistique Berbères*. Editions du CNRS, Paris, 1984.
- Claudot, H. Tifinar'. *Du brin à la plume*. Daufin, Etudes et Documents Berbères, 6, pp. 187-190.
- Hachid, M. *Les Premiers Berbères. Entre Méditerranée, Tassili et Nil*. Ina-Yas. Edisud, 2001.
- Haralambous, Y. *Un système Tex Berbère*. Etudes et Documents Berbères, 11. 1994 : pp. 43-53.
- Hawad. *Buveurs de braises*, Ed. bilingue touareg/français, Meet, Saint-Nazaire, 1995.
- Hawad. *Caravane de la soif*, Edisud, Aix-en-Provence, 1987.
- Hawad. *Chants de la soif et de l'égarément*, Edisud, Aix-en-Pronevce, 1987.
- Hawad. *Détournement d'horizon*, éd. Grèges, Montpellier, 2002.
- Hawad. *L'Anneau-Sentier*, Ed. de l'Aphélie, Céret, 1989.
- Hawad. *Le coude grinçant de l'anarchie*, Paris-Méditerranée, Paris, 1998.
- Hawad. *Notre horizon de gamelles pour une gamelle d'horizons*. Paris-Méditerranée, Paris 2001.
- Hawad. *Sahara Vision atomique*. Paris-Méditerranée, Paris 2003.
- Hawad. *Testament Nomade*, Sillages, Paris, 1987, rééd. Amara, 1989.
- Imazighène, revue de Agraw Imazighen, Paris, mars 1998.
- Johannes Gutenberg. Le caractère.
http://fr.encyclopedia.yahoo.com/article/ni//ni_3191_p0.html
- Lhot, H. *Les gravures rupestres de l'Oued Djerat (tassili-n-Ajjer)*, Alger, 1976.
- Lhote, H. *Les gravures de l'Oued Mammanet*. Les Nouvelles Editions Africaines. 1979.
- Marcy, G. *Les inscriptions libyques bilingues de l'Afrique du Nord*. Imprimerie Nationale, 1936.
- Nait Abdellah, A. *Awal amazigh*, Université Western Ontario, 1988.
- Paun Silvia. *Signes et messages*. Revue Agadir O'fla n° 6, mai 2004.
- Sadeg, R. *Motifs berbères*. Imprimerie l'Artisan. Boufarik, 1991.
- Siraj, A. *Aux origines de l'écriture au Maghreb*. L'Essentiel, février 2002.
- Skounti, A., Lemjidi A., Nami M. *Tirra. Aux origines de l'écriture au Maroc*. Publication de l'IRCAM, Rabat 2003.
- Yacine, T. *Tradition et Modernité dans les sociétés berbères*. Actes de la Table Ronde. Editions Awal. Paris 1989.



Gravures et inscriptions de Tizi-n-Tirghist (Haut Atlas)

Planche 1



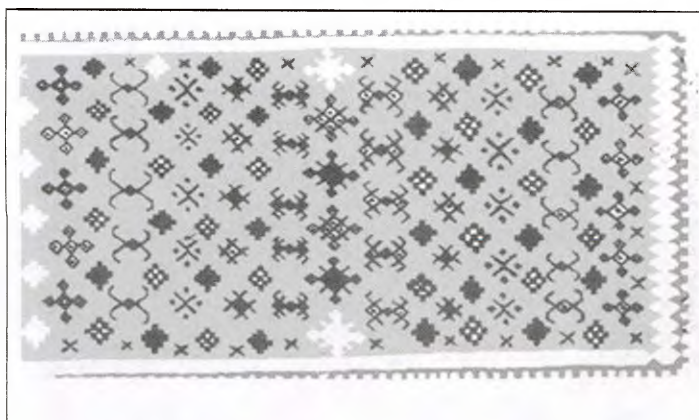
Signes et symboles gravés sur des dalles de Yagour, Haut-Atlas

Planche 2

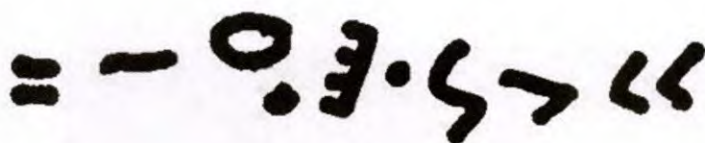


Signes gravés sur des dalles de Azib-n-Ikkis (Yagour, Haut Atlas).

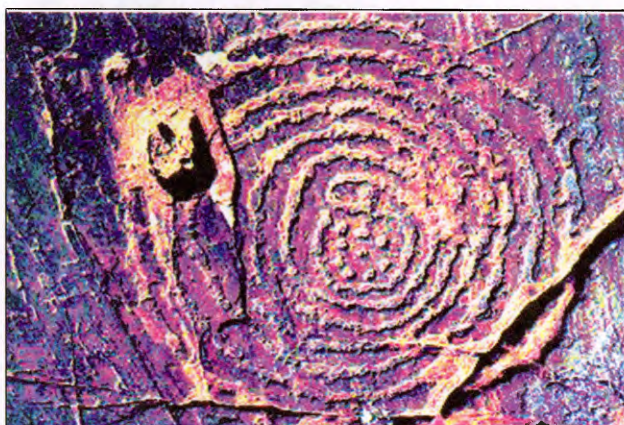
Planche 3



Tapis décoré par Motifs (Haut Atlas).



Inscription (Oukaymden, Haut Atlas).



Spirale de Tizi-n-Taynant (Haut Atlas).

Planche 4



Décoration murale à Errachidia (Maroc).



peinture murale à TisiOuzou(Algérie)

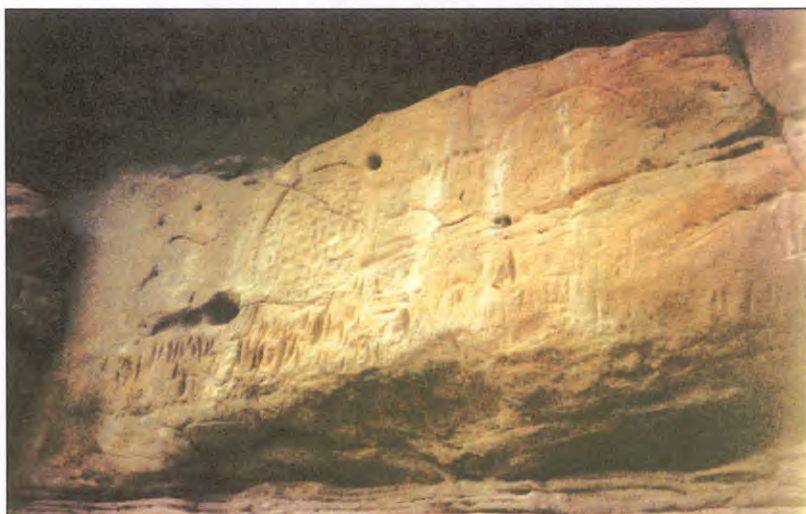


Décoration du plafond (roseaux peints) à Figuig (Maroc).

Planche 5

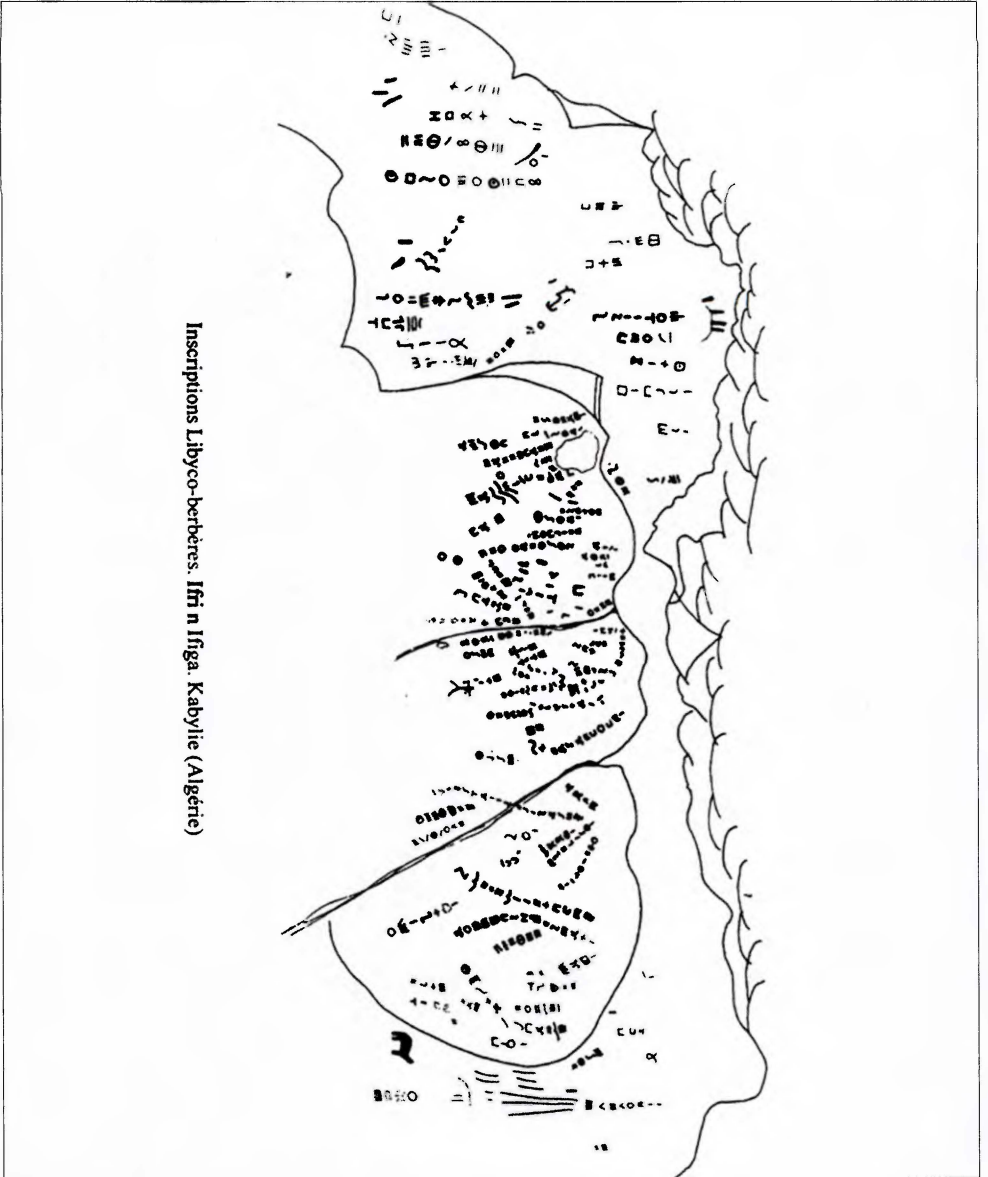


Inscriptions à Tinzouline (Zagora).



Roues Solaires et Croissants (Tizi-n-Taynant, Ouarzazate).

Planche 6



Inscriptions Libyco-berberes. Ifri n Ifga. Kabylie (Algérie)

Planche 8

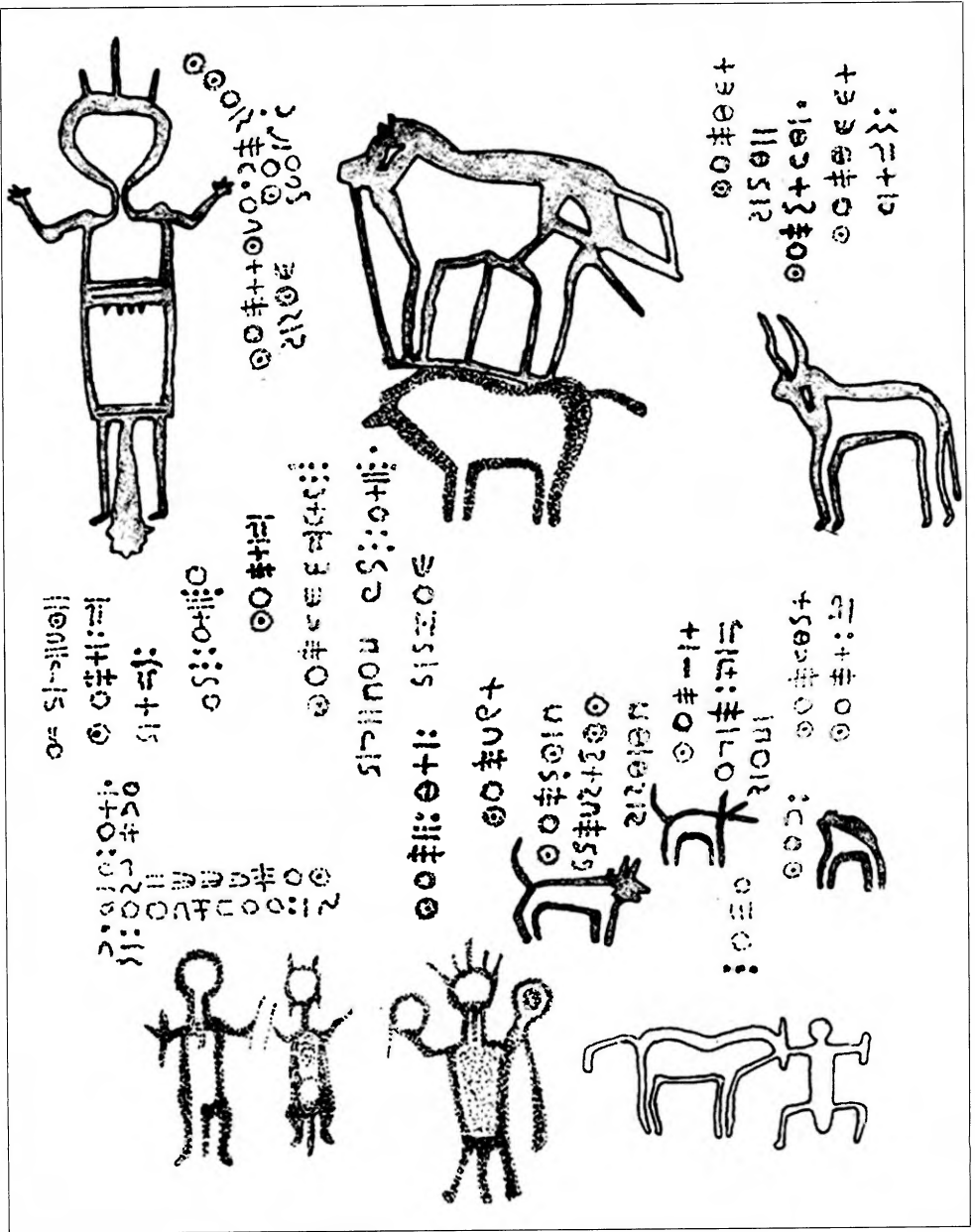
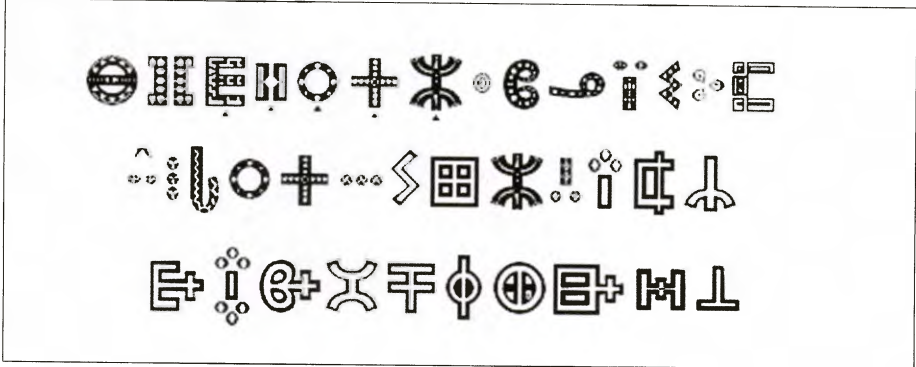
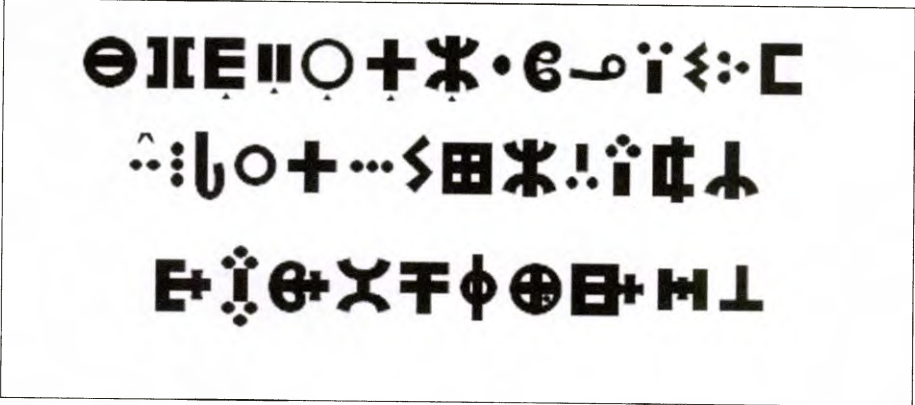
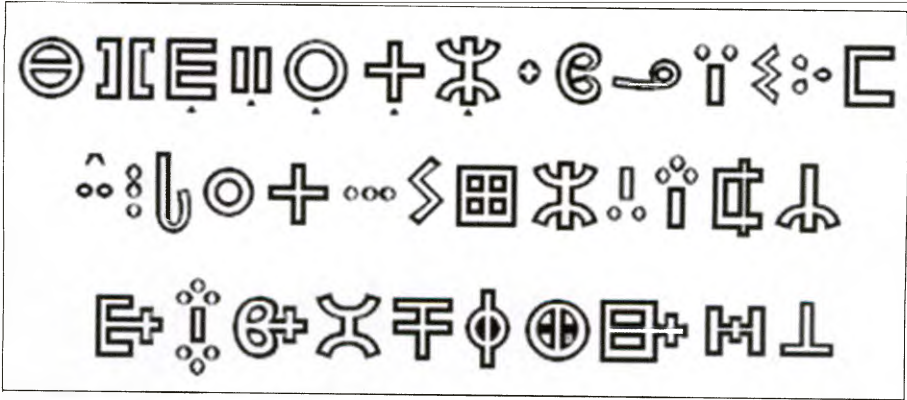


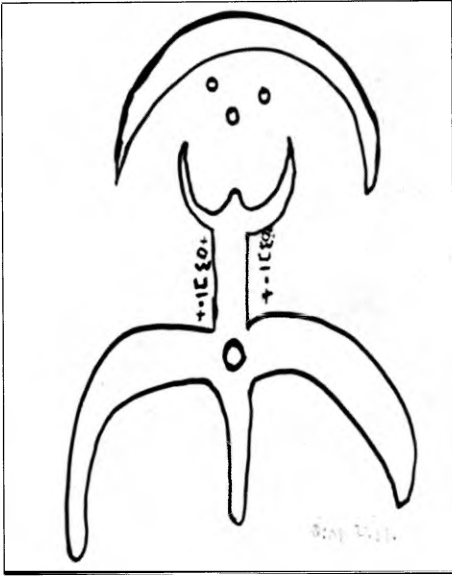
Planche 9

Anou Elias, Vallée de Mammanet (Niger)
 Henri Lhote. Les gravures de l'Oued Mammanet.
 Les Nouvelles Editions Africaines. 1979



Différentes polices ornées (tifinaghe-touareg).

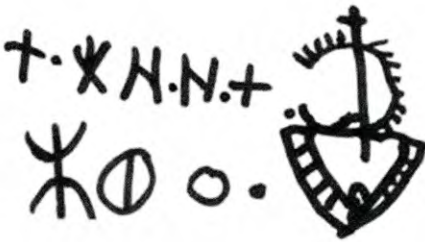
Planche 10



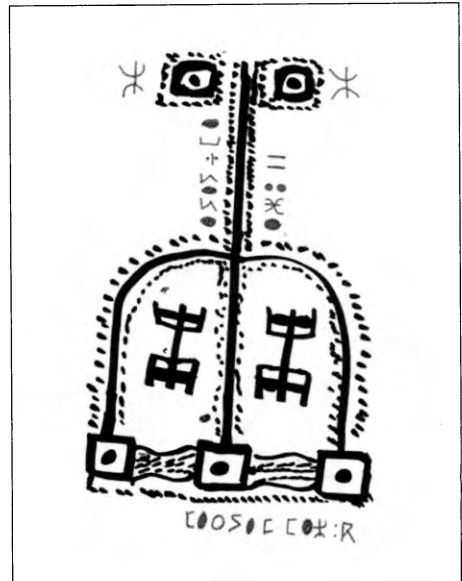
Ouafa Boudi, 12 ans (Figuig).



Abderrazzak Dounan, 12 ans (Figuig)



Zahra Ait Ouffi, 14 ans (Dadess).



Meryam Marzouki, 12 ans (Figuig)

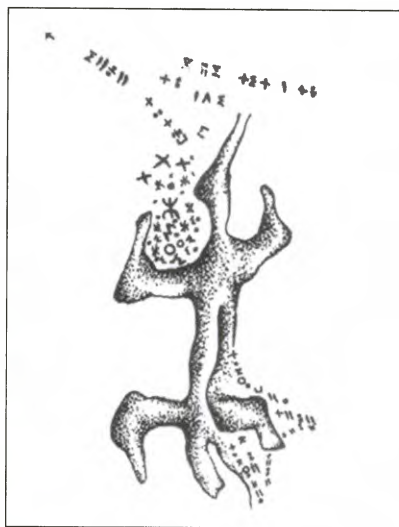
Planche 11



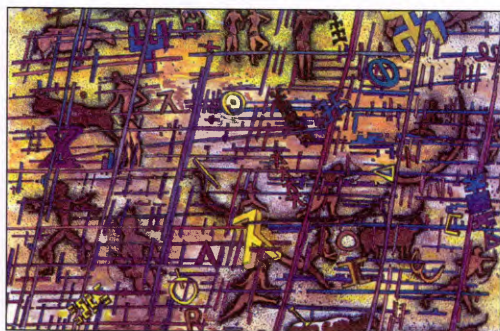
Zahra Ait Ouffi, 14 ans (Dadess).



Hawad. Poète et Calligraphe (Niger)



Adrar, peintre calligraphe (Libye)



Bourihan, peintre (Algérie)

Planche 12

CREATION TYPOGRAPHIQUE CONTEMPORAINE : LES GRECS DU ROI

Franck Jalleau

Graphiste concepteur à l'Imprimerie nationale

De la réhabilitation des caractères typographiques exclusifs en plomb, à la création de caractères pour l'édition, ou bien encore pour les documents fiduciaires, voici déjà une quinzaine d'années que je dessine des caractères pour l'Imprimerie nationale. Revisiter les formes de caractères depuis celles de Garamont jusqu'à celles de Gauthier, en passant par les formes de Grandjean ou de Luce, pour les adapter aux nouvelles technologies, est mon quotidien depuis toutes ces années. Il faut avouer que pour un concepteur de caractères, certaines situations peuvent être pires. Toutes ces heures à observer, à comparer, à dessiner, à étudier ces caractères resteront un grand privilège, mais surtout un enseignement exemplaire. Chaque étude est une leçon nouvelle d'où l'on sort un peu plus riche.

Revisiter les Grecs du roi, fleuron de la collection du cabinet des poinçons fut en 2002 ma nouvelle investigation. Chaque fois que j'avais en main un texte composé en Grec du Roi je m'abandonnais à une lecture peu commune de ses formes. Ne lisant pas le Grec, ni ancien ni moderne, ma lecture se réduisait à observer les caractères, la silhouette des mots ainsi créés, la danse ininterrompue de leurs accentuations. Cette danse abstraite que le rythme de cette écriture offre à la page, transformait ma lecture en un voyage visuel fascinant. Ignorant le sens des signes, mais armé de toute mon expérience de réhabilitation typographique, le pari me semblait réaliste.

Mon premier travail a été de constituer une bibliothèque de formes la plus exhaustive possible afin de me familiariser avec le système de ligature du grec ancien. Je me trouvais évidemment, à l'Imprimerie nationale, dans un contexte confortable, ayant à tous moments la possibilité d'observer l'écriture d'Ange Vergès, de regarder les poinçons originaux de Garamont ainsi que d'étudier les ouvrages composés en Grecs du roi pour François Ier. Cependant, l'objet de cette réhabilitation concernait bien la gravure de Claude Garamont. Il ne s'agissait pas de repartir de l'écriture d'Ange Vergès comme l'avait fait Garamont au XVI^e siècle. La démarche restait pourtant la même, puisque cette nouvelle version des Grecs ne serait constituée que d'un seul dessin découlant de l'observation formelle des trois corps gravés par Garamont. L'une des grandes particularités des Grecs du roi réside avant tout dans la multitude des ligatures qui les composent. Chacune de ces ligatures a bien entendu nécessité la gravure d'un poinçon, ceci vous donne une

idée de l'ampleur du travail. Disposant aujourd'hui d'un tout autre outil il me fallait effectuer un tableau de correspondance précis en vue de réduire au maximum le nombre de ligature, afin d'optimiser l'utilisation d'une police aussi complexe. Comme Garamont a pu le rencontrer tout au cours de l'adaptation de l'écriture d'Ange Vergès, il lui a fallu effectuer un certain nombre de concessions et d'interprétations des formes originelles, tout en conservant la substance même de l'écriture du calligraphe. Mon travail de dessinateur comme le sien, a consisté à observer l'ensemble des trois corps qu'il avait gravé afin de pouvoir restituer aux textes, composés avec ces caractères revisités, la même âme.

Commentaire d'un calligraphe

Pour quelqu'un qui pratique la calligraphie ou plus simplement l'écriture, quel spectacle que d'avoir entre les mains un manuscrit original d'Ange Vergèce! Le maniement de la plume d'oie, la réaction de l'encre sur le parchemin n'ont plus aucun secret pour ce scribe. Il nous offre une écriture qui coule et bouge si naturellement que l'on sent immédiatement cette symbiose entre le geste, la fente immédiatement cette symbiose entre le geste, la frellement que l'on sent immédiatement cette symbiose entre le geste, la forme et la pensée sans la moindre contrainte évidente. Avant d'apprécier la richesse de chacune de ces formes dans le détail, on observe la couleur générale du texte, ce juste équilibre des marges, la précision de l'interlignage, pour ensuite arriver à cette seconde lecture, où, l'œil se trouve surpris, envahi par ce foisonnement de signes, tantôt ligaturés ou simples, basculant vers la droite ou vers la gauche. Quelle surabondance d'accents et d'esprits, dessus et dessous, qui égayent la page! Remarquons aussi à quel point l'écriture d'Ange Vergèce est maîtrisée: sa régularité devient presque une écriture typographique. Cette écriture dévoile bien plus que la virtuosité d'un calligraphe: l'immense culture et la pensée intérieure de cet érudit. Cette écriture "savante" où l'invention est illustrée par l'ensemble des ligatures, révèle la vitesse d'exécution du calligraphe. Tout l'intérêt des Grecs du Roi réside dans cette multitude de signes créant un rythme mature et nerveux, où l'élégance arrive à son summum.

De la réhabilitation des caractères typographiques exclusifs en plomb, à la création de caractères pour l'édition, ou bien encore pour les documents fiduciaires, voici déjà une quinzaine d'années que je dessine des caractères pour l'Imprimerie Nationale. Revisiter les formes de caractères, depuis celles de Garamont jusqu'à celles de Gautier, en passant par les formes de Grandjean ou de Luce, pour les adapter aux nouvelles technologies, est mon quotidien depuis toutes ces années. Il faut avouer que pour un concepteur de caractères, certaines situations peuvent être pires. Toutes ces heures à observer, à comparer, à dessiner, à étudier ces caractères resteront un grand privilège, mais surtout un enseignement exemplaire. Chaque étude est une leçon nouvelle d'où l'on sort un peu plus riche.

pi epsilon rho iota

περι

Ligature à quatre signes

gamma epsilon nu

γεν

Ligature à trois signes

omikron nu

ο ν

Ligature à deux signes

Alpha, zeta, alpha et zeta écrits de gauche à droite



LES GOSCH DU ROI



Manuscrit d'Anglo-Saxons

LES OSCE DU ROI



Lesen del'Antenre «le Dantes»



Nominative

ganna rho am lora

ganna rho am lora

LES GRÈCS DU EUI



Deux à l'analyse

γ ρ ο
gamma rho omicron

LES GRÈCS DU EUI



Cursive définitive

γ ρ ο

LES GRÈCS DU EUI



Rythme: largeur des caractères

LES GRÈCS DU EUI



Rythme: poids des caractères



Revoir aux lettres et décorer de vos écrits



qu'il est écrit ou après écrit

ε̃ η α ι



qu'il est

ε̃

LES GREECS DU ROI

gamma epsilon nu

LES GREECS DU ROI

tau rho

tau rho omicron

τ ρ

τ ρ ο

LES GREECS DU ROI

epsilon pi epsilon iota

ε π ε ι

LES GREECS DU ROI

epsilon pi epsilon iota delta etc

ε π ε ι δ η

DESSINER UNE POLICE TIFINAGHE

Alexis Logié

En préambule, je distinguerai les deux plus grandes familles de caractères typographiques : les caractères de titrage, parfois appelés « fantaisie », utiles pour l'écriture de texte courts, logotypes, titres, enseignes ou slogans publicitaires ; et les caractères dits « de labeur », dont le dessin est conçu pour faciliter la lecture de textes longs, livres, articles de journaux, ou de magazines....

C'est sur cette dernière catégorie que mon exposé portera.

Pour qui s'attelle à la tâche de dessiner une police tiffinagh « de labeur », la première difficulté est l'absence de tradition calligraphique cursive-à ma connaissance. J'entends par là une écriture rapide occasionnant de nombreuses déformations des glyphes, selon le principe d'isochronie entre les tracés manuels des différents signes ; ainsi que la création de ligatures comme par exemple l'esperluette « & » ou « @ », le Lam'alif en arabe ou le « B » allemand.

Comme déjà signalé plus haut, un caractère de labeur doit faciliter la lecture de textes longs sans fatigue oculaire.

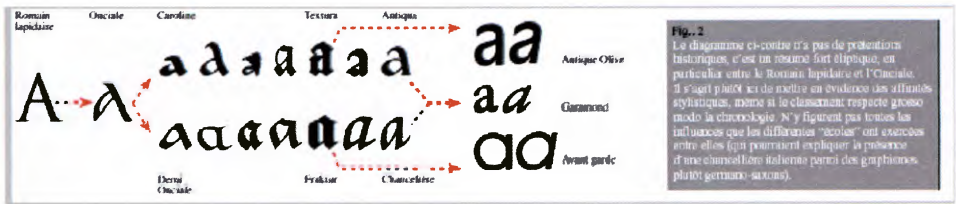
Son dessin doit donner une sensation de fluidité, l'œil du lecteur étant guidé d'un signe vers le suivant d'une manière harmonieuse sans heurt. La lecture ne doit pas être hachée pour que le lecteur ne perde pas le fil du texte. Dans l'idéal, la lecture est même plus rapide que la parole.

Pour ce faire le typographe aura recours à deux « techniques », ou deux procédés:

- 1 - S'appuyer sur le dessin calligraphique
- 2 - Appliquer à son dessin original des corrections optiques destinés à équilibrer la perception du glyphe (par exemple les creux pénétrants des lettres comme le N capital-voir Fig.1)



Les déformations calligraphiques que l'on constate entre les caractères « originels » et les écritures actuelles guident le travail du typographe, car elles sont toutes destinées à rendre la graphie plus rapide et plus fluide, selon des principes d'économies de mouvements de la main si l'on y associe des critères de lisibilité (1), qui caractérisent la « belle écriture », force est de constater que cette économie de mouvement de la main facilite le travail oculaire et cérébral associé à la lecture, ce qui tend à définir des critères ergonomiques de dessin des caractères. Je souligne d'ailleurs que ces considérations sont valables dans les traditions calligraphiques arabes autant qu'occidentales. Les copistes ont mis des siècles de pratique intensive à obtenir les déformations qui ont engendré nos écritures actuelles, en étant, de plus, influencés par le contexte culturel, religieux ou politique de leurs différents lieux et époques. Que l'on pense simplement à l'évolution phénoménale entre la forme des inscriptions lapidaires romaines et l'avènement de la minuscule carolingienne pour aboutir à nos caractères bas de casse contemporains (voir figure2).



Cette mutation a été spectaculairement accomplie par la confrontation d'influences graphiques et culturelles latines, germanes et celtes, aussi bien que par des décisions politiques brusques charlemagne imposant l'emploi de la Caroline dans les communications officielles, ou dans le cas de l'arabe, l'empire Ottoman diffusant le style « Diwani ».

Cette tradition, le typographe n'en dispose pas dans le cas du tiffinagh. Nous avons d'ailleurs coutume avec mes étudiants d'en plaisanter en ces termes outranciers : « le passage du caractère lapidaire à la typographie informatique a pris deux mille ans aux occidentaux, nous avons six mois pour faire la même chose avec le tiffinagh ». C'est, bien sur, outrageusement caricatural, mais cela résume assez bien l'inquiétude du typographe devant la tâche à accomplir.

Pour l'accomplir tout de même, le typographe doit puiser dans une pratique calligraphique personnelle (peut évidente dans mon cas, n'étant pas amazighophone et gaucher, de surcroît), pour trouver où placer ses courbes, ses pleins, ses déliés, le centre optique des lettres, et pourquoi pas créer des ligatures.

L'autre pratique sur laquelle le typographe peut s'appuyer est une tradition typographique choisie en référence.

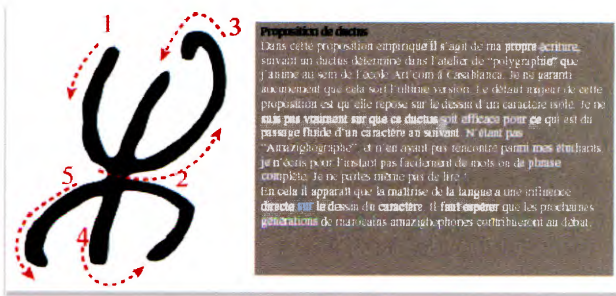
Certains signes tfinagh sont en effet proches de signes latins, grecs ou cyrilliques, ce qui laisse envisager la possible création de la version tfinagh d'une police de caractère occidentale quelconque.

Le danger est là de produire quelque contresens culturel en dessinant une sorte de tfinagh déguisé en Elzéviere, en Didot voire en Fraktur ; nous ne sommes à l'abri de rien !

Le risque est d'aboutir à une graphie « colonisée » dont le dessin serait tellement influencé par une forme occidentale (en ce qui me concerne) qu'il devient absurde de tenter d'exprimer un fond culturel spécifique à l'aide de ces signes.

Je suis là partisan du métissage, pas de l'abatardisation.

Il est néanmoins bien entendu que la création d'une police typo de labeur tfinagh doit s'appuyer sur l'expérience typographique des autres cultures, inutile d'inventer l'eau chaude tous les matins ! Mais elle doit le faire avec lucidité et clairvoyance, c'est-à-dire en retrouvant le « ductus » ⁽²⁾ sous tendant le dessin du signe typographique pour pouvoir s'appuyer dessus.



Une culture analphabète n'est pas muette pour autant

Ici je voudrais insister sur l'importance de comprendre, de recenser et d'intégrer les signes existants comme « vocabulaire graphique » de base du dessin typographique du tfinagh. Dans le cas du tfinagh nous disposons de motifs tissés, de différentes formes de gravure, en particulier sur métal (bijouterie) et des ornements peints sur la poterie, ou le travail du cuir, voire le tatouage et les motifs au henné. L'analyse de ce patrimoine graphique pourrait être d'une grande utilité en ce qui concerne l'établissement des proportions d'un caractère, et même inspirer des modes de composition de texte. Qu'il me soit permis de remarquer que cela implique une diffusion accessible de ces informations. Une publication en ligne de ce genre de fond pourrait grandement faciliter le travail.

Le dilemme italique

Dans un premier temps, j'étais instinctivement opposé à la création d'un tiffinagh italique. Ce me semblait une absurdité culturelle, l'italique étant né d'un environnement culturel spécifique ayant peu de choses en commun avec le contexte actuel (Alde Manuce, 1501).

D'autre part la décision de proposer les versions romaines et italiques d'un même caractère est un choix ultérieur, lui-même fonction de circonstances particulières.

Puis, à la réflexion, je me suis dit qu'il valait mieux que des graphistes s'occupent du design plutôt que de se faire infliger quelque mauvais traitement par des informaticiens, ou pire, des programmes. Après tout, les mutations typographiques occidentales ont été le fruit d'un contexte culturel, mais également le reflet des évolutions techniques de leur temps. En commençant par l'invention de Gutenberg, en passant par la mise au point de nouveaux alliages qui engendrent le dessin des didots, ou les répercussions de la photocomposition sur la typographie au xx e siècle.

La vulgarisation de la pratique typographique, avec la multiplication des ordinateurs et des logiciels de traitement de texte et de « création graphique » fait partie de notre contexte culturel et technologique. A ce titre, l'apparition de caractères tiffinagh « italisés » semble inéluctable. Nous savons déjà les conséquences que ces traitements informatiques ont eu dans le passé sur la typographie occidentale, et la tristesse infinie du typographe contemplant un titre en Bookman gras italisé élargi à 180%(tout capital). De plus, il me semble évident que le développement d'une presse tiffinagh induira une demande en caractères permettant une hiérarchisation de l'information comparable à ce qu'on observe dans la presse occidentale ou arabe, et il faut bien reconnaître que l'italique est souvent bien utile⁽³⁾.



Dans le cas de la création d'une fonte tiffinagh italique, la nécessité de fonder solidement le travail typographique sur la cursivité est une évidence. On ne saurait se contenter d'une inclinaison automatique du caractère (slant, comme disent les anglo-saxons) (fig.4).



Fig. 4
Des différences de dessin qui semblent imperceptibles (ici sur l'Antique Olive) jouent un grand rôle dans la fluidité et la lisibilité du texte composé.

Avant d'être une inclinaison de l'axe des lettres, l'italique doit plutôt être vue comme la « cursivisation » d'un caractère mécanique comme dans le Garamond, ou à rebours, comme la mécanisation d'une écriture cursive, le Bodoni en fournissant un bon exemple (fig.3). Comment appliquer les mutations graphiques que l'on observe sur un « a » elzévire entre le romain et l'italique à un signe comme le z tiffinagh ? (fig.5)



Fig.5
Essais d'italisation sur la base du caractère Aïta que j'ai dessiné. Sur la première ligne, le caractère est simplement incliné. Sur la deuxième ligne figure une proposition (non aboutie) tracée rapidement pour la circonstance d'un Aïta italique. Le dessin résulte de la confrontation entre la version romaine et le dessin de la page précédente.

S'agit-il d'ailleurs réellement de mutation de l'un vers l'autre ? A bien regarder les « a » Garamond et Bodoni de l'exemple, on s'aperçoit que les deux dessins n'ont en fait pas grand-chose en commun. Certes les deux versions sont « assorties », mais il pourrait tout aussi bien s'agir de deux polices de caractère différentes. Le fait d'assimiler l'italique à une version oblique du romain est dû à notre habitude de leur juxtaposition. En fait le concept d'italisation du caractère apparaît comme légèrement impropre. Au même titre que pourrait l'être la « romanisation » de caractères obliques. Pourquoi ne pas proposer de redresser les nombreux « chancery » du marché ? Rendons à Manuce la monnaie de sa pièce, il a penché nos caractères, redressons les siens !

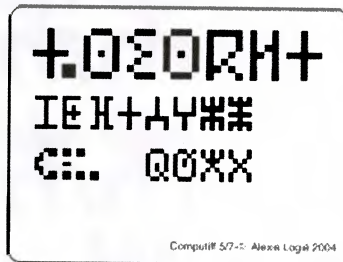
Trêve de plaisanterie et revenons à notre Tifinagh italique. Je pense que sans une recherche, une collecte et une analyse d'écritures cursives, la tâche est impossible, ou produira des résultats discutables, en ce qui concerne un hypothétique caractère de labour. Le développement de l'alphabétisation Amazigh fournit actuellement une formidable occasion de collecter de multiples écritures manuelles à des fins

d'études et il serait utile que l'IRCAM organise cette collecte de matière brute, et sa diffusion.

Travaux Pratiques

Je vous présente ici quelques travaux que j'ai réalisés personnellement. J'ai animé cette année un atelier de recherche typographique à l'école Art'Com de Casablanca, baptisé Atelier de Polygraphie. Le but de ce cours est de rechercher des solutions aux problèmes typographiques posés par l'édition de document multilingues-et multigraphes-en arabe, Tifinagh et Latin. Mon travail en a été influencé, bien entendu.

La première proposition, computiff 5/7, concerne la création d'un alphabet destiné aux afficheurs électroniques, types tableaux a diodes lumineuses. La contrainte est d'utiliser une grille de cinq pixels sur sept qui semble être la norme sur ce genre de machines. Cette contrainte explique la décision de boucher les poinçons des voyelles et du yac que vous voyez à la troisième ligne.



La proposition Latiffa pourrait aussi être classée dans la famille des computrices chère à Alessandrini. L'exercice est là de concevoir simultanément un graphisme commun aux trois graphies.



Malheureusement il apparaît des différences d'aspect dans le « gris typographiques » des textes composés. Ceci est dû à la hauteur d'œil qui n'est pas suffisamment assorti entre le latin et l'arabe, ainsi qu'à l'interlignage problématique du tifinagh. L'absence de montantes et descendantes me fait envisager que la ligne de base du tifinagh pourrait être plus haute, le dessin général étant plus allongé et dépassant sous la ligne.

Il est intéressant de noter que la première ambition de ce travail était une tentative de dessiner une version tiffinagh de l'Optima. Le résultat ne fut pas du tout convaincant et Imad a fait le choix d'adapter les techniques du dessin des incises parallèlement à une recherche de ductus spécifique au tiffinagh plutôt que de se contenter du démontage /remontage d'une typo latine. Le résultat, que vous voyez ci-contre fonctionne plutôt bien en composition, même si il faudrait sans doute une graisse plus importante pour les utilisations en petits corps.

On notera que le dessin propose des solutions aux problèmes d'approches différents du Aita. En particulier, le dessin du ya n'est pas envisagé comme un élément du dessin des autres voyelles, de manière modulaire. Au contraire il s'agit d'un dessin spécifique, qui ne peut être confondu avec les autres lettres rondes, comme le yas. J'ajouterai que dans une composition bilingue, ce caractère est très plaisant confronté au zapf Humanist.

D'autres travaux d'étudiants sont également intéressants et les activités de l'atelier rencontrent une adhésion insoupçonnée de la part des étudiants. J'espère que l'avenir confirmera cet intérêt des jeunes générations de graphistes.

Bibliographie

Calligraphie Latine-Initiation

Véronique Sabard, Vincent Geneslay, Laurent Rébéna Editions Fleurus

Science et Vie n°204

Septembre 1998

Hors série : le cerveau et le mouvement

Pleins et déliés-Paolo Vivani. Pages 36 à 47

Concernant l'utilisation de motifs traditionnels adaptés à la typographie voir sur le site de pierre di sciullo son travail sur l'Aligourane et l'Amanar, même s'il ne s'agit pas de l'alphabet Tifinagh IRCAM.

<http://www.quiresiste.com/>

QUELQUE ASPECTS DE LA NUMÉRISATION DES POLICES DE CARACTERES : CAS DE TIFINAGHE

Youssef AIT OUGUENGAY

Généralités sur La Typographie numérique :

La typographie numérique est un sujet très pointu qui présente deux aspects différents mais complémentaires : un aspect artistique et un aspect informatique.

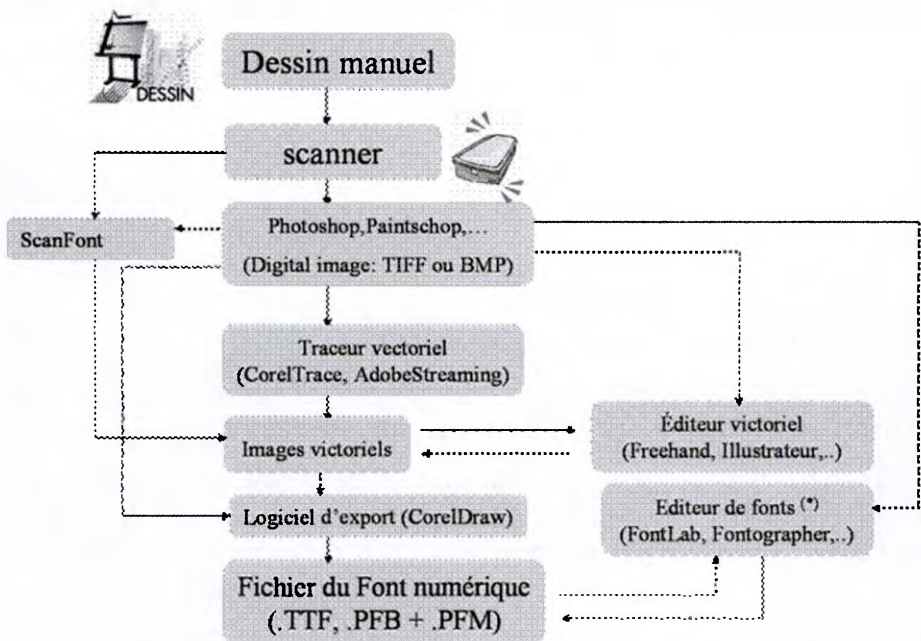
En fait, la définition que nous donnerons à la typographie n'est pas aussi général que son vrai sens : c'est-à-dire que nous ne nous intéressons qu'à la partie de la typographie numérique qui s'occupe de la création et du dessin des caractères.

Un créateur de polices de caractères utilisera soit des techniques de dessins traditionnelles (sur papier), soit des techniques de dessin sur ordinateur, soit encore un mélange des deux. Ceci nécessite plusieurs phases de travail en commençant par la création des caractères numérisés et en passant par la mise en forme jusqu'à l'optimisation et en fin la gestion des fontes créées.

Souvent les concepteurs de fontes (professionnels) créent leurs dessins manuellement, puis ils numérisent les contours du caractère point à point à l'aide d'un programme adéquat. Une autre façon de procéder est de passer le dessin du caractère à travers un numériseur pour créer un plan de bits, puis faire traiter ce plan de bits à l'aide d'un programme de tracé automatique de contour. Les contours résultants doivent être ensuite retouchés par un éditeur de contour (Fontlab, Fontographer,...). Cette dernière étape est obligatoire car l'extraction des contours ne produit qu'un rendu partiel.

Les typographes manipulateurs d'ordinateurs, eux, peuvent préférer de passer leurs fontes sur ordinateur dans toutes les étapes, ainsi ils peuvent utiliser un logiciel de traitement de dessins comme par exemple CorelDraw, Illustrator ou autres pour la création directe de leurs dessins numérisés, et ensuite Exporter ces dessins aux éditeurs de contours (Fontographer, FontLab) pour les opérations d'optimisation et les retouches finales avant la génération des fichiers fontes.

Le schéma suivant montre les différents parcours que peut prendre un typographe pour produire des polices de caractères.



(*) Possibilité de produire les fontes directement.

Bases mathématiques de la typographie numérique :

Cette partie explique les techniques actuelles de la représentation des courbes de polices de caractères.

DESCRIPTION DES COURBES (OUTLINES) :

Les langages de description des documents actuels comme PostScript ont besoin de descriptions de fontes qui doivent rester invariantes lors des transformations les plus raffinées. Pour cela, Les systèmes commerciaux de manipulation et d'optimisation de fontes décrit les formes des caractères en utilisant des fonctions mathématiques sous formes des « splines » cubiques ou quadratiques

Les splines cubiques à deux dimensions (Bézier, B-spline, etc.) sont décrit par des morceaux de courbes polynomiales paramétrés. Puisqu'ils sont capables de générer des formes de contours très détaillées, on les utilise pour décrire les différentes parties d'un caractère.

Les fonctions splines sont devenues un outil fondamental du graphisme sur ordinateur. On les retrouve dans tous les programmes de CAO et il est certainement bon pour un développeur de fontes d'en connaître l'existence et les principes.

Les splines dont il est question ici sont utilisées, la plupart du temps, pour le dessin de courbes régulières et esthétiques. Les deux pionniers de cette technique, Bézier et De Casteljau, travaillaient respectivement chez Renault et chez Citroën et avaient comme but le dessin des carrosseries de voiture.

Je ne prétends pas que les équations qui vont suivre explicitent entièrement la théorie mathématique qui est derrière les splines de Bézier. Je vise seulement à donner les éléments moteurs de la chose.

LES SPLINES DE BEZIER :

Une Courbe de Bézier est une courbe paramétrique qui permet, par construction itérée, de réaliser un **arc de courbe continu** d'extrémités imposées, et avec des **Points de Contrôle** qui n'appartiennent pas nécessairement à cette courbe mais en définissent les **tangentes**.

Les courbes de Bézier sont généralisées au degré K. mais les courbes les plus utilisées sont ceux de degré 3. Une courbe de Bézier de degré 3 comporte 4 points de contrôle (k -> k+1). L'équation paramétrique d'une courbe de Bézier est donnée dans la base des polynômes de Bernstein.

Par définition la courbe de Bézier associée à un ensemble de n + 1 points P_0, P_1, \dots, P_n , est l'ensemble de points $B_n(t)$:

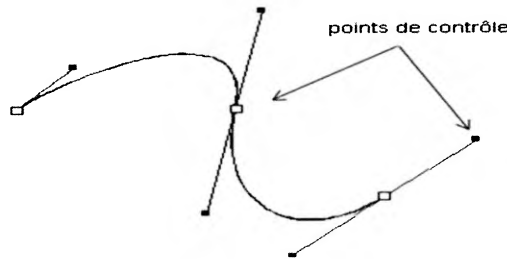
$$B_n(t) = \sum_{i=0}^n B_i^n(t)P_i \quad \text{Avec : } t \in [0,1]$$

$B_i^n(t)$ sont les polynômes poids de Bernstein.

Dans le cas d'une courbe de degré 3, la plus usuelle, on peut écrire explicitement la courbe à l'aide des polynômes de Bernstein de degré n = 3 :

$$x(t) = (1-t)^3x_0 + 3(1-t)^2tx_1 + 3(1-t)t^2x_2 + t^3x_3$$

$$y(t) = (1-t)^3y_0 + 3(1-t)^2ty_1 + 3(1-t)t^2y_2 + t^3y_3$$



Analytiquement parlant on peut construire itérativement et de façon graphique une courbe de Bézier par l'algorithme de **De Casteljau**.

ALGORITHME DE DE CASTELJAU (BEZIER DE DEGRE N):

Le principal intérêt de cet algorithme est Une construction géométrique très intuitive et par itération.

◆ Soient b_0, b_1, \dots, b_n des points du plan, $b_i = x_i, y_i$ et $t \in IR$

◆ Les points de la courbe de Bézier sont :

$$b_i^r(t) = (1-t)b_i^{r-1}(t) + t b_{i+1}^{r-1} \quad \text{Où} \quad b_i^0(t) = b_i$$

$$1 \leq r \leq n$$

$$0 \leq i \leq n-r$$

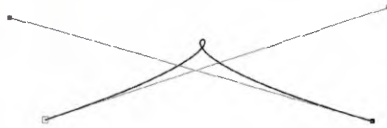
Alors $b_0^n(t)$ est le point de la courbe de Bézier de degré n correspondant à la valeur du paramètre t .

☞ Intérêt des courbes de Bézier :

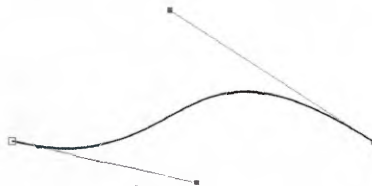
- ◆ Elles donnent une bonne idée de l'allure de la courbe.
- ◆ Une modification de la courbe s'obtient par déplacement des Points de contrôle.
- ◆ On peut concentrer plusieurs points de contrôle pour mettre en valeur une zone déterminée.
- ◆ Pour éviter d'obtenir un degré trop élevé, on peut raccorder deux ou plusieurs courbes de Bézier. On obtient une courbe composite (spline).

☞ Intérêt de degré 3 :

En plus des courbes d'une plus forte régularité, il permet de dessiner des **plis** (comme ceux de la cubique d'équation : $y = x^3 - 3x$, en $x = 1$ ou -1), ou des **points d'inflexion** (comme celui de la cubique d'équation : $y = x^3 - 3x$, en $x = 0$), ce que le degré 2, avec ses arcs de paraboles, ne sait pas faire !



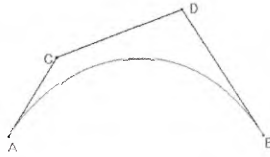
Point de plis



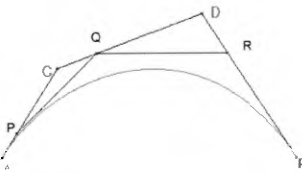
Point d'inflexion

Approche graphique :

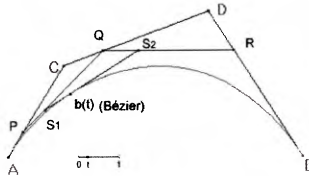
Graphiquement, on peut dessiner une courbe de Bézier sans calcul grâce à l'algorithme de Casteljau :



- on choisit un paramètre t : par exemple $t = \frac{1}{3}$ ($0 < t < 1$)
- on subdivise les segments du polygone ABCD chacune par trois, pour trouver les points PQR.



- on répète cette subdivision avec les segments PQ et QR pour donner le segment S_1S_2 :



- finalement on applique notre paramètre t au segment S_1S_2 ...on démontre mathématiquement que Le point $b(t)$ résultant appartient au courbe de Bézier!
- Pour construire toute la courbe de Bézier il suffit de retrouver le point $b(t)$ pour chaque t appartenant à l'intervalle $[0,1]$.

LES COURBES B-SPLINES (BASIS SPLINES) :

Comme les splines de Bézier, les courbes B-splines sont définies par des points de contrôle. De plus, les B-splines ont la propriété d'être locales: ce qui veut dire que le fait de modifier un des point de contrôle ne modifie la B-spline que sur quatre intervalles. On pourra donc modifier un segment de courbe, en déplaçant un point de contrôle sans influencer d'autres parties du dessin. Contrairement au courbe de Bézier qui est totalement modifiée dès qu'on déplace un point de contrôle : on dit que la méthode de Bézier est une *méthode globale*. La méthode B-spline est dite *méthode locale*.

Nous nous contenterons ici du cas le plus courant, celui des B-splines cubiques (ordre 3) uniformes, dans le cas le plus simple.

LES FONCTIONS DE BASE DES COURBES B-SPLINES :

Une B-spline « uniforme », peut être définie comme une suite de Courbes de Bézier accolées les unes aux autres de façon à ce que les raccordements entre elles soient suffisamment lisses aux points de raccord. Les courbes de Bézier qui composent la B-Spline sont définies chacune par un paramètre local t de temps décrivant toujours [0,1] ; si on décale simplement ce paramètre d'une courbe de Bézier à la suivante, on fait décrire alors à t tout l'intervalle [0, N] (N nombre de courbes de Bézier), ce qui évite la répétition du même t, et donc autorise une définition paramétrique globale de la B-Spline. Nous considérons donc n+1 points de contrôle $p_i = (x_i, y_i), i = 0, 1, 2, \dots, n$ Sur $(p_i, p_{i+1}), i = 1, 2, \dots, n-1$ l'intervalle. L'équation de la courbe B-spline d'ordre k est :

$$B_i(t) = \sum_{i=0}^n P_k b_{i,k}(t)$$

Chaque point de contrôle de la B-spline est associé à une fonction de base $b_{i,k}(t)$ définis par une relation récursive :

$$b_{i,k}(t) = \left[\frac{t-t_i}{t_{i+k}-t_i} \right] b_{i,k-1}(t) + \left[\frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} \right] b_{i+1,k-1}(t) \quad \text{avec : } k \geq 2$$

pour k=1 :

$$b_{i,1}(t) = \begin{cases} 1 & \text{si } t_i \leq t \leq t_{i+1} \\ 0 & \text{sinon} \end{cases}$$

$b_{i,k}$ est un polynôme de degré (k-1) sur chaque intervalle $t_i \leq t \leq t_{i+1}$. Et le vecteur $(t_0, t_1, \dots, t_{n+k})$ doit être spécifié.

Le paramètre t parcourt l'intervalle $[t_i, t_{i+1}]$. Les fonctions sont les tracés sur des fonctions de base associées aux points voisins de l'intervalle appelés aussi fonctions de pondération. est nul en dehors de l'intervalle $[k, k+n]$. Le graphique suivant montre la fonction de basée associée à un point.

Quelques propriétés des courbes B-splines sont :

- ◆ Composés de (n-k+2) courbes de Bézier d'ordre k, de classe C^{k-2} aux nœuds .
- ◆ Chaque point qui appartient à la courbe est affecté par k point de contrôle.
- ◆ Chaque point de contrôle affecte k intervalle $[t_i, t_{i+1}]$.

LES COURBES B-SPLINE QUADRATIQUE (N=3):

Se sont les courbes utilisées pour le dessin des polices de caractères TrueType.

A titre indicatif, les fonctions de bases des courbes B-spline quadratiques sont :

$$\begin{aligned} b_{0,3} &= t^3 / 6 && \text{pour } 0 \leq t \leq 1 \\ b_{1,3} &= (-3t^3 + 12t^2 - 12t + 4) / 6 && \text{pour } 1 \leq t \leq 2 \\ b_{2,3} &= (3t^3 - 24t^2 + 60t - 44) / 6 && \text{pour } 2 \leq t \leq 3 \\ b_{3,3} &= (4-t)^3 / 6 && \text{pour } 3 \leq t \leq 4 \end{aligned}$$

Ils ont plusieurs Propriétés avantageux :

- Une B-spline quadratique passe par le point milieu.
- en ce point, la tangente à la spline est confondue avec le segment défini par ces deux points de contrôle.
- la spline est située dans le triangle défini par un point de contrôle et les deux points milieu environnants.
- Adaptée au codage du contour des caractères à haute définition.

L'optimisation du rendu des polices de caractères :

Le problème de la typographie numérique a toujours été posé au niveau de l'affichage sur supports à faible résolution comme les écran d'ordinateurs et les imprimantes (600 dpi ou moins)...Le typographe est forcé de mener toujours une guerre sans cesse contre les mauvaises interprétations des contours par l'interpréteur des contours des oeils (glyphes).

LE HINTING :

Techniquement le hinting est l'ensemble d'informations introduite dans un fichier de police de caractères. L'interpréteur (rasterizer pour les anglophones) utilise ces informations, en plus de la forme conçue des glyphes, pour faire des décisions sur les pixels à allumer ou éteindre lors de l'affichage des caractères sur écran. Et artistiquement le hinting est l'habilité de faire apparaître la forme d'un glyphe sur la surface d'affichage aussi mieux que possible.

Le hinting est donc un mélange de techniques et de l'art. Le besoin de hinting se résume dans le fait que quand un glyphe est affiché à une petite taille et sur un écran à faible résolution, alors on aura peu de pixels disponibles pour dessiner le glyphe et donc des détails clés comme les largeurs des bars, les hauteurs, etc. Deviennent

alors irréguliers voir même ils disparaissent, ce qui rend indispensable le travail de correction des irrégularités.

Le hinting est une étape essentielle dans la production des fontes de qualité. Il est indispensable dans toute fonte destinée à être lisible dans des petites tailles sur des écrans à faible résolution.

TRUETYPE : UN LANGAGE DE HINTING

Le TrueType, connu comme un format de polices de caractères, est en réalité un langage de hinting et de description de caractères qui mit à disposition un espèce de travail (*framework*) avec des fonctionnalités généraux pour la définition des courbes de la polices de caractères et les règles d'adaptation de la matrice d'affichage (grid-fitting pour les anglophones) aux différents glyphes. L'interpréteur du TrueType doit appliquer les règles de « grid-fitting » associées à la description du glyphe en déformant et adaptant ses courbes à la matrice d'affichage. Après le grid-fitting, l'interpréteur TrueType s'occupe du remplissage des glyphes proprement dit.

En plus de ça l'interpréteur est capable de détecter et corriger les erreurs d'arrondissement qui arrivent quand une courbes ou une dimension quelconque du glyphe a une largeur de moins d'un pixel. Sans ces corrections les glyphes paraissent sur l'écran troués et déformés.

Les fontes décrites par le langage TrueType donnent des informations sur :

- ◆ Les métriques.
- ◆ Les points de références.
- ◆ Les lignes de contrôle verticales.
- ◆ Les valeurs critiques (Snapping values).

La description des courbes est construite par des points donnés qui définissent les contours B-splines quadratique et les morceaux de segments linéaires. En fait, Les extrémités d'un segment spline appartiennent à la courbe parent, les points de contrôle intermédiaires définissent le comportement du spline et les points de la courbe proprement dits sont numérotés selon le sens d'orientation du contour.

Le langage de description des « hints » est un langage binaire codé qui incorpore des instructions pour initialiser des pointeurs sur les éléments constructeurs du glyphe, pour :

- ◆ Calculer les distances entre les points du glyphe pointés.
- ◆ Appliquer les opérations mathématiques sur des valeurs enregistrées auparavant dans la mémoire cache.
- ◆ Changer ensuite les places des points individuels ou série de points du glyphe selon les résultats trouvés à partir de ces opérations mathématiques.

Les points supportant les courbes du glyphe peuvent être déplacés verticalement, horizontalement, ou dans n'importe quelle direction donnée par le vecteur de liberté courant. En plus, le langage inclus des instructions pour :

- ◆ la mesure des valeurs de projection d'un vecteur.
- ◆ l'ajustement des angles d'inclinaisons pour les diagonaux.
- ◆ Des sauts vers les valeurs stockées dans la table des valeurs de contrôle (CVT) pour une taille donnée ou une rangée de taille et ceci uniquement pour certains instructions spéciales.

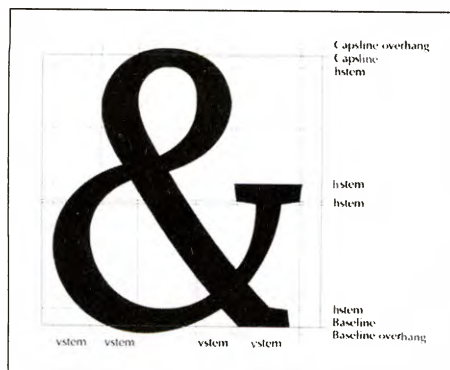
Le TrueType est un langage de hinting relativement compréhensible, mais ne donne aucune solution déjà défini et général pour le hinting des glyphes. Les producteur d'une fonte doivent eux même développer leurs stratégie de hinting. Ils peuvent aussi bien appliquer leurs hints manuellement sur chaque fonte, ou ressortir leur stratégie en produisant un outil qui automatise la production des hints adéquats.

LA PHILOSOPHIE DE HINTING POSTSCRIPT TYPE 1 :

L'approche Adobe du hinting est un peu différente des stratégies décrites auparavant. Les techniques du « grid-fitting » décrites dans la section précédentes pour le TrueType sont basées sur la déformation et l'adaptation des courbes de glyphe à la matrice de pixels. Quant au cas du format Postscript Type 1 les techniques appliquées sont basées sur l'optimisation de la distribution des lignes et colonnes de la matrice des pixels contenus dans l'espace remplis par les courbes du glyphe.

En terminologie Type 1, les *hints* ne sont que des spécifications des bandes horizontales et verticales limitant l'espace du glyphe. Ces bandes sont données par exemple sous formes de paires de lignes de référence (comme les paires « ligne de base, ligne de base limite», «ligne maj., ligne maj. Limite», etc) d'autres paramètres définissent la taille et la profondeur d'arc dont la correction optique affecte l'aspect visuel du caractère.

Les instructions individuels nécessaires pour produire des lignes réguliers et des contours bien courbés sont spécifiés par une paire de coordonnées (x) pour les largeurs horizontales (noté *hstem* en anglais) et une autre paires de coordonnées (y) pour les largeurs verticales (*vstem*). Pour un réglage d'une série d'instructions donné, les bandes définies par les paires de coordonnées horizontales et verticales ne doivent pas se coïncider.



L'interpréteur Type 1 est capable d'optimiser, pour chaque caractère, la distribution des lignes et colonnes de pixels par bandes spécifiées par les lignes de référence de la fonte et des instructions individuels.

Si un caractère donnée nécessite des instructions qui se coïncident, alors un premier ensemble d'instructions est décrit et appliqués au caractère. Ensuite, un nouvel ensemble d'instructions peut être utilisé pour améliorer la forme apparente du caractère en utilisant des bandes surchargeant ceux décrit par le premier ensemble. Cette philosophie de hinting n'est pas aussi flexible que celle qu'offre le langage du hinting TrueType. Les « hints » ne contient que des informations sur les rangées verticales et horizontales fixant l'espace de remplissage du glyphe. Et c'est à l'interpréteur d'utiliser cette information d'une façon intelligente et d'optimiser la distribution des lignes et colonnes de pixels tout au long de l'espace occupé par le glyphe.

Grâce à ces capacités de décision et d'analyse, l'interpréteur Type 1 va créer encore plus d'opérations qu'un interpréteur TrueType qui doit optimiser le glyphe selon des règles prédéfinis. Mais dans le premier cas le rendu reste toujours à la merci du l'interpréteur différemment au deuxième cas qui se veut comme un programme informatique écrit par un développeur.

LES PARAMETRES DU HINTING :

La qualité typographique d'une police de caractères est contrôlée à travers plusieurs paramètres de hinting dont les principaux sont les suivants :

ALIGNEMENT :

Un attribut très important à considérer lors de hinting est le contrôle des différents alignements, à l'intérieur de la font et entre les différentes fonts de la famille. Généralement, les éléments similaires en taille et hauteur doivent être affichés de la même façon dans les petites tailles, car la différence d'un ou de deux pixels à des petites tailles est plus notable que dans le cas des grandes corps. En pratique ceci se fait en forçant les majuscules à avoir la même hauteur, et faire la

même chose avec les minuscules. Ces contrôles se font au niveau de la ligne descendante (marquant le point le plus bas dans la font) et au niveau de la ligne ascendante (marquant le point le plus haut).

Les capitales, par exemple, doivent être proprement alignés dans un mot, phrase ou texte, et plus encore, ces mêmes capitales doivent être alignées aussi avec leurs différentes variantes à l'intérieur de la famille. Il est donc intéressant de s'assurer, par exemple, que les lettres d'une phrase, écrite en régulier, s'aligne bien entre elles et aussi avec un mot ou une phrase écrite en italique ou en gras.

LA LISIBILITE :

C'est aussi un attribut critique d'une fonte bien faite. Chaque glyphe doit être identifiable comme un caractère particulier, sinon aux faibles résolutions ou à des petites tailles d'affichage il peut être difficile de représenter et de différencier les caractères. Par exemple : le «k» à une taille de 4 points est pratiquement illisible sur écran :



6 points



4 points



9 points

Dans cet exemple les lettres sont saisies à partir du MS Word puis agrandis de 300 %.

L'ESPACEMENT :

Un autre contributeur à la faible lisibilité est l'inconsistance de l'espace inter caractères. Cette irrégularité d'espacement est en général, le résultat de deux choses:

- Le dépassement, vers le haut ou vers le bas, de la largeur de l'image bitmap de caractère par rapport à son contour.
- La différence entre la largeur de l'image bitmap en Pixels et la largeur du caractère lui même dans un affichage de haute résolution.

L'espace entre caractères d'une part et entre mots d'autre part doit apparaître régulier pour l'œil. Il faut aussi être sûr que les caractères ne se chevauchent pas dans les petites tailles. Ceci peut se résoudre une tâche très difficile au cours de laquelle il faut faire des choix au dépit du rendu ! Prenant par exemple la cas d'un « n » (yan) : c'est un caractère qui occupe 4 pixels d'espace en largeur mais qui ne fait qu'un pixel en lui-même. Est-ce qu'il faut le mettre au milieu d'espace ou l'aligner à droite ou à gauche ? Ceci contribue dans la lisibilité de la fonte et

nécessite une décision. Les instructions du hinting TrueType donnent plusieurs issues avec ce genre de problèmes.

LE CONTROLE DES DIMENSIONS :

Dans une fonte bien « hintée » les largeurs et autres dimensions sont contrôlées, ceci est important aussi bien à l'intérieur de la fonte en question qu'aux relations entre les autres fontes de même famille. Par exemple, à une même taille la fonte gras doit être plus large que celle régulière. Plusieurs considérations doivent être prises en compte pour une bonne qualité de hinting :

- ▶ A quelles tailles les largeurs des caractères passent à deux, trois, quatre ou cinq pixels ? est ce que ceci est bien pour le rendu de la fonte ? est ce que ça doit rester encore fixe jusqu'à autre taille ?
- ▶ Comment la fonte régulière se différencie de celle gras ou italique ? ça doit être possible de contrôler les valeurs des largeurs pour produire une valeur différente à chaque taille.
- ▶ Qui saute en largeur le premier : les majuscules ou les minuscules ? quel est le cas le plus souhaitable ? (ce problème ne se pose pas pour le Tifinaghe)
- ▶ Est-ce que c'est les caractères arrondis qui sautent en largeur avant les horizontaux, les verticaux et les diagonaux ou vice versa ?

Chaqu'une de ces considérations est décisive dans le rendu de la fonte sur écran. S'il n'y a pas de différence entre deux largeurs d'une seule fonte sur écran alors il sera difficile de faire la différence entre le titre et le texte. Et si les caractères arrondis saute en largeur très vite, alors ça va affecter en même temps le niveau de gris du texte et l'espacement.

LA COULEUR D'APPARENCE (NIVEAUX DE GRIS):

La couleur est un élément critique dans l'aspect d'un texte destinée à l'affichage, ça n'affecte pas uniquement le cas des écrans à faible résolution mais aussi ceux à haute résolution et aussi les textes imprimés. En fait, la couleur d'un texte qui n'est pas régulière à travers les lettres et les lignes rend la lecture désagréable, c'est pourquoi il faut une couleur uniforme.

En typographie le mot couleur référence les « niveaux de gris » existantes entre le blanc et le noir. La noircisse des glyphes et la blancheur des arrière plans se combine pour produire la couleur du texte. Qu'elle soit grasse ou éclairée, l'essentiel c'est que la police de caractère ait un niveau de gris uniforme.

Plusieurs facteurs influence sur la couleur d'une fonte, par exemple :

- ▶ Le degré du Différence entre les grandes et petites largeurs à l'intérieur des glyphes.

- Les tailles des espaces internes des glyphes
- **Les Espaces inter caractères et interlignes.**
- L'irrégularité des pontes et des largeurs des diagonaux.

LA SYMETRIE :

La symétrie existante dans chaque glyphe doit aussi être contrôlé à travers la fonte. Malgré les subtilités souvent présentes dans les contours de caractères grâce aux bars diagonaux, une méthode efficace pour garder la symétrie des diagonaux est de les traiter symétriquement dans les petites tailles ; car Le fait de travailler la symétrie avec un petit nombre de pixels est plus simple qu'autrement.

Les signes esthétiques locaux :

Les catégories discutées ci-dessus affectent la fonte en totalité. Il est important de considérer aussi les particularités de la forme bitmap de chaque glyphe. Le hinting est très utile dans ce sens qu'il aide à garantir que les bitmaps ne sont pas déformé ou peu lisible et que les formes arrondis et les contours esthétiques ne perdent pas leur style.

Les types de polices de caractères :

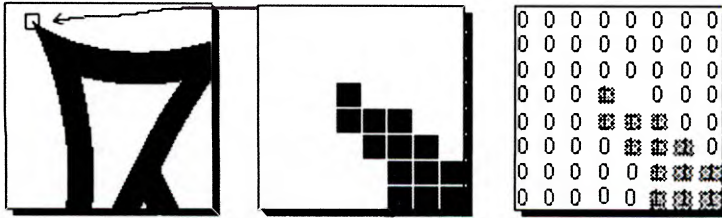
Une police de caractère est en général un ensemble de caractères avec un dessin bien précis (glyphes) et un code informatique. Ces caractères comprennent en général des caractères alphabétiques, des chiffres et des signes de ponctuation, ainsi que différentes informations, notamment du hinting pour gérer par exemple l'espace entre caractères... Une police peut se décliner en plusieurs variantes ou styles : les plus habituels sont le gras (bold), l'italique (italic) et le gras-italique (bold-italic). La police de base est dite normale (regular). Mais on trouve aussi des extra-gras (extra-bold), des maigres (thin), des étroitisés (condensed), des éclairés (light), des éclairés italiques (light Italic), etc...

Sur le plan informatique, une police est un programme, contenu dans un ou plusieurs fichiers. Chaque style est contenu dans un ou des fichiers distinct(s). pour créer des polices de caractères, il existe deux modes : le mode bitmap et le mode vectoriel. Le premier mode ne permet l'utilisation de la police qu'à des corps bien déterminés. Le second mode permet au contraire l'emploi de pratiquement n'importe quel corps avec un résultat très satisfaisant.

Les polices de caractères fabriquées en mode bitmap sont dites des polices Bitmap. Pour le mode vectoriel il existe plusieurs formats qui sont généralement les plus utilisés.

LES POLICES BITMAP :

Une police bitmap est un ensemble de caractères décrits par des matrices binaires, des **tableaux de points** noirs ou blancs.



Plus les caractères sont grands, plus il faut de points pour les décrire, et plus la taille du fichier de la police est importante.

Pour ce type de police, on a besoin d'autant de fichiers de description de police que de corps, de genre, de graisse,... et ces attributs se combinent ! Le Times italique gras de corps 12 est différent du Times romain (droit) gras de corps 12... Et, de mieux en mieux, il peut être nécessaire de définir des polices différentes selon l'orientation de la feuille (portrait ou paysage). Les polices bitmap consomment donc beaucoup de place, et il convient d'en tenir compte en exploitation courante.

Une autre particularité des polices bitmap est qu'il est difficile de les utiliser dans des tailles autres que celles disponibles. Dès que l'on change la taille d'un caractère (en particulier si on tente de l'agrandir), il apparaît un "effet d'escalier" fort désagréable : les traits inclinés sont formés d'une série de "marches". En effet, lors de l'agrandissement, le logiciel se contente d'augmenter la taille des points constituant le caractère, en les remplaçant par des petits carrés, en gardant les proportionnalités entre les parties du glyphe sans y ajouter aucune correction intelligente, ainsi que le montre le dessin suivant...



LES POLICES DE CARACTERES VECTORIELLES :

POSTSCRIPT TYPE 1

Le langage PostScript est un langage complet de description de page ou encore de composition graphique, et pas seulement de dessin de caractères. Il est conçu au départ pour décrire précisément un produit *imprimé* final, mais il est possible de le visualiser à l'écran. Le langage PostScript est d'une très grande richesse, permettant

de dessiner des choses très élaborées avec des instructions très abstraites. Il a été créé dans les années 1980 par Adobe et s'est imposé vers 1985 comme standard sous l'impulsion du duo Apple-Adobe et des imprimantes LaserWriter.

Ce langage comprend une partie dédiée à la description des polices vectorielles. C'est seulement cette partie « polices » du langage qui intéresse les typographes. Le langage Postscript utilise les courbes de Béziérs comme base mathématique pour la description des caractères. Les polices conçues avec ce langage sont naturellement appelées polices *PostScript*.

On notera que l'affichage de ces caractères sur un écran nécessite des ressources informatiques légèrement plus importantes que les polices TrueType (voir plus loin), et que les programmes permettant cela ne sont pas intégrés par défaut dans les versions actuelles de Windows et MacOS. Par contre, les imprimantes Postscript sont très répandues (en général toutes les lasers sauf les lasers bas de gamme comprennent le PostScript).

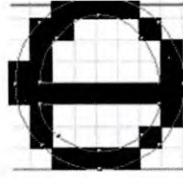
Les polices Type 1 se composent de deux types de fichiers:

- Les fichiers qui sont utilisés par le système pour l'affichage sur l'écran (extension *PFB*).
- les fichiers utilisés par le système d'exploitation pour le lissage à l'écran (grâce à l'utilitaire ATM) et à l'impression (extension PFM). Sur PC, c'est la même chose que sur Mac OS.

LE FORMAT TRUETYPE :

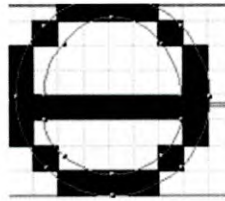
Les polices vectorielles conçues selon la norme TrueType sont principalement utilisées pour l'affichage sur écran dans les systèmes Windows et MacOS. Cette norme est issue d'un travail d'Apple et ensuite de Microsoft qui a très vite adhéré à TrueType pour s'affranchir de la dépendance de la technique équivalente de PostScript d'Adobe. Cette norme a apparue sur MacOS en 1991. Ce sont ces types de polices que l'on rencontre le plus couramment dans les environnements MacOS et Windows, car les programmes permettant leur affichage sont directement intégrés dans ces systèmes.

Les polices TrueType possèdent une particularité par rapport aux polices Type 1 en ce sens qu'elles ne sont composées que d'un seul fichier. Tout l'intérêt des TrueType réside dans leur faculté d'adaptation à la taille du caractère. Lorsqu'un caractère doit être affiché ou pixellisé dans une petite taille voici comment les contours occupent réellement la grille de pixels de l'affichage.



« Caractère sans hinting »

La forme de la police n'est pas respectée en raison de la faible résolution de l'affichage dans des corps aussi petits. Pour éviter ces mauvaises adaptations à de petites tailles, les polices TrueType comportent des instructions (hints) qui activent ou désactivent certains pixels pour que le contour de la police soit le plus fidèlement reproduit sur l'écran ou lors d'une pixellisation.



« Caractère avec hinting »

Encore faut-il que les typographes ajoutent ces instructions, d'où la complexité de la tâche. Seuls Adobe, Agfa, Monotype, Bitstream, etc., proposent la plupart de leurs polices dans les deux formats type 1 et TrueType. Ils existent Des logiciels de conversions, mais ils ne sont pas tous performants : Pendant la conversion, les indications de polices Type 1 peuvent être dégradées en passant au format TrueType et vis versa.

TYPE 1 VS TRUE TYPE :

Plusieurs différences techniques marquent les rapports entre les deux formats TrueType (TT) et PostScript Type 1 (PS1):

La première différence entre TT et PS1 est le fait qu'ils utilisent deux sortes de mathématiques différentes pour décrire leurs courbes : formats quadratiques B-Splines pour le format TrueType, et les courbes de Bézier pour PostScript Type 1.

La conversion entre les deux types est imparfait même si mathématiquement parlant les courbes de Bézier sont des splines quadratiques (B-Splines) particuliers, il y a toujours des erreurs d'arrondissement dans les conversions des deux sens TT -> PS1 et PS1 —> TT. Le sens PS1 —> TT fait perdre plus d'information que vice versa et l'information de hinting ne se translate pas de la même manière dans les deux directions.

Un avantage de TT sur le ps1 est que le TT permet un hinting meilleur :

- ▶ PS1 : peut avoir des « hints » :
 - verticalement et horizontalement.
 - liens entre points.
 - interpolation.

Plusieurs de ses hints peuvent avoir une largeur en pixels (Fût) à laquelle ils peuvent être liée.

- ▶ TT : fait tous ce qui précède plus d'autres instructions notamment :
 - Le contrôle des diagonaux.
 - Le déplacement aléatoire des points des contours des glyphes à des tailles spécifiques.

Cette habilité à changer la place des points permet un traitement rapproché de l'image bitmap produite par le contour du glyphe à n'importe quelle taille spécifiée pendant la production de la fonte.

Ces différences sont les résultats d'une différence plus large entre la façon d'interpréter les deux formats : PostScript utilise des fichiers fontes moins riches et un interpréteur plus intelligent, alors que TrueType traite des fichiers fontes riches en informations et un interpréteur moins spécialisé.

En fait, cette différence dans la façon d'interprétation des hints est stratégique : les hints de PostScript Type 1 montre à l'interpréteur quelles parties de glyphe doivent être contrôlées, et puis le contrôleur interprète tous ceci par ça propre intelligence. Quant au TTF, des « instructions » (commandes machine) très spécifiques sont mises dans la fonte pour contrôler comment les glyphes vont apparaître. Ces « instructions » sont exécutées selon les conditions d'affichage de la fonte. Le producteur de fonte a un contrôle potentiel sur ce qui va se passer quand la fonte est manipulé par l'interpréteur dans différentes conditions. Mais le hinting de TT demande un sérieux effort pour prendre avantage de ces multiples possibilités.

Un autre facteur de différence est le fait que le processus d'interprétation du TrueType est intégré d'office dans plusieurs systèmes d'exploitation : Les deux systèmes MAC OS et Windows supporte TrueType. Ces systèmes vont interpréter leurs fontes pour l'affichage sur écran, et aussi pour la gestion de leurs envois vers l'imprimante sous formes d'images bitmaps ou des formats compréhensible par l'imprimante.

Des différences dans la façon de stockage physique existe aussi entre le TT et le PS1 : Les fontes TrueType ont l'avantage d'être stockés sous forme d'un seul fichier de données, les fontes Postscript sont sous formes de deux fichiers séparés : un contient les courbes des glyphes et l'autre contient les données métriques du hinting

(largeurs des caractères, etc.). Dans Windows les extensions de deux fichiers sont successivement «*.PFB*» et «*.PFM*». Sur Unix les deux fichiers ont respectivement les extensions «*.pfb*» (printer font binary) ou «*.pfa*» (printer font ascii) et pour le fichiers des métriques «*.afm*».

LE FORMAT OPEN TYPE :

Le format open type est comme un « conteneur » pouvant contenir les deux types de technologies Postscript et TrueType. Les applications et les systèmes d'exploitations n'aurons plus à se préoccuper de quel type de courbes y est !

Du point de vu utilisateur, le format open type est unique aussi bien pour le MAC que pour PC ou n'importe quelle autre plate-forme. Mais vu de l'intérieur le format Opentype a deux formes possibles : une basée sur le TrueType et l'autre sur le Postscript Type1.

La structure général du fichier de fonte est la même pour les deux versions et ils ont les mêmes fonctionnalités à part quelques différences techniques résumés dans ce tableau :

version	Associé à TrueType	Associé à Postscript Type 1
courbes	2éme ordre, comme les polices de caractères TrueType	3éme ordre, comme les polices de caractères Type 1.
hinting	Instructions TrueType	Hints déclaratifs Type 1
Extension de fichier	<i>.TTF</i>	<i>.OTF</i>
Commentaire	Basiquement toute fonte TrueType peut être appelée open type « TT-flavoured », mais on l'appel usuellement « fonte open type » quand elle contienne des fonctionnalités spéciales.	Les données courbes sont enregistrées sous un format CFF (compact font format) qui est supposé être plus adéquat aux systèmes Postscript (Mac OS, imprimante postscript,...)

L'OpenType apparaît comme une solution adéquate pour la guerre de fontes menée par les grands constructeurs (Microsoft & Apple), mais il faut d'abord que ce format soit supportée et intégré au niveau des systèmes d'exploitation, et les applications nécessitent d'être retravaillés pour prendre avantage des fonctionnalités avancées de ce format.

La numérisation de Tifinaghe : un défi à relever

« Les inscriptions amazighes trouvées concernent l'origine de l'écriture au Maroc, et plus largement en Afrique du Nord. Plusieurs questions restent posées

concernant l'apparition, les usages, les supports et la longue disparition au Maroc de cette écriture.»⁽¹⁰⁵⁾

USAGES DE L'ECRITURE :

« ..D'après les documents qui ont été partiellement déchiffrés, il semble que les usages de l'écriture amazighe se limiteraient à des dédicaces, des nécrologies et à la consignation de courts messages. [...] peu d'essais de déchiffrement d'inscriptions rupestres marocaines ont été faits; il est donc trop tôt pour savoir le contenu des inscriptions rupestres du Maroc. Seul un travail de déchiffrement de longue haleine (collecte des inscriptions, analyse pluridisciplinaire, étude comparative entre différentes régions de la zone historico-culturelle, etc.) permettra, à l'avenir, de restituer quelques contenus. »⁽¹⁰⁵⁾

SUPPORTS DE L'ECRITURE :

« La quasi-totalité des inscriptions a été exécutée sur un support rocheux (souvent du grès) qu'il s'agisse de gravures ou de peintures. Certaines inscriptions ont été également exécutées sur des vases ou des amphores. Il est possible que des supports périssables comme le bois, le cuir ou le textile, aient été utilisés, mais ils ne nous sont pas parvenus. »⁽¹⁰⁵⁾

PARTICULARITES DE L'ECRITURE TIFINAGHE :

UNE GRAPHIE HISTORIQUE :

La graphie Tifinaghe, différemment à la graphie latine, est pourvue d'une vraie évolution typographique. En effet le dessin des lettres Tifinaghes s'est différencié en fonction du son support d'écriture utilisé, De sorte que son écriture est généralement monumentale et lapidaire, on l'avait utilisé pour graver sur la pierre des formes rectilignes et anguleuses. Certes, cette écriture se distingue nettement des rondeurs de l'écriture arabe des copieurs maniant le calame par exemple, Mais, malheureusement elle n'a pas reçu les soins nécessaires pour passer au largeur du temps d'une technologie d'impression à une autre...

Les caractères de la graphie latine, quant a eux, ont été « copiés » d'une technologie d'écriture à l'autre successivement : ainsi les poinçons de plomb de Gutenberg ont commencer par copier l'écriture manuscrite, ensuite la technique du photocomposition n'a fait que copier les caractères de plomb et les premières fontes numérisées n'ont été que des copies de ce qu'existait. Toutefois, Il faut toujours un certain temps pour voir des caractères adaptés à la nouvelle technologie.

(105) - « Terra, aux origines de l'écritures au Maroc »

Le Tifinaghe n'as pas eu la chance de se développer étape par étape en ayant le temps suffisant pour nous transmettre sa véritable histoire (au moins c'est ce qui apparaît). Ceci mis en lumière à nos jours des soucis typographiques majeures.

En effet, Quel typographie pour le Tifinaghe ? Est-il possible de réduire des siècles d'existence de cette graphie en une génération en copiant directement les formes de glyphes depuis les inscriptions rupestres ? Certes, on peut schématiser les caractères du Tifinaghe en des formes géométriques simples (à l'exemple du Tifinaghe-ircam) mais ces caractères n'ont il pas porté un jour des signes esthétiques locaux selon la main du dessinateur ou graveur ? s'il s'avère que nous ne pourrions d'aucune façon « copier » les formes de glyphes du Tifinaghe des anciens monuments et inscriptions manuscrites amazighes qui auraient existés, alors ce sera une aventure hors paire d'offrir à cette magnifique graphie ce qu'elle mérite en traitement typographique.

Dans le court terme, avec le lancement de la numérisation du Tifinaghe, l'histoire entière du ce dernier risque de perdre beaucoup de son originalité. Puisque, au moment où les typographes et les grandes fonderies internationales comme celles de Microsoft, Adobe, etc, vont se lancer dans la création des polices de caractères Tifinaghes, alors rien ne fera marcher ces machines de production en arrière : Les courbes, les métriques, les algorithmes et les stratégies de hinting, etc, seront tous figés. Pour éviter cette catastrophe, il faut doter ces machines de production par des glyphes qui soient, typographiquement, les plus fidèles possible à leurs vrais formes : sans erreur qui ne sera pas ainsi pour les fabricants de fontes. Bien entendu, les outils de la typographie numérique pour l'optimisation du rendu de la graphie sont communs à toutes les graphies.

LES GLYPHES DE TIFINAGHE :

Chaque système de graphie est influencé complètement par les outils de son écriture. La graphie Tifinaghe, elle, a été marquée par les particularités de l'art rupestre, puisque la majorité des écritures anciennes trouvées jusqu'à maintenant sont des scripts gravés sur la pierre. Et c'est sans doute de là que le Tifinaghe a fondé sa forme actuelle.

En effet, de point de vue géométrique, Tifinaghe est schématisé sous forme d'un ensemble de figures géométriques à la fois simples et variées : cercles, demi-cercles, verticaux, horizontaux, diagonaux,...

Un seul glyphe peut rassembler plusieurs de ces formes pour former un contour géométrique. Le tableau suivant divise le répertoire des caractères Tifinaghes :

Verticale	⊕ ✖ ✖ ⑆ ⑆' + E I E C H Z U X Y H
Horizontale	⊖ x 8 ✖ ⑆ ⑆' Z X X' + E C H Y H U
Diagonale	⑆ ⑆' λ x x' Λ Q O X Z
Cercle	⊙ ○ 8 8' ⊖ ⊕ ⊙ Q
Demi-cercle	✖ ✖ c
Trait cassé	ξ >

DES GLYPHES A TRAVAILLER :

Il est connu dans les environnements des typographes et programmeurs que chaque glyphe de chaque graphie a ses particularités. Donc des algorithmes et des stratégies sont établis pour remédier minutieusement à chaque difficulté, ceci constituera par la suite le patrimoine très précieux de la société ou du typographe y travaillant.

Le Tifinaghe ne fait pas l'exception de ce qui précède. En fait, Le répertoire des glyphes du Tifinaghe se constituent remarquablement de deux types de formes qui rendent la tache d'optimisation de rendu typographique assez intéressante : les formes arrondies et les diagonales. Sachant que les logiciels d'édition de fontes ne traitent pas tous les diagonaux, le « hinting » des diagonaux et des formes courbées - afin de les rendre correctement lisse sur écran - n'est pas une tâche aussi évidente qu'on peut probablement le croire.

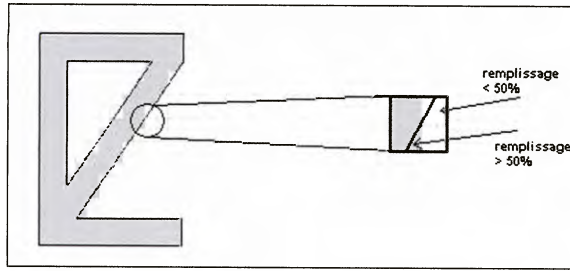
LES DIAGONAUX ET LES COURBES CIRCULAIRES :

En effet, Le « hinting » des diagonaux et des courbes circulaires, de façon à les rendre régulières et correctes à l'affichage, se révèle une tâche nécessitant beaucoup de techniques et d'expériences et qui consomme beaucoup du ressources, notamment un temps de travail considérable pour l'optimisation du rendu sur écran.

Environ 80% des caractères du Tifinaghe contiennent des parties circulaires, courbées et des diagonaux. Ces formes nécessitent un traitement spécial pour éviter les problèmes d'arrondissement des images bitmaps des glyphes.

Dans Tifinaghe, les diagonaux prennent des dimensions assez grandes, souvent du bout en bout prenant toute la largeur du caractère (exp. ⑆, λ, x,...), ce phénomène expose les glyphes de Tifinaghe beaucoup plus au problème très classique des « escaliers » :

Comme décrit dans l'image suivante pour la lettre Z « yaq » :



Les formes circulaires sont aussi encombrantes dans le « hinting » des glyphes que les diagonaux.

En réalité, le « hinting » est une notion relative à la graphie en question, en effet les mécanismes de « hinting » définis par les pionniers de la création des polices de caractères pour les fontes latines sont différents de ceux de l'arabe, l'hébreu, le devanagari, etc. Chaque graphie a ses particularités et ses algorithmes d'interprétation. En s'intéressant plus au Tifinaghe, cette magnifique graphie verra aussi, au fil de temps, sa technique de hinting jaillir, sans doute grâce aux travaux de typographie appliquée au Tifinaghe.

Conclusion :

La graphie Tifinaghe est une graphie d'un âge d'environ 25 siècles. Elle a sa propre histoire et ses formats d'apparition n'ont cessé de se développer depuis l'apparition de la toute première écriture amazighe jusqu'au néo-Tifinaghe. Aujourd'hui, et comme tout alphabet, le Tifinaghe a besoin de suivre les nouvelles technologies. Et ceci, du point de vue typographique, avec la forme la plus originaire et le rendu le plus optimal possible.

En fait, à la promotion typographique du Tifinaghe plusieurs défis s'en suivent: que ça soit dans le domaine de la typographie traitant le didacticiel et l'enseignement de la graphie ou au niveau de la typographie numérique traitant la production des polices de caractères et l'optimisation de leurs rendus. Ce dernier domaine de la typographie numérique s'est révélé un sujet clé pour réussir l'affichage du tout texte sur écrans à faible résolution (traitement de texte, Web,..), et ceci depuis l'apparition de la typographie numérique. Pour remédier à ces problèmes, les promoteurs de la typographie numérique ont mis à disposition plusieurs outils de retouche de dessins et procédures qui permettent le traitement des plans de bits des glyphes. et produire des polices de caractères de qualité.

Bibliographie Succinte :

- <http://www.microsoft.com/typography/>
- Ahmed Skounti, Abdelkhalek Lemjidi et El Mustapha Nami « Tirra, aux origines de l'écriture au Maroc »
- Roger D. Hersch "Font Rasterization: the State of the Art"
- Jacques André, Roger D. Hersch « Enseigner la typographie numérique », rapports de recherches N° 1618 INRIA-Rennes.
- Y. Haralam bons, «Fontes et Codages», O'Reilly France, 2004, ISBN2-84177-273-X

LA POLICE OPENTYPE HAPAX BERBÈRE

Patrick Andries

Conseils Hapax <http://hapax.qc.ca>
Membre du consortium Unicode

Cet article décrit la police Hapax Berbère. Elle a été dessinée dans le cadre de la proposition⁽¹⁰⁶⁾ d'addition de l'écriture tifinaghe à l'ISO/CEI 10646 et à Unicode.

Une police OpenType libre de droit

Hapax Berbère est une police OpenType⁽¹⁰⁷⁾ libre de droit. Ceci signifie que vous pouvez la télécharger⁽¹⁰⁸⁾ et la distribuer sans aucune restriction. Notez qu'elle n'est pas optimisée pour un affichage de qualité à l'écran, mais qu'elle devrait être appropriée pour l'impression de textes illustratifs écrits en tifinaghe.

Différents sous-ensembles

La police Hapax Berbère est composée de plusieurs ensembles de caractères pour répondre aux besoins des utilisateurs voulant transcrire des textes berbères à l'aide de l'écriture tifinaghe ou de l'écriture latin.

Certains systèmes informatiques - plus particulièrement les interpréteurs de commande de systèmes d'exploitation - ne peuvent prendre en compte qu'une police de caractères à la fois et requièrent la présence des caractères Ascii de base. C'est pour permettre l'utilisation du tifinaghe sur ces systèmes qu'on a inclus les caractères de base latins dans la police Hapax Berbère.

La police Hapax Berbère comprend en outre plusieurs sous-ensembles tifinaghes: les 4 sous-ensembles contemporains codés dans l'ISO/CEI 10646 :2003, des formes régionales, historiques et des ligatures.

(106) - Cf. <http://hapax.qc.ca/propo_tifinagh.pdf>.

(107) - Pour une description des polices OpenType, veuillez consulter l'article « Unicode et les polices - deux mondes » dans ces mêmes actes. Il est préférable de lire au préalable cet autre article.

(108) - À télécharger ici : <<http://hapax.qc.ca/polices/hapaxber.ttf>>, l'inventaire des œils de cette police est dressée dans le document suivant : <<http://hapax.qc.ca/polices/inventaire-des-oeils.pdf>>.

Sous-ensemble latin

Le sous-ensemble latin de la police comprend les lettres françaises habituelles ainsi que celles nécessaires à la transcription du berbère selon divers systèmes (Ġ, h, H, ε, Ğ, ğ, , γ, R, z, ?, _ , _ , dz, _ , dz ,ts, tf, ^h ,^A ,^j ,^w k^w, ġ, Đ, đ Đ,đ Ĥ, š, z,). On peut ainsi s'en servir pour écrire un titre comme celui de la grammaire «Tajerrumt n tmaziřt tamirant» de Kamal Naït-Zerrad.

Ce même sous-ensemble latin comprend le caractère _ qui prend dans la police Hapax la forme d'un 'aïn : Cette lettre s'appelle officiellement LETTRE MINUSCULE LATINE EJ RÉFLÉCHI par analogie de forme avec le U+0292 LETTRE MINUSCULE LATINE EJ⁽¹⁰⁹⁾ mais elle sert à noter la fricative pharyngale sonore. Pour des raisons liées à la facilité d'emploi⁽¹¹⁰⁾, on a préféré utiliser cette lettre latine à la lettre arabe Unicode U+0639 LETTRE ARABE 'AÏN. Il faut, toutefois, préciser que certains spécialistes pensent qu'une nouvelle lettre Unicode devrait être codée à cet effet (mais cela implique des délais avant de pouvoir échanger des documents utilisant ce signe) et que d'autres maintiennent que les utilisateurs devraient utiliser la lettre arabe et utiliser les caractères de commande bidirectionnel pour éviter les écueils auxquels nous faisons allusion dans la note de bas de page.

Composition des diacritiques latins

La police OpenType Hapax Berbère compose automatiquement les lettres latines adjointes à des diacritiques. Ainsi, si on adjoint un U+030A DIACRITIQUE ROND EN CHEF⁽¹¹¹⁾ à un U+0067 LETTRE MINUSCULE LATINE G, on obtient normalement un g_. Cette composition est automatique et relativement simple. Certaines compositions sont, cependant, plus difficiles à réaliser correctement et nécessitent l'intervention d'un moteur de rendu OpenType qui prend en compte les substitutions définies pour l'écriture latine.

Ce moteur de rendu se nomme Uniscribe sous Windows, il est utilisé par la plupart des applications, toutefois certaines d'entre elles - plus particulièrement celle du progiciel MS Office - ne font pas appel à Uniscribe pour certaines chaînes de caractères latins alors que d'autres comme Notepad utilisent toujours Uniscribe

(109) - Ce signe sert à noter des sons habituellement écrits en français à l'aide d'un « j » comme dans « jouet ».

(110) - En effet, l'insertion d'une lettre arabe (du bloc 0630-06FF) enclenche l'algorithme bidirectionnel qui déplace le curseur et les lettres d'une façon peu intuitive à l'intérieur de mots qui autrement n'utilisent que des lettres latines. Essayez, par exemple, d'écrire Le mot pour « équilibre » en berbère s'écrit ____ _del avec le 'aïn arabe (U+0639), sans manipulation complexe vous verrez sans doute à l'écran Le mot pour « équilibre » en berbère s'écrit ____ _del. L'article de Richard Ishida dans ces mêmes actes explique la raison de ce comportement surprenant et comment, en théorie, y remédier (ce n'est pas nécessairement facile dans MS Word).

(111) - Les noms sont les noms officiels français de l'ISO/CEI 10646:2003. En chef, signifie « suscrit », « écrit au-dessus », cf. couvre-chef, le terme a été préféré à « suscrit » pour éviter toute confusion avec « souscrit » qui lui est bien utilisé dans l'ISO/CEI 10646 :2003.

même avec des caractères latins. C'est ce qui explique que la même chaîne de caractères à diacritiques s'affichera différemment dans Word, dans PowerPoint et dans Notepad.

MS PowerPoint 2003 SP1 Sur Windows XP SP2	dẑ ƒ̂ D̂ d̂ d̂ efficace
MS Word 2003 SP 1 Sur Windows XP SP2	dẑ ƒ̂ D̂ d̂ d̂ efficace
NotePad 5.1 Sur Windows XP SP2	dẑ ƒ̂ D̂ d̂ d̂ efficace

Ainsi la police Hapax Berbère substitue-t-elle, par exemple, automatiquement aux lettres « ffi » une ligature quand le moteur de rendu effectue des substitutions OpenType pour des chaînes latines comportant plusieurs caractères de base, ce qui n'est ni le cas pour PowerPoint ni pour Word.

Les d à point et à macron souscrits sont le résultat d'une substitution OpenType (GSUB) plutôt que de déplacements relatifs OpenType (GPOS). Soit l'application de règles comme celles-ci :

```
ccmp {
  d uni0323 uni0320 → uni004403200323 ;
}
```

Ce qui se lit comme suit : un « d » suivi d'un U+0323 DIACRITIQUE POINT SOUSCRIT et d'un U+0320 DIACRITIQUE SIGNE MOINS SOUSCRIT doit être remplacé par le glyphe nommé «uni004403200323». Glyphe qui n'a pas de caractère Unicode associé. On aurait pu choisir un nom plus parlant comme « d_point_moins_souscrit » mais on a préféré suivre les recommandations d'Adobe dans le choix des noms de glyphe. En suivant cette convention, même lorsque des documents PDF ne contiennent pas d'informations faisant le lien entre les caractères Unicode et les glyphes de la police Hapax Berbère (cela dépend du logiciel qui a créé le PDF), des programmes tels que le lecteur d'Acrobat ou des moteurs de recherche comme Google essaieront d'analyser les noms des glyphes utilisés dans le PDF et de déterminer les caractères Unicode sous-jacents à ce texte.

On a préféré utiliser des GSUB plutôt que des GPOS pour deux raisons : l'une, noble, parce que les GSUB sont plus rapides que les GPOS, l'autre, pragmatique, parce que le logiciel de conception de polices utilisé, FontLab 4.6, ne prend pas en charge les GPOS avec ancre.

Le « ccmp » qui entoure cette substitution indique la fonctionnalité mise en œuvre par ces substitutions. Il s'agit de substitutions de composition, cette fonctionnalité est implicitement active et s'applique à toutes les écritures. Notons que c'est une des seules substitutions qui fonctionnent dans Word 2003 pour des caractères de base latins. Ce n'est malheureusement pas le cas de PowerPoint 2003 qui ne permet pas de telles substitutions de composition pour les caractères latins⁽¹¹²⁾.

AVIS de crénage

La police Hapax Berbère crène de nombreuses paires de lettres. C'est-à-dire qu'elle ajuste l'espacement entre les lettres d'un mot, pour éviter les effets disgracieux que l'on retrouve par exemple entre un A majuscule et un V majuscule. Remarquez ci-dessous comme la distance entre le A et le V est réduite par la police Hapax dans des applications comme NotePad ou BabelMap⁽¹¹³⁾ afin de produire un résultat du plus bel effet.

AVIS Times New Roman (Office)

AVIS Hapax Berbère (Office)

AVIS Hapax Berbère (pas Office)

Il est également possible de créer les caractères à l'aide de tables TrueType (le format précurseur à OpenType), ces tableaux moins puissants que la fonctionnalité correspondante OpenType sont pris en charge par les versions actuelles des logiciels de la suite Office de Microsoft. Nous n'avons toutefois pas ajouté ces tableaux TrueType à la police Hapax Berbère.

Les caractères tiffinaghes

La police contient les glyphes pour le bloc tiffinaghe codé par à l'ISO 10646 et Unicode. Elle contient exactement dans la zone à usage privé d'Unicode de très nombreuses formes régionales, archaïques et des ligatures.

(112) - Il semble que les prochaines versions de Windows (au nom de code provisoire de « Longhorn ») et d'Office devraient lever un bon nombre de ces restrictions.

(113) - <http://hapax.qc.ca/BabelMap_fr.html>

Le bloc tifinaghe Unicode est divisé en quatre sous-ensembles de caractères tifinaghes :

1. le jeu de base préconisé par l'Institut royal de la culture amazighe du Maroc (Ircam) ;
2. le jeu étendu de l'Ircam ;
3. d'autres lettres néotifinaghes en usage ;
4. des lettres touarègues modernes dont l'usage est attesté.

Jeu de base de l'Ircam

Le premier sous-ensemble reprend les symboles retenus par l'Institut royal de la culture amazighe du Maroc (Ircam) dont l'un des objectifs est l'unification de la graphie des dialectes berbères marocains contemporains tout en s'inscrivant dans la continuité historique de l'alphabet tifinaghe. À cette fin, l'Ircam a procédé à une analyse des données linguistiques du berbère marocain et des différentes variantes graphiques berbères existantes.

Sur le plan linguistique, l'Ircam n'a retenu que les oppositions phonémiques fonctionnelles et pertinentes dans le berbère marocain. Le nombre des unités phonémiques est établi à trente-trois (33).

Unicode ne code cependant directement que 31 lettres des 33 unités phonémiques de base pour leur adjoindre une lettre modificative. En effet, on peut former les deux unités phonémiques \mathfrak{R}^w (« \mathfrak{g}^w ») et \mathfrak{R}^u (« \mathfrak{k}^w ») à l'aide des consonnes \mathfrak{K} (« \mathfrak{k} ») et \mathfrak{X} (« \mathfrak{g} ») et de la lettre modificative U+2D6F $_$ prévue pour marquer la labialisation (le son « \mathfrak{w} ») d'une consonne en général. On obtient de la sorte un code compact et élégant qui permet d'éviter les complications liées au codage d'une même lettre exprimée à la fois sous une forme précomposée (\mathfrak{R}^u) et décomposée ($\mathfrak{X}+_$). Le $_$ se trouvant étant codé après toutes les lettres codées, le codage Unicode n'a pas de conséquence fâcheuse sur un tri qui utiliserait les numéros de caractère comme clés de tri, il s'agira sans doute du tri de repli pour les logiciels qui n'auront pas encore été adaptés pour offrir un tri tifinaghe explicite. Il est effet aisé de vérifier qu'avec ce codage l'ordre $\mathfrak{X} < \mathfrak{R}^u < \wedge$ est bien vérifié.

Notons que U+2D6F n'est pas un diacritique mais une lettre modificative, c'est-à-dire que cette lettre a sa propre chasse. Il s'agit d'un petit signe qui se colle à la lettre précédente mais qui ne la surmonte pas.

Jeu étendu de l'Ircam

Étant donné l'intérêt des caractères historiques relevés dans les textes scientifiques sur le tifinaghe et la nécessité de pouvoir rendre, dans certaines études

scientifiques ou dans les productions littéraires comme les pièces de théâtre, certaines particularités phonétiques régionales (comme la spirantisation des consonnes occlusives) et certains sons étrangers (comme /v/ et /p/, relevés également dans certains parlers marocains), l'Ircam a défini un jeu étendu de caractères.

Cette liste ne contient que six spirantes (obtenues par rotation des occlusives correspondantes) ainsi que les deux lettres YAVAΔ et YAP –. Pour l'Ircam, les affriquées peuvent être notées par des suites de deux lettres : +⊙ pour YATS, Λ✱pour YADZ, +C pour YATCH et ΛI pour YADJ.

Autres lettres néotifinaghes en usage

Le troisième sous-ensemble est formé de quatre lettres néotifinaghes utilisées fréquemment dans le reste du Maghreb et qui ne peuvent être considérées comme des variantes d'œil des caractères (de base ou étendus) de l'Ircam.






Ainsi, on pourrait être étonné de trouver dans le bloc Unicode les deux lettres _ et _ car dans l'orthographe préconisée par l'Ircam ces deux variantes régionales (la première néokabyle, la seconde prônée par Salem Chaker et encore utilisée par les Touaregs) qui représentent dj // doivent s'écrire sous la forme de deux lettres : Λ « d » + I « j ». Il faut cependant se souvenir qu'Unicode ne consigne pas un ensemble de sons mais de lettres et, pour les utilisateurs Kabyles et Touaregs, ces deux lettres (I et †) sont bien utilisées et sont différentes de la forme conjointe ΛI recommandée par l'Ircam.



























Lettres touarègues modernes dont l'usage est attesté

L'écriture tfinaghe touarègue traditionnelle ne note pas les voyelles (si on exclut le ya final et les semi-voyelles yi et you en fin de mot). Le sous-ensemble touareg au sein du bloc Unicode ne code pas les créations récentes, encore peu acceptées, comme les voyelles touarègues.

Bloc tifinaghe Unicode (U+2D30-U+2D7F)

Clé

	Tifinaghe Ircam de base
	Tifinaghe Ircam étendu
	Autres lettres néotifinaghes
	Lettres touarègues modernes attestées
	Réservé pour un codage ultérieur

	2D3x	2D4x	2D5x	2D6x	2D7x
0	◦	⊖	‡	Δ	
1	⊖	∅	!	⊏	
2	⊖	⋮	δ	⋈	
3	⌘	∧	⊘	⌘	
4	⌘	⌒	○	↑	
5	⌘	×	Ⓚ	⌘	
6	⌒	⋮	⌒		
7	∧	∅	⋮		
8	∨	⋮	⋮		
9	⊖	⋈	⊖		
A	⊖	⊏	⊖		
B	⊘	×	Ⓒ		
C	⌘	#	+		
D	⌒	⌒	⌘		
E	⋮	⊏	Ⓒ		
F	⌘	⊏	⊖		

Autres glyphes tfinaghés

La police Hapax Berbère contient également des glyphes pour les différents caractères libyques de Karl-G. Prasse (cf. un extrait ci-dessous), des ligatures touarègues, des variantes néoti_naghés, des lettres tfinaghés du Niger - que ce soient celles de l'Association pour la promotion des tfinagh ou celles du Tawellemmet - , les voyelles proposées par Hawad, les caractères dits « sahariens » et des caractères canariens. Comme ces lettres ne sont pas codées dans Unicode on leur a attribué des numéros de caractères de la zone privée (U+E000 à U+F8FF). Les différents « b » libyques mentionnées par Karl-G. Prasse ci-dessous sont donc codées dans Hapax Berbère dans l'intervalle U+F000 à U+F006.

Inventaire des oeils du tableau « Caractères libyques » de Karl-G. Prasse (1972, pp.153-154)1

Valeur	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
b F00x	⊙	⊖	⊠	⊕	⊞	⊟	⊠									
d F01x	∏	∏	∏	∏	∏	∏	∏	∏								
d F02x	∏	∏	∏	∏												
f F03x	⌘	⌘														
f F04x	∏	∏	∏	∏												
g F05x	∏	∏	∏	∏	∏	∏	∏	∏	∏	∏	∏	∏	∏			
g F06x²	∏	∏	∏	∏	∏	∏	∏	∏	∏	∏						
h F07x	∏	∏	∏													
h F08x	∏															

Biconsonnes optionnelles

Il existe un grand répertoire de ligatures (parfois appelées « biconsonnes ») touarègues. Tous les groupes consonantiques ne forment pas des ligatures. Seules les groupes de lettres dont la dernière consonne est t (p. ex. la désinence du féminin) ou ceux dont la première consonne est n peuvent fusionner. Ces ligatures sont loin d'être obligatoires. Il suffit d'ailleurs de parcourir les lettres adressées au Père de

LES GLYPHES DE TIFINAGHE :

Chaque système de graphie est influencé complètement par les outils de son écriture. La graphie Tifinaghe, elle, a été marquée par les particularités de l'art rupestre, puisque la majorité des écritures anciennes trouvées jusqu'à maintenant sont des scripts gravés sur la pierre. Et c'est sans doute de là que le Tifinaghe a fondé sa forme actuelle.

En effet, de point de vue géométrique, Tifinaghe est schématisé sous forme d'un ensemble de figures géométriques à la fois simples et variées : cercles, demi-cercles, verticaux, horizontaux, diagonaux,...

TEXTE (a)

1 : | : • □ ⊙ ▪ [sceau] Δ □ ⊙ ± | ^{S T}

2 □ | : • || | : Δ ○ | ' : || : | □ ○ ■ :

3 : || ' : || ' | : : • | ' | : ○ | : : || ' |

4 ■ ⊙ | : || : ○ : || : ○ : ⊙ + ^ Δ +

5 : > ⊙ : # > : : • ^ | < : + ○ : +

6 + | || ⊙ < | ⊙ ⊙ : : T ^

7 > | ^N _F ⊙ > ^ + ⊙ + + | □ ⊕ | : •

8 : || | ^{RT}

9 ⊙ || ' | : : • : || ... > | ○ | : : • ^ | : •

1 aWa NəK MūSA aGg aMaSTaN

2 aMəNū KaL N aHaGgaR iNnāN əHūləΓ-iN MaRaBW

3 HuLlaN HuLlaN əNniΓ-āK iNniN NəK əRiNəΓ HuLlaN

(114) - Rappelons que, dans ses variantes touarègues, le tifinaghe peut s'écrire de gauche à droite ou de droite à gauche. La forme caractéristique de certaines lettres asymétriques indique clairement le sens du texte. Ainsi le « k » touareg prend-il la forme _ dans un texte gauche-à-droite mais dans un texte droite-à-gauche.

Liant et antiliant

Il existe deux caractères Unicode qui permettent d'encourager ou de décourager la formation des ligatures au rendu, il s'agit du U+200D LIANT SANS CHASSE et du U+200C ANTILIAN SANS CHASSE, respectivement.

Comment coder des textes contenant des biconsonnes ?

La police Hapax Berbère contient de très nombreuses ligatures, elles ne sont cependant pas codées dans Unicode, mais plutôt dans la zone à usage privé. On pourrait croire qu'il n'est donc pas possible de coder des textes Unicode qui afficheraient des ligatures sans avoir recours à la zone à usage privé et à un accord préalable entre les différentes parties pour convenir de la position et de la signification de chacune de ces ligatures. Ce n'est pas possible si on désire envoyer des ligatures, mais il existe une autre façon d'obtenir des ligatures à l'affichage. Il s'agit d'utiliser les caractères LIANT et ANTILIAN.

Prenons, par exemple, le mot fictif «fta» (c'est-à-dire « afta ») et introduisons un liant entre le F et le T. On obtient alors la suite de caractères suivante :

U+2D30 U+2D3C U+200D U+2D5C U+2D30

Selon la police et le programme d'affichage utilisés, ce « mot » pourra s'afficher sous la forme d'un mot non ligaturé ou ligaturé. Si vous utilisez NotePad et la police Hapax Berbère sous Windows XP, le résultat apparaîtra sous la forme ligaturée de «fta». Dans d'autres cas, la chaîne apparaîtra sous la forme non ligaturée «fta».

Pour que cela fonctionne, la police Hapax Berbère définit la fonctionnalité OpenType « liga » sous la forme d'un ensemble de substitutions de ce type :

```
liga {  
  uni2D30 afii301 uni2D5C → uni2D30D200D2D5C ;  
}
```

Pour l'instant cette fonctionnalité n'est pas prise en compte par Word de Microsoft, mais bien par Notepad, InDesign d'Adobe, Illustrator CS d'Adobe, Photoshop CS d'Adobe, le traitement de texte Mellel pour Mac OS X et très probablement par les prochaines versions des systèmes d'exploitation Windows Longhorn de Microsoft et Mac OS X (peut-être même dans la version 10.4 Tiger).

Avantages du liant

Mais pourquoi avoir recours au liant ? Son plus grand intérêt réside dans le fait qu'il préserve le modèle caractère-glyphe⁽¹¹⁵⁾. Ce qui signifie qu'on peut facilement

(115) - Voir l'article du prof. Lahbib Zenkour dans ces mêmes actes à ce sujet.

ignorer le liant dans des processus comme la recherche, du tri, de rendu (avec une autre police par exemple) ou des outils de correction orthographique et obtenir le même résultat qu'un texte qui ne l'utiliserait pas.

On peut donc recommander la formation de ligatures (elles sont optionnelles, rappelons-le) à l'affichage tout en s'assurant que les logiciels interprétant les textes continueront de fonctionner de la même manière avec des textes pourvus de liant qu'avec ceux qui ont sont dépourvus.

Écrire des textes touaregs

La police Hapax berbère a implicitement adopté pour les caractères du bloc tfinagh Unicode une apparence néotifinaghe. Pour donner aux caractères Unicode une apparence touarègue, on aurait pu choisir d'avoir recours à une fonctionnalité OpenType comme « salt » (variantes stylistiques), seulement l'utilisation d'une telle fonctionnalité nécessite habituellement un choix interactif peu conciliable, par exemple, avec la rédaction de pages Web en touareg. La solution la plus simple nous a semblé de simplement produire une autre police à l'aspect touareg indexée à l'aide des mêmes valeurs Unicode. Libre à vous de créer de nouvelles polices qui correspondent mieux à une tradition calligraphique touarègue particulière en vous servant d'Hapax Touareg comme base.

Échantillon en "Hapax Berbère" : ◦ ⊖ ⊕ ℥ № + ⊕

Correspondance Hapax Touarg : • ⊖ ⊕] [| / + □

L'affectation de cette police à un passage HTML peut se faire à l'aide d'une feuille de style CSS ou de l'attribut style :

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html;
    charset=utf-8">
  <title>Lettre n° 12 au Père de Foucauld</title>
</head>
<body style="font-family: hapax touareg">
<h1>Lettre n&deg;12 au Père de Foucauld</h1>
  <ol>
    <li>+O8++O ï+·+ ⊕ ^ ï | ⊙ ⊕ O ∴ ◦</li>
    <li>+C^ï++ | 8 ï C ∫ ✖O</li>
  </ol>
</body>
</html>
```

Cette petite page HTML s'affichera comme suit dans Firefox Mozilla :

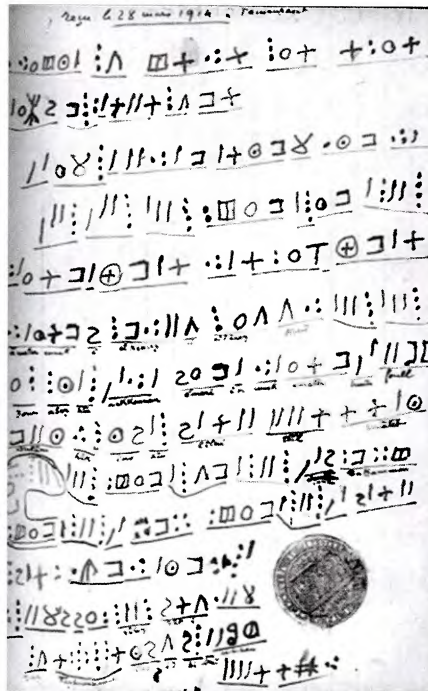


Lettre n° 12 au Père de Foucauld

1. +O:+ +O: +:+⊠ Λ: |⊙⊠O:••
2. +⊠ Λ: +∨+ | :|⊠ ∫XOI

Écrire des textes tfinaghés droite-à-gauche

Dans certaines régions, comme l'illustre la lettre reçue par le Père de Foucauld ci-dessous, on préfère écrire les tfinaghés de droite à gauche tout en réfléchissant les caractères.



Pour écrire les caractères tfinaghés de droite-à-gauche, il faut d'abord forcer la directionnalité droite-à-gauche. Cela peut se faire en XHTML grâce à la balise `<bdo>`. Pour « miroiter » les caractères, il existe deux possibilités : la confection d'une police à œils réfléchis ou l'ajout à une police standard comme Hapax Berbère de la fonctionnalité OpenType « rtl » (variantes droite-à-gauche). La fonctionnalité « rtl » prend la forme d'une série de substitutions biunivoque d'un glyphe gauche-à-droite par le glyphe correspondant droite-à-gauche :

```

rtla {
    uni2D39 → uni2D39.dag;
    uni2D3A → uni2D3A.dag;
    ...
} rtla;

```

Idéalement, selon les spécifications OpenType, dans un contexte forcé droite-à-gauche⁽¹¹⁶⁾, les logiciels de rendu devraient réfléchir les glyphes des caractères qui appartiennent à une écriture gauche-à-droite (c'est la directionalité implicite du tfinaghe en Unicode) et pour lesquels la police OpenType sélectionnée définit des variantes droite-à-gauche. Malheureusement, cette fonctionnalité n'est pas encore prise en compte par Uniscribe et donc la grande majorité des applications Windows.

C'est pourquoi nous avons créé une police Touareg droite-à-gauche⁽¹¹⁷⁾ aux œils réfléchis qui peut servir à écrire des textes touaregs traditionnels écrits de droite à gauche.

On écrira donc en HTML les premières lignes de la lettre à Foucauld illustrées ci-dessus de la sorte :

```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>Lettre n&deg; 12 au Père de Foucauld</title>
</head>
<body style="font-family:hapax touareg d&g;font-size:14pt;">
<bdo dir="rtl">
<h1>Lettre n&deg; 12 au Père de Foucauld</h1>
<ol>
<li>+O%++O ; + ; + ⊕ ∧ ; | ⊙ ⊕ ○ ; ° </li>
<li>+⊠∧ ; +⊠+ | % ; ⊠ ⋈ * ○ | </li>
<li>...</li>
<li value="11"> | ; °⊠⊠⊙ ; ⊠ | &#x200d;∧ ° % + | ⋈ ; </li>
</ol>
</bdo>
</body>
</html>

```

(116) - Précisé, par exemple, par une balise <bdo> en HTML.

(117) - Les polices Hapax Touareg et Hapax Touareg D&G sont téléchargeables aux adresses suivantes : http://www.hapax.qc.ca/polices/hapaxtouareg.ttf et http://www.hapax.qc.ca/polices/hapaxtouaregdag.ttf. Vous pouvez les modifier, mais ne vous plaignez plus par la suite !

Le ‍ ; est un appel de caractère numérique qui représente un liant, ce liant permet de favoriser la formation de la ligature « nd » qui, dans cette variante touarègue, prend la forme \wedge .

On obtient alors dans l'Explorateur Internet de Microsoft, le résultat suivant :

Lettre n° 12 au Père de Foucauld

```

•:O⊗⊙| :∧ ⊞+:+ :O+ +:O+ .1
  IOXʔ ⊙: | +\\+ :∧ ⊙+ .2
                                     ... .3
  :| + : •∧⊙: ⊙⊙:•:| .11
  
```

Remarquez que dans les deux cas décrits ci-dessus - que l'on force le texte à s'écrire de droite à gauche ou non - le texte brut⁽¹¹⁸⁾ (à savoir « +O8+ +Oï +:⊕ ∆: IO⊕O :o +⊕ ∆: +#+ | 8:⊕ ʔ#O| ») reste identique. On respecte donc le modèle caractère-glyphe et on permet de la sorte aux mêmes outils informatiques d'opérer sur ces textes sans en connaître le rendu exact.

Attention à la ZUP et au bloc latin !

L'auteur espère avoir montré qu'il est tout à fait possible de représenter⁽¹¹⁹⁾ de manière satisfaisante le berbère (y compris le touareg) à l'aide d'Unicode et de polices OpenType.

Si vous deviez créer une police berbère, contrairement à ce qu'on trouvait avant la normalisation ISO des tifinaghes, n'affectez surtout pas à vos glyphes les numéros de caractère du bloc latin, cela irait totalement à l'encontre de la philosophie Unicode et de manière plus pragmatique signifierait qu'une série de processus ne fonctionneront pas correctement : la vérification orthographique et la recherche de textes tifinaghes sur Google, par exemple. N'utilisez pas non plus la zone à usage privée pour coder des caractères tifinaghes si vous voulez vous assurer que vos données puissent être échangées avec des tiers sans un accord préalable avec ce tiers. Employez plutôt les numéros de caractères Unicode et utilisez des polices différentes pour afficher des formes différentes ou les substitutions OpenType (dans des fonctionnalités de type « variante ») dont nous avons parlé au cours de cet article.

(118) - Le texte dépourvu de tout balisage ou mise en forme (« le stylage »).

(119) - Pour la transcription en écriture arabe, deux lettres (peut-être plus) semblent manquer à Unicode : $_$ et $_$, elles se trouvent à des fins illustratives aux positions U+F860 et U+F863 dans la zone à usage privé. Elles ne peuvent être utilisées dans un texte courant en écriture arabe puisque l'algorithme bidi ne connaît pas ces lettres (il les considèrera sans doute comme des lettres latines et ne connaît pas ses aspects contextuels). Une proposition sera bientôt présentée à l'ISO afin de compléter le jeu de caractères arabes afin de permettre la représentation des textes berbères écrits à l'aide de l'écriture arabe, plus particulièrement ceux du Sous-saharien marocain.

CODAGES, PLURIEL, DES CARACTERES

Jacques André

IRISA/INRIA-Rennes
Campus universitaire de Beaulieu
F-35042 Rennes cedex

Résumé. Les utilisateurs d'ordinateurs ne sont pas toujours conscients que ce qu'ils voient sur leur écran ou en sortie d'imprimante n'est jamais exactement ce qu'il y a dans les fichiers qu'ils ont en mémoire ou qu'ils transmettent. En particulier, les textes sont codés et il y a de nombreux codages, variant notamment selon l'usage qui est fait de ces fichiers.

Unicode n'est qu'un mode de codage parmi d'autres et on montre ici de quel type de codage il s'agit.

1. Introduction

Une des évolutions marquantes de l'informatique aura été l'invention de la souris et de l'écran, plus précisément de méthodes ergonomiques d'utilisation des ordinateurs basées sur une gestuelle simple et efficace : on travaille directement sur l'écran pour obtenir ce que l'on veut (par exemple en sortie d'imprimante). Ce progrès considérable, qui a permis la diffusion massive de l'informatique, revient en fait à cacher la programmation fastidieuse et la manipulation délicate des fichiers. Mais ce faisant, on cache aussi certains aspects de ces fichiers, notamment leur codage interne. Certains utilisateurs peuvent alors ne pas comprendre que ce qu'ils voient sur leur écran est une interprétation, un décodage, de ce qu'il y a dans leurs fichiers. Ne pas comprendre ce qu'est un codage. Et ensuite ne pas comprendre ce qu'est un codage comme Unicode, s'en plaindre, trop lui demander ou au contraire le sous-employer.

Nous nous proposons ici, en partant d'exemples naïfs, quitte parfois à noyer le poisson dans l'eau, de montrer la variété des codages de caractères et qu'ils n'ont de sens qu'au travers des outils (programmes) qui les utilisent ou les convertissent. Nous montrerons ainsi en quoi Unicode [2] n'est qu'un codage parmi d'autres, laissant aux autres auteurs de ce colloque de montrer ce qu'il apporte de plus et la façon de s'en servir.

2. Mémoire et codage des caractères

On peut considérer, en première approximation, que la mémoire d'un ordinateur est une suite de cases dans lesquelles on peut mettre des chiffres ; supposons qu'on peut en mettre deux⁽¹²⁰⁾. Les contenus possibles des cases sont alors

00	01	...	99
----	----	-----	----

. Un fichier peut-être vu comme un suite de telles cases.

Pour pouvoir mettre autre chose que des nombres, notamment des lettres, dans ces fichiers, il « suffit » de dire qu'à un nombre correspond une lettre, par exemple que

21

 correspond à « A », que

22

 correspond à « B », ..., que

46

 correspond à « Z ». On appelle « codage » la relation entre ces nombres et ces caractères et on dit, par exemple, que « B » a pour code 22.

Il faut que l'on sache entrer des caractères dans un fichier : lorsque l'on tape sur une touche, par exemple « B », alors le code correspondant, ici

22

, est entré en mémoire. Il faut aussi pouvoir « sortir », imprimer ou afficher sur un écran, une lettre, c'est-à-dire construire l'ensemble des pixels noirs et blancs ressemblant à, par exemple, un « B ». C'est une machinerie assez complexe⁽¹²¹⁾, mais on peut comprendre aisément qu'il suffit d'avoir 26 programmes, numérotés de 21 à 46 pour imprimer les 26 lettres dont les codes sont 21, 22, ..., 46.

Un tel codage comprend, ici, 100 possibilités. On peut alors définir des codes pour les minuscules, par exemple dire que

51

 correspond à « a », que

52

 correspond à « b », etc. ; on peut définir des codes pour certains signes, par exemple dire que

81

 correspond à « ' », que

81

 correspond à «. », que

83

 correspond à « » (le caractère espace), etc. Par ailleurs, il faut aussi pouvoir écrire des chiffres et on leur donne aussi un numéro, comme pour les lettres, en disant que

11

 correspond à « 1 », que

12

 correspond à « 2 », etc.

Finalement, un texte est donc un fichier, une série de cases contenant des nombres mais à chacun desquels on donne « implicitement » la valeur d'un caractère selon un code donné. La figure 1 montre une phrase en mémoire utilisant le code que nous venons de proposer.

30	81	51	59	83	16	17	83	51	64	69	81
J	'	a	i		6	7		a	n	s	.

Figure 1. *Un fichier : en haut contenu numérique du fichier, en bas interprétation selon notre codage.*

(120) - Dans un premier temps, on considère qu'il s'agit de chiffres décimaux, 0 à 9. On verra qu'il s'agit bien sûr des fameux chiffres binaires et qu'Unicode parle plutôt de chiffres hexadécimaux 0 à 9, A, B, C, D, E, et F. Il ne s'agit finalement que d'un niveau de codage supplémentaire ! Souvent ces « cases » sont des octets de huit bits (valant 0 ou 1) ce qui permet donc écrire les nombres de 0 à 2⁸-1 soit 255.

(121) - On trouvera une synthèse sur ce sujet dans [3].

3. Le codage ASCII et ses « variantes »

De nombreux codages du type de celui que nous venons de montrer ont été proposés et bien sûr ceci a conduit à une « Tour de Babel » des écritures, chaque fabricant d'ordinateur ayant son propre codage (voire plusieurs). Pour communiquer entre eux, par cartes ou rubans perforés, par bandes magnétiques, disquettes, Internet, etc. il a bien fallu s'entendre. D'où la naissance de normes de codage.

La première norme ⁽¹²²⁾ qui aura vraiment eu une résonance internationale et qui perdure, aura été l'ASCII. Son principe est de coder 128 caractères⁽¹²³⁾ dont seulement 95 caractères sont affichables ou imprimables (les codes de valeur 0 à 31 et celui 127 sont dits « de contrôle » et n'ont pratiquement plus de sens aujourd'hui - voir section 4).. La table 1 montre les caractères affichables d'Ascii.

Tableau 1 : le codage ASCII

Code	Car.	Code	Car.	Code	Car.	Code	Car.	Code	Car.	Code	Car.
32	(espace)	48	0	64	@	80	1.1.1.P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	U	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	

En Ascii, le texte de notre premier exemple s'écrit donc :

(122) - De nombreux autres codages ont bien sûr été proposés, même avant l'informatique (et notamment ceux du télex). Voir [1] pour l'histoire de ces codages.

(123) - 128=27 : ASCII utilise des cellules de 7 bits valant 0 ou 1. On parle souvent de codage à 7bits. Notons qu'on parle aussi d'« Ascii 8 bits », ce qui n'existe pas formellement : il s'agit souvent d'une des extensions à 8 bits, privées, de l'Ascii.

74	39	97	105	32	54	55	32	97	110	115	46
J	.	a	i		6	7		a	n	s	.

Figure 2. Le fichier de la figure 1 cette fois en Ascii ; en haut : code Ascii (en base 10) en bas : interprétation

Ce codage est encore très utilisé, d'autant plus qu'il est à la base du codage Latin-1 et d'Unicode et qu'il sert à coder des codages plus étendus (voir section 10).

4. Balisage et premiers protocoles de niveau supérieur

Les « caractères de contrôle » qui apparaissent dans le codage Ascii sont en fait le souvenir de codes utilisés essentiellement à des fonctions de transmission (par exemple « message reçu ») ou de mise en page (par exemple « fin de ligne ») dans des codages plus anciens (comme par exemple le Téléx). Cette idée de mélanger des caractères avec des « commandes » a été reprise, notamment en photocomposition, avec le concept de balise (en anglais mark ou tag). Une balise se distingue des caractères de commande par plusieurs points :

- ▲ une balise n'est plus un caractère spécial mais la suite de caractères ordinaires;
- ▲ une « syntaxe » doit permettre de distinguer les balises des caractères ordinaires,
- ▲ il faut alors un programme (appelons le un formateur) pour manipuler le texte du fichier ainsi codé, pour en extraire les commandes et les appliquer.

Ces balisent permettaient essentiellement de

- ▲ faire des choix graphiques pour les caractères (police, corps, etc.),
- ▲ faire des mises en pages (justification, interlignage,...).

1.1.1.2 Exemple

Le pilote de la photocomposeuse suppose que la syntaxe des balises est la suivante :

- ▲ les balises ont la forme <xxx> où < et > sont des caractères réservés (ils ne servent qu'à ça) et où xxx est une suite de caractères. Par exemple <FL> est la balise indiquant « Fin de Ligne ».
- ▲ si on veut utiliser les caractères « < » et « > » dans un texte, il faut utiliser respectivement à leur place « <<> » et « <>> ». Pour écrire « si x>0 » il faut alors coder « si x<>>0 ».

Si on veut obtenir le texte suivant :

Les principaux manuels typographiques sont :

- ◆ Manuel typographique de Fournier, 1764,
- ◆ Manuale tipografico de Bodoni, 1818,
- ◆ etc.

Le code est le suivant :

```
<F1> <CP12>Les principaux manuels typographiques sont :<FL>  
<C>- <F2><CP12>Manueltypographique<F1><CP12> de Fournier, 1764,<FL>  
<C>- <F2><CP12>Manuale tipografico<F1><CP12> de Bodoni, 1818,<FL>  
<C>- etc. <FL>
```

Où

- <F1> signifie « prendre la fonte 1 » (supposée être ici du Times romain)
- <F1> signifie « prendre la fonte 2 » (supposée être ici du Times italique)
- <CP12> en corps 12
- <FL> fin de ligne
- <C> espace de 1 cadratin

Chaque formateur a en fait son propre langage de balises.

5. Codage visible ou codage caché

Cette forme de codage est en gros celle que l'on trouve dans de nombreux langages de formatage tels que TeX, HTML, etc. Le processus de traitement est alors le suivant :

1. écrire, en général à l'aide d'un éditeur de texte, le fichier codé, en tapant explicitement les balises voulues,
2. soumettre ce texte codé au formateur qui fera le travail voulu (impression notamment)
3. vérifier la sortie ; en cas d'erreur, reprendre le fichier et le corriger avec l'éditeur : faire donc autant d'étapes 1 et 2 que nécessaires.

Bien qu'elle ait certains avantages (notamment un plus grand contrôle de la part de l'utilisateur), cette méthode a l'inconvénient d'être fastidieuse et surtout d'être très dépendante du langage propre au formateur.

C'est alors que, grâce à l'invention des souris et de l'utilisation d'écran, sont apparues des méthodes très différentes de « programmation » : alors que l'on donnait des « commandes » pour piloter l'ordinateur et par là les imprimantes, on s'est mis à dire non plus « comment faire » mais « ce qu'on veut » plus précisément à montrer sur écran ce que l'on veut obtenir. Ces commandes ont ainsi imité la pratique des graphistes (couper-coller par exemple) mais aussi, par la combinaison

de menus, de cliques, etc., donnent l'impression aux utilisateurs qu'ils travaillent directement sur le document final, alors qu'en fait ils travaillent, sans le savoir, sur un texte balisé... En effet, supposons qu'on ait déjà tapé en Microsoft-Word ⁽¹²⁴⁾ un bout de texte et que l'on voit sur notre écran

- Manuale tipografico de Bodoni, 1818,

alors le fichier contient « -Manuale tipografico de Bodoni, 1818,». Si maintenant on sélectionne avec la souris Manuale tipografico et que l'on clique sur « italique», alors le fichier est modifié pour contenir

- {\i Manuale tipografico} de Bodoni, 1818,

C'est-à-dire que le formateur a, d'une part, ajouté dans le fichier la balise \i et mis les « parenthèses » {et} pour délimiter la portée de l'italique et, d'autre part, réactualiser l'affichage sur l'écran en tenant compte de l'italique demander par l'utilisateur.

Cette façon de cacher les codes à l'utilisateur va aussi être utilisée pour des caractères spéciaux.

6. Langages structurés et langages de description de page

Afin d'être aussi universels que possible, les langages balisés comme celui décrit en section 4 ont aussi évolués, mais dans deux directions opposées.

6.1. Documents structurés

La première a été d'utiliser des balises le plus génériques possibles et indépendantes de toute notion graphique, par exemple de dire « ceci est un titre » (et de ne pas se soucier de savoir s'il sera imprimé en gras ou en corps 16, ce choix étant fait à un autre moment) ; ce balisage est par ailleurs « parenthésé », soit qu'il existe une balise spécifique pour marquer la fin de la portée d'une balise de début (par exemple <i> marque le début de l'italique et </i> la fin), soit qu'un symbole de « parenthèses » marque la portée (par exemple les accolades dans {\i ...} de notre exemple en section 5). Enfin, le document est présenté comme un arborescence d'éléments emboîtés les uns dans les autres (une section est formée de sous-sections, elles-mêmes formées de paragraphes, etc.).

(124) - Pour prendre le formateur le plus répandu. Pour avoir une idée du fichier sur lequel Word travaille, il suffit de l'écrire dans un fichier en format RTF (Rich Text Format) et de le lire avec un autre formateur. Le véritable fichier interne de word (généralement en .doc) comprend aussi d'autres informations, notamment l'historique c'est-à-dire la trace des couper-copier-coller, etc.

L'exemple donné en section 4 peut être balisé de façon structurée comme :
<paragraphe>

Les principaux manuels typographiques sont :

```
<liste>
  <elem><livre>Manuel typographique</livre>
de Fournier, 1764,
  </elem>
  <elem><livre>Manuale tipografico</livre>
de Bodoni, 1818,
                                </elem>
                                ...
                                </liste>
</paragraphe>
```

Parmi les langages structurés, on peut citer LaTeX, HTML et XML, mais aussi le format PDF.

6.2 Langages de description de page

Alors que les langages structurés ignorent autant que faire se peut le moteur d'impression, une autre façon de faire est justement de s'adresser à eux. Typiquement, PostScript est un langage de commandes graphiques traçant des figures géométriques sur une surface (feuille de papier ou écran) ; les caractères ne sont que des figures géométriques particulières⁽¹²⁵⁾

6.3 Processus éditorial

Ces types de langages ne sont pas antinomiques mais complémentaires. La façon de travailler est en effet la suivante :

1. L'utilisateur entre un document dans un fichier, soit en le codant lui-même, soit en utilisant un système interactif comme Word qui va donc produire de façon cachée un fichier balisé.
2. Quand l'utilisateur demande l'impression de son document, le formateur traduit le fichier balisé en un fichier pour un langage de description de page et l'envoie à l'imprimante qui va alors noircir le papier comme voulu.

(125) - Ces caractères sont des éléments d'une « fonte » dont PostScript a défini pratiquement le premier modèle, les fontes de Type 1. Voir [3].

7. Codage de caractères et protocoles de niveau supérieur

Les exemples précédents montrent qu'un document (ici un texte) est un fichier contenant des caractères et des balises⁽¹²⁶⁾ décrivant la mise en page, mais aussi spécifiant certains caractères... En effet la balise <livre> de la section 6.1 peut indiquer non seulement que ce qui suit est le nom d'un livre mais aussi que ce nom doit être mis en italique. De même la balise <elem> indique le début d'un élément de liste mais aussi que cet élément est à marquer par un tiret «-» ou mieux par «-» comme on le fait en typographie fine. De même, en typographie fine, on utilise des ligatures : on remplace par exemple les deux lettres « fi » par la ligature « fi ». Certains langages de formatage proposent alors des conventions comme celle d'écrire <lig>fi</lig>. Un morceau de fichier comme

```
<elem><livre>Manuale tipogra<fig>fi</fig>co</livre>
```

comprend donc

- ▲ des caractères codés ;
- ▲ des balises codées à l'aide de caractères définissant un protocole supérieur à celui du codage (ici donc <elem> impliquant l'emploi d'un tiret cadratin « - », <livre> celui d'italiques jusqu'à </livre> et <lig>fi</lig> l'usage de la ligature « fi ».

Un autre exemple de protocole supérieur est celui qui disait que dans une chaîne écrite en Ascii, la succession d'une voyelle (par exemple « e » et d'un accent flottant (par exemple « ` ») indique la voyelle surmontée de cet accent. (« è »). Ceci fait partie des extensions à un code ; mais en Unicode il s'agit même de la forme normale des caractères accentués !

Il est important de voir qu'alors un fichier n'a de sens que pour un formateur donné en connaissant le codage des caractères et le sens des balises. Un fichier Word écrit sur un PC n'est pas toujours lisible sur un autre...

8. À propos des fontes

8.1 Codage des fontes

Sans entrer ici dans le détail⁽¹²⁷⁾, on peut dire que les fontes utilisent un autre système de codage. : à chaque caractère imprimable correspond un dessin auquel est affecté un « nom » (une suite de lettres) et non plus un numéro comme dans le codage Ascii. Par exemple, le dessin de la lettre « a » s'appelle... a, celui de la lettre « é » s'appelle « egrave » et celui de la ligature « fi » s'appelle fi. Pour imprimer un texte, le formateur demande donc à la fonte d'utiliser les caractères dont il donne la suite des noms. Exemple (en supposant que le fichier contienne <lig>fi</lig>de le.).

(126) - Elles mêmes écrites avec des caractères. Nous ne donnerons ici que des exemples où l'Ascii est suffisant pour écrire ces balises.

(127) - Voir à ce sujet [3].

Caractère codé	Nom du caractère à dessiner	Caractères dessinés
<		
l		
i		
g		
>		
f		
i		
<		
/		
l		
i		
g		
>	fi	fi
d	d	d
e		
`	egrave	è
l	l	l
e		
.	e point	e.

Signalons, mais ceci ne relève plus de cet article, que le choix des fontes est également indiqué dans le fichier interne. Par dans le fichier RTF de notre texte on trouvera

`\fonttbl{\f0...{...Times New Roman;}...}` définition de la fonte f0

...

`{\f0 ... Les principaux manuels... }` utilisation de la fonte f0

et de même pour les indications de corps, de couleur, etc.

8.2. Fontes et glyphes

Si on utilise la fonte Times New Roman, on imprime un certain « a ». Si on utilise Courier bold italic, alors ce qui est imprimé est « a » qui n'est pas tout à fait identique au précédent, tout comme les deux suivants « a » et « a ». On dit que ce sont des « glyphes » du même « caractère abstrait » qu'est la minuscule latine a. De même « fi » est un glyphe correspondant aux deux caractères successifs « f » et « i ».

On peut dire en gros que les fontes impriment des glyphes alors que les codages traitent de caractères.

- ◆ **codage par nom** ; par, exemple « Å ; » pour Å (HTML), « \delta » pour (TeX),
- ◆ **codage par numéro**, on donne le numéro du code, par exemple en décimal. Ainsi « é » dont le numéro de code est 233 en latin-1 peut se coder é en HTML.

De nombreux navigateurs ou créateurs de pages Web utilisent les deux dernières méthodes pour coder en Ascii n'importe quel caractère Unicode !

11. Unicode

Au vu de ce que nous venons de voir, Unicode est un codage de caractères avec les propriétés suivantes :

- ▲ Unicode affecte à chaque caractère un numéro et un nom (par exemple le couple [65, LETTRE LATINE MAJUSCULE A] pour « A »). Ces numéros sont compris entre 0 et 1 048 576 mais il est d'usage de les numéroter en hexadécimal, le code de « A » étant alors 41.
- ▲ Il existe des formes de codage « condensé » permettant aux fichiers d'être plus courts (UTF8 par exemple). Par ailleurs certains formateurs ont donné la possibilité de surcoder Unicode en n'utilisant que les caractères Ascii. Ainsi « é » dont le code Unicode est E9 (en base 16 soit 233 en base 10) peut-il être surcodé é ou é ; en HTML.
- ▲ Unicode ne traite que des caractères, pas des glyphes. On ne distingue donc pas un « A » corps 12 d'un « A » corps 8 ou 16, ni un « A italique » d'un « A gras ». Toutefois, pour être compatible avec des codages plus anciens, Unicode a conservé certaines ligatures comme « fi »... Toute spécification de glyphe doit se faire avec un protocole de niveau supérieur (par exemple des balises) dans un fichier.
- ▲ Unicode laisse donc les fontes s'occuper du rendu (de l'impression, de l'affichage, etc.) des glyphes associés à des caractères mais leur fournit certaines informations sur les caractères, globalement (absence de chasse de tel caractère) ou contextuellement (par exemple sens d'écriture).
- ▲ Unicode n'est pas un langage de formatage. Toutes les commandes de mise en page, de transmission, etc. doivent aussi utiliser un protocole de niveau supérieur.

Bref, Unicode n'est vraiment que le mode de remplissage de la partie textuelle « abstraite » d'un document utilisant d'autres commandes pour tout ce qui n'est pas du texte et dont l'utilisateur ne voit souvent que l'effet graphique final.

Orientation bibliographique

- [1] Jacques André, «Caractères, codage et normalisation», *Document numérique*, vol. 6, no 3-4, 2002, p. 13-49 ; http://www.irisa.fr/faqtypo/unicode/chappe_a_unicode.pdf
- [2] Patrick Andries, « Introduction à Unicode et à l'ISO 10646 », *Document numérique*, vol. 6, no 3-4, 2002, p. 51-87, <http://hapax.qc.ca/> ; voir aussi les actes de cette conférence.
- [3] Yannis Haralambous, *Fontes et Codages*, O'Reilly, Paris 2003

PATRIMOINE, FONTES TYPOGRAPHIQUES ET WEB

Jacques André
IRISA/INRIA-Rennes
Campus universitaire de Beaulieu
F-35042 Rennes cedex

La typographie - ne faut-il pas toujours le rappeler ? - n'est pas qu'une question de technique informatique mais aussi d'art et de culture donc d'histoire. Pendant longtemps cette culture est restée l'affaire un peu « privée » des typographes car leur savoir était pratique, oral ou confiné dans des documents difficiles d'accès. Aujourd'hui, grâce au web, cette culture va se généraliser à tout le monde, ce qui ne peut être qu'un bien. Encore faut-il que ce qui est sur le web soit utilisable.

1. MANUELS DE TYPOGRAPHIE SUR LE WEB

On commence à trouver divers manuels de typographie (en langues européennes) sur le web dont ils utilisent plus ou moins les possibilités.

1.1 Mode image seulement

L'ouvrage est scané et les images mises sur le web. Exemples :

- ▲ Geofroy Tory, Champfleury - art et science de la vraie proportion des lettres, 1529 ;

<http://gallica.bnf.fr/scripts/ConsultationTout.exe?O=N050961>

- ▲ Martin Dominique Fertel, La science de l'imprimerie contenant des instructions très faciles pour se perfectionner dans cet art, Saint Omer, 1723
<http://polib.poleuniv-lille-npdc.fr/data/XVIII/II/1/index.html>

L'absence de codage du texte ne permet pas de l'étudier (recherches de mots par exemple). Mais ça permet de diffuser des documents anciens souvent inaccessibles. Ceci permet aussi d'étudier la mise en page, les notices d'imprimeurs et colophons, les planches (notamment de caractères), etc.

1.2 Mode image et mode(s) texte

En plus des pages scannées, le site (ou le DVD) propose une version codée du texte (translittération de la langue ancienne ou traduction en langues modernes). Exemples :

▲ Albrecht Dürer, *Underweysung der messung*, 1505 ;
<http://www.octavo.com/collections/projects/duruwm/index.html>

▲ *Manuale tipografico* del Cavaliere Bodoni Giambattista, 1818.
<http://www.octavo.com/collections/projects/bodtip/index.html>

Ceci permet de lire le texte en langue moderne et éventuellement de faire des recherches sur le texte original.

1.3 Édition savante

En plus du texte et des images, on ajoute des commentaires d'un ou plusieurs spécialistes du domaine, ou des liens sur d'autres ouvrages connexes. Exemple :

Pierre-Simon Fournier, *Manuel typographique*, Paris, 1754

<http://www.irisa.fr/faqtypo/BiViTy/Fournier-Manuel.html>

On arrive ainsi au concept d'hypertexte.

2. PROBLEMES DU CODAGE DES CARACTERES ANCIENS

Si Unicode⁽¹²⁹⁾ permet le codage de tous les caractères « exotiques » (comme ceux dans le Manuel de Fournier), certains caractères « anciens » posent problèmes:

- ▲ Les ligatures qui ne sont pas des caractères pour Unicode,
- ▲ Les abréviations qui ne sont pas non plus des caractères pour Unicode,
- ▲ Les caractères oubliés par Unicode (comme certains caractères de la Pléiade).

Dans les deux premiers cas, on peut utiliser des codages de plus haut niveau⁽¹³⁰⁾, comme notamment ceux pour les manuscrits médiévaux. Dans le dernier cas, il faut se battre pour les faire inclure dans Unicode !

En conclusion, pour qu'un manuel typographique soit utilisable par des typographes sur le web, il faut :

- ▲ Disposer d'images de très haute qualité
- ▲ Que le texte soit scanné puis codé sans pertes d'information (utilisation de telle ou telle ligature et non leur translittération)
- ▲ Qu'une structure hypertexte en permette une lecture « savante » et ouverte

(129) - Patrick Andries, « Introduction à Unicode et à l'ISO 10646 », Document numérique, vol. 6, no 3-4, 2002, p. 51-87, <http://cooptel.qc.ca/%7Eepandries/pdf/intro-Unicode.pdf>; voir aussi les actes de cette conférence.

(130) - Jacques André, « Numérisation et codage des caractères de livres anciens », Document numérique, vol. 7 (2003), n° 3-4, 17 pages.

Le Web Multilingue OU L'INTERNATIONALISATION DU WEB

*Najib Tounsi,
ntounsi@emi.ac.ma, tounsi@w3.org.
Ecole Mohammadia d'Ingénieurs,
Bureau W3C Maroc.*

1. INTRODUCTION

Une société vraiment globale s'appuie pour beaucoup sur l'accès à de l'information qui s'accommode à de multiples cultures et langues. L'internationalisation de l'Internet et des technologies du Web signifie s'intéresser aux différences dans les langues les scripts et les cultures.

Cet article est une introduction à l'internationalisation du World Wide Web, plus communément appelé le Web. On y décrit l'essentiel de ce qu'il faut savoir pour comprendre comment le Web est et sera réellement mondiale.

Le Web est l'univers des informations accessibles à travers le réseau Internet. Ces informations se trouvent dans des documents reliés entre eux par des liens hypertextes, et formatés pour être présentés à un utilisateur qui peut ainsi navigateur d'un document à un autre en suivant les liens. Le Web tire sa force de son universalité. Pour atteindre son potentiel maximum, le Web doit accommoder tout le monde et toutes les cultures. C'est là l'un des objectifs du World Wide Web Consortium, W3C [1], un groupement d'industriels, d'experts et d'universitaires qui collaborent pour créer les standards du Web.

Le Web est constitué de documents, appelés aussi ressources, contenant des informations sous formes de textes ou multimédia. Chaque ressource a une adresse unique dans le Web qui permet de l'identifier. Le contenu d'un document est représenté dans un certain code, stocké dans un ordinateur serveur, transmis selon un protocole et présenté sur un support donné par un client, le plus souvent un navigateur sur un PC, mais ce pourrait être un téléphone portable ou tout autre terminal d'accès. Nous allons aborder cela sous les quatre aspects suivants: le codage des caractères, le formatage des documents, comment indiquer les informations sur l'encodage et les langues d'un document Web et pour terminer l'adressage d'un document.

2. LA REPRESENTATION DES INFORMATIONS MULTILINGUES (CODAGE UNICODE)

Pour être traité par ordinateur, un texte doit d'abord être numérisé. La numérisation consiste à attribuer à chaque lettre ou caractère un nombre, code unique, qui permet de le désigner. Par exemple à la lettre majuscule 'A' est attribué le nombre 64, et à la lettre 'ب' est attribué le nombre 1576.

Historiquement, les ordinateurs parlaient anglais et utilisaient uniquement les caractères latins, les symboles et les signes communs et les chiffres. Il suffisait de peu de valeurs pour codifier tous ces caractères. Les premiers codes, e.g. US ASCII, utilisaient 7 bits sur un octet, soit 128 valeurs de codes possibles [0..127]. Une première extension, est l'usage des valeurs restantes d'un octet [128..255], pour accommoder les caractères d'autres pays. Selon le cas et le pays, une même valeur, 232, peut être un 'è' français ou un ' _ ' grec. La série ISO-8859-n, est le nom donné à ces codes qui ont l'avantage de la simplicité (n = 1 pour les caractères latins occidentaux, n = 6 pour l'arabe, n = 7 pour le grecque etc.). Mais ces codes ont deux inconvénients majeurs: Ils ne peuvent représenter un texte multilingue général autre que anglais+français ou anglais+grecque etc. et surtout, ils ne peuvent couvrir toutes les langues du monde, dont certaines ont plus de 128 signes.

1.1.1.3 Unicode

Unicode est un code universel qui, parce qu'il utilise des nombres sur 16 bits, peut couvrir toutes langues et les systèmes d'écriture au monde. Chaque lettre dans le monde a un code unique (appelé Codepoint) qui lui est définitivement attribué. Unicode est développé conjointement par le Consortium Unicode[2] et l'ISO[2a] (ISO 10646). Pour des raisons de compatibilité historique, les premières valeurs de 0 à 127 sont les mêmes que celles du US ASCII. Les lettres arabes, sont par exemple dans l'intervalle hexadécimal [U+0600..U+06FF]. La valeur hexadécimale 0627 (1276) est le Alif. Unicode utilise la notation U+0627. La langue amazigh est dans l'intervalle [U+2D30 .. U+2DFF].

Les valeurs Unicode (Codepoint) sont des nombres abstraits affectés à des caractères. En tant qu'entiers, ils peuvent être représentés en machine par leur valeur binaire sur 16 bits. Mais cette représentation a quelques inconvénients:

- ▮ Elle dépend du matériel, poids faible d'abord vs poids fort d'abord .Ce qui oblige à introduire des correctifs.
- ▮ Si une majorité de texte est en caractères anglais, il y aura beaucoup de bits à 0 dans les fichiers. La moitié des octets.
- ▮ Internet aime bien les octets ou mots de 8 bits.

1.1.1.4 UTF-8

UTF-8 est une méthode pour représenter en mémoire (et transporter) des caractères Unicode. On parle d'encodage d'un texte. UTF-8 est la plus simple et la plus utilisée d'une série, Unicode Transformation Format, incluant UTF-16, UTF-32.

UTF-8, transforme une valeur Unicode en une séquence d'octets. Les valeurs de 1 à 127 (alias US ASCII) sont mémorisées sur 1 octet. Ainsi UTF-8 est transparent pour les textes en anglais. Les autres valeurs, correspondant aux autres langues du monde, sont représentées sur 2, 3, et parfois jusqu'à 6 octets. Voir [3] pour plus détails.

Les caractères arabes, calés dans l'intervalle [U+0600..U+06FF] sont mémorisées en UTF-8 sur 2 octets.

	Lettre	CodePoint	UTF-8
Alif	ﺍ	U+627	D5-A7
Beh	ﺏ	U+628	D5-A8
Jim	ﺝ	U+62C	D5-AC
Del	ﺩ	U+62F	D5-AF

Table Représentant des lettres arabes Unicode.

Valeur Unicode et paire d'octets UTF-8.

L'information sur l'encodage utilisé dans un document Unicode doit faire partie du document. Sinon, un logiciel ne saura pas comment interpréter le texte mémorisé.

Indication de l'encodage. Exemple de mécanisme permettant de marquer l'encodage d'un texte:

```
en      <meta http-equiv="content-type"    content="text/html;
XHTML  charset=UTF-8">
en CSS2 @charset "UTF-8";
en XML  <?xml version="1.0" encoding="UTF-8" ?>
```

L'encodage d'un document doit se trouver le plus tôt possible dans le fichier. En fait, les caractères qui précèdent cette information sont en générale en US-ASCII.

Unicode/ISO 10646 a permis de définir un Modèle de Caractère pour le Web[3], publié par le W3C, qui est commun aux langages tels que HTML, XHTML, XML, CSS ou RDF. [4]

Toute l'architecture du Web est basée sur le format XML, lui même basée sur Unicode encodé UTF-8.

3. LE FORMAT DE DOCUMENT DANS LE WEB (LANGAGE XHTML)

Les documents présents dans le Web, doivent être formatés pour être traités par les agents utilisateurs. Pour un navigateur Web, le format est défini par le standard XHTML, *Extended Hypertext Markup Language*. XHTML peut être utilisé conjointement avec CSS, qui définit le style de présentation d'un texte.

Un premier exemple en XHTML

```
<p>Ces mots: <span lang="ar-MA" dir="rtl">قط ماهر</span> sont en arabe</p>. (1)
```

Cet exemple en XHTML, montre un paragraphe en français (<p></p>) qui contient un texte en arabe, balisé ici par Ces balises servent ici à marquer le texte auquel on applique les attributs lang et dir, respectivement la langue et la direction d'écriture du texte marqué.

En CSS, le langage des styles, on aurait:

```
: lang(ar-MA) { direction:rtl; font-family:Traditional Arabic; }
```

avec le sélecteur de style : lang() prévu spécialement pour les langues. La propriété 'direction' du texte est définie dans ce sélecteur.

Notre texte devient:

```
<p>Ces mots: <span lang="ar-MA">قط ماهر</span> sont en arabe</p>.
```

On voit aussi l'intérêt d'utiliser les styles pour les langues, car on peut y définir les propriétés de présentation comme la police Traditional Arabic pour les caractères en arabes.

Noter cependant, que si la police ou la couleur est affaire de style, ce n'est pas le cas de la direction d'un texte. C'est pour cela, qu'il est recommandé d'indiquer la direction d'un texte par le balisage en utilisant XHTML, comme en (1).

Remarque: Unicode a prévu aussi des caractères spéciaux (invisibles) qui, pour le cas de l'Arabe, contrôlent la directionnalité d'un texte. Mais il est recommandé de préférer le langage de balisage HTML à ces caractères de contrôle, pour des raisons d'uniformité et surtout parce que les balises sont visibles. Exemple, les deux caractères U+202B (Right to Left Embedding, RLE) et U+202C (Pop Direction Format, PDF) indiquent un début d'affichage droite-gauche et sa fin. L'attribut dir, pour un passage directionnel, a le même effet.

1.1.1.5 Les Balises ou Attributs XHTML Liés à I18N

Meta informations sur l'encodage. Attribut encoding pour des documents XML, balise meta avec charset pour (X) HTML et directive @ pour le langage de style CSS. (voir § indication d'encodage ci-dessus.)

La Langue d'un Document. L'attribut lang dont la valeur est un identificateur de langue (e.g. ar, fr...), éventuellement étendue par un tirait et l'indication de pays (ar-MA, fr-CA ...). Par exemple <html lang="fr-CA"> indique la langue du contenu du fichier XHTML est le français canadien. Les identificateur de langues sont définis par la standard RFC3066 [5] (voir plus loin.)

L'attribut lang sert à introduire aussi bien la langue du document en entier, qu'à indiquer le changement de langue dans une partie du document. On verra plus bas, que les deux usages sont en fait bien distincts.

Indications de la Direction d'un Texte. L'attribut dir indique la direction rtl, droite à gauche, ou ltr, gauche à droite d'une portion de texte (appelée alors passage directionnel). Par défaut, un texte est considéré écrit de gauche à droite. Voir exemple juste ci-dessus. A ce propos, Unicode a défini un algorithme, bidirection algorithm, que les agents utilisateurs utilisent pour rendre un texte bilingue latin et arabe par exemple. Cet algorithme résout de façon transparente la plupart des situations d'affichages bidirectionnelles. Cependant, il est possible de changer le comportement automatique de l'algorithme.

La balise **bdo** (bidi override) permet de spécifier la direction d'affichage du texte désirée. <bdo dir="rtl">exemple de texte</bdo> force l'affichage de son texte de droite à gauche, ce qui donne ici "etxet ed elpmexe".

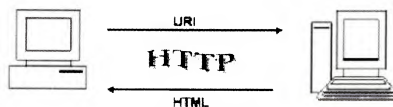
Ruby Notation. La notation Ruby, a à voir avec des systèmes d'écriture d'extrême orient. Elle permet d'annoter un texte par un autre texte en petit caractère et situé juste en dessus par exemple.

4. L'ENCODAGE ET LA LANGUE D'UN DOCUMENT WEB

Mais, en ce qui concerne l'internationalisation du Web, les deux informations les plus importantes pour un document sont l'encodage et la langue de son contenu. L'encodage parce que pour lire un texte un agent utilisateur a besoin de comprendre en quel code il est, et la langue parce que cela permet à des agents évolués d'adapter comme il faut la restitution du contenu à l'audience correspondante.

Nous allons en donner les principes de base et cela sous trois angles: coté serveur du document, coté agent utilisateur et côté protocole d'échange HTTP (*HyperText Transert Protocol*).

Considérons le schéma de transaction type entre un navigateur (l'agent client) agent client et un serveur



Dans ce schéma, le navigateur émet une requête URI, le serveur envoie la page Web en réponse et le navigateur affiche la page reçue. Parfois, le navigateur envoie un formulaire avec des données que le serveur stocke dans une base de données. Mais cela ne change pas fondamentalement les principes de base présentés ici.

1.1.1.6 Négociation d'Encodage et de Langue

Quand un serveur envoie un document à un agent client, il l'accompagne, dans l'entête HTTP de la transaction, de l'information (Content-Type, Content-Language) sur le contenu du document. Typiquement, les paramètres envoyés en entête par le serveur pour accompagner la page Web seront:

Content-Type: text/html; charset=utf-8; Content-Language: fr-CA

qui indiquent que le document servi est un texte en français canadien, formaté en HTML et qu'il est en Unicode encodé utf-8. (Remarque: le mot charset est mal utilisé à ce propos. En XML, on dit encoding.)

Un serveur Web est configurable pour les valeurs à affecter à ces paramètres (sinon c'est ISO-8859-1 et l'anglais) Exemple de configuration du serveur Apache:

```
AddType 'text/html; charset=UTF-8' html
AddLanguage fr .fr
```

pour servir tous les fichiers avec l'extension .html comme utf-8. Si un fichier possède “.fr” dans son extension, il est servi avec Content-Language: fr.

Un agent client peut indiquer aussi dans sa requête la liste des langages et/ou encodages qu'il sait reconnaître.

Accept-Charset: utf-8, iso-8859-1, *;
Accept-Language: ar-MA, fr;

Ici, le navigateur accepte de recevoir les pages Web de préférence en utf-8, mais iso-8859-1 aussi est accepté, ensuite un encodage quelconque. La langue arabe est préférée, ensuite le français. Cette dernière indication est particulièrement utile pour aider le serveur à choisir le bon document dans le cas de plusieurs versions en plusieurs langues d'un même document (e.g. index.html.ar, index.html.fr ...)

1.1.1.7 Informations Méta.

Nous avons vu précédemment qu'on peut déclarer dans la page Web elle-même les informations sur l'encodage du contenu (encoding de XML, charset de HTML). En fait, il est recommandé de toujours le faire, même si cela peut faire double emploi avec la valeur du paramètre "Content-Type" envoyée par dans l'entête HTTP du document. En cas de conflit entre l'encodage caractères du document et l'entête HTTP, c'est cette dernière qui a la priorité.

Remarque: Dans la pratique, les choses sont beaucoup complexes. La transaction HTTP entre le client et le serveur dépend aussi du fichier à envoyer: HTML ou XHTML? XHTML servi comme HTML ou servi comme XML? On peut trouver dans [6] plus de détails sur ce point.

1.1.1.8 Spécification de la Langue du Contenu et Pages Multilingues.

L'information sur la langue d'un document est extrêmement importante pour certaines applications, de plus en plus en voie d'apparition, comme les navigateur vocaux ou les lecteurs d'écran ayant à synthétiser la voix, les correcteurs d'orthographe ayant à déterminer quel dictionnaire utiliser, les traducteurs, ou tout simplement un navigateur ayant à styliser l'affichage d'un texte. Par exemple le navigateur Mozilla applique des fontes adaptées à certaines langues. La fonction lang() de XSLT, permet d'extraire d'un fichier XML les éléments relatifs à une langue donnée. Indiquer la langue d'un texte est aussi une aide pour l'accessibilité, ou la localisation par exemple.

Les auteurs de contenus doivent avant tout déclarer langue d'un document HTML considéré dans sa totalité. L'attribut lang utilisé avec la balise <html> est prévu à cet effet.

```
<html lang="ar-MA"> ... document en arabe ... </html>
```

En XML, c'est xml:lang qu'il faut utiliser, comme attribut de l'élément racine dans ce cas.

```
<?xml...?><MonTexte xml:lang="ar-MA">...document en arabe ...</MonTexte>
```

(Noter cependant qu'en XHTML, il faut utiliser les deux attributs lang et xml:lang

```
<html lang="ar-MA" xml:lang="ar-MA"xmlns="http://www.w3.org/1999/xhtml">
```

Sauf dans le cas où le document est servie en tant que xml, auquel cas seul xml:lang suffit).

Les auteurs de contenus ensuite déclarer le changement dans la langue à l'intérieur d'un document. La méthode est la même que précédemment.

```
<p>En arabe <span lang="ar"> قَط </span> signifie chat.</p>
```

```
<p>En anglais, on dit <span lang="en-EN">cat</span>.</p>
```

En réalité, si on marque la langue d'un document (balise <html>) ou la changement de langue d'un passage de texte de la même façon (attribut lang ou xml:lang), il y a une différence subtile entre les deux cas.

Les identificateurs de langue.

Les identificateurs de langue fr, ar-MA ... sont donnée par le standard RFC3066 [5], basé sur la standard ISO[7] des codes de pays. Le sous-identificateur primaire (fr ou ar dans ar-MA) représente la langue, et la suite éventuelle (-MA dans ar-MA), sert à qualifier le dialecte ou l'usage par rapport à un pays ou une région donnés.

Langue primaire d'un document. L'attribut lang dans la balise <html> est destiné en fait à indiquer la langue du document en tant que tout, c'est à dire la langue de l'audience à qui la page est principalement destinée. On parle de langue primaire dans ce cas. Par exemple l'anglais est la langue de la page d'accueil de yahoo.com et le français celle de yahoo.fr.

Si le document est adressé à une audience multiple, par exemple une page en deux colonnes, une en arabe et une en traduction française, on aura dans ce cas deux langues primaire, déclarées:

```
<html lang="ar, fr" ...>.
```

Langue de traitement. L'attribut lang appliqué à une portion de texte est destiné à indiquer s'il y lieu, un changement de langue et donc mode dans le traitement de cette portion, si la langue le nécessite. On parle de langue de traitement dans ce cas. Les applications mentionnées juste ci-dessus, rentrent cette catégorie.

5. ADRESSES WEB MULTILINGUES (IRI ET IDN)

Une adresse Web est un pointeur vers une ressource sur le Web comme par exemple une page Web.

```
http://www.ircam.ma/index.html
```

Les adresses Web sont des URIs (Uniform Resource Identifiers) définies par le standard RFC3986 [8] et exprimées typiquement en caractères anglais (sous-ensemble de US ASCII en fait.) Cela s'explique pour des raisons historiques. Les ordinateurs et les protocoles sont essentiellement développés en anglais.

Pour être simples à créer, transcrire, mémoriser ou interpréter, les adresses Web sont généralement composées de mots faciles à retenir. Pour les langues autres que l'anglais, il est donc naturel d'utiliser leurs caractères pour exprimer leurs mots dans les adresses, plutôt que de les transcrire en caractères latins (شركة vs Charika).

Des développements récents permettent d'ajouter d'autres caractères à ceux de l'anglais utilisés actuellement dans les URIs. On parle alors de IRI (Internationalized Resource Identifiers).

```
http://عنوان.ويب.ma/
```

```
http://www.w3c.org.ma/في-10-نقط/إكس - إم - في
```

Le développement des IRIs vient en fait comme complément aux URIs. Une IRI est une URI dont les caractères peuvent comprendre des caractères Unicode/ISO10646. Une correspondance est ensuite définie entre une IRI et une URI, ce qui permet d'utiliser une IRI là où on peut utiliser une URI. Cette approche garantit la compatibilité totale avec les logiciels et les applications existantes utilisant des URIs.

Une URI se compose de 3 parties: le protocole "http://", un nom de domaine "www.exemple.com" et le chemin d'accès à la ressource sur le serveur Web désigné "/dossier/page.html". C'est la partie centrale qui est importante pour l'internationalisation. L'écriture http:// ou ftp:// reste pour l'instant telle que, et la plupart des OS actuels gèrent des noms Unicodes.

Pour que les IRIs marchent, il faut: 1) que le format de document accepte des caractères non latins dans les adresses. C'est le cas de HTML, XML, ... 2) Que l'agent utilisateur accepte la saisie de ces caractères dans la zone adresses. 3) Que le protocole d'échange (e.g. HTTP, FTP...) puissent porter des informations de ce type et 4) que le serveur supporte des noms de fichiers en UTF-8 (eg. Linux, MacOS-X, Windows).

1.1.1.9 Noms de Domaines internationalisés IDN

La notion d'IRI, est elle-même basée sur la notion IDN (Internationalized Domain Name)[9]. Exemple ".ma"

Un nom de domaine internationalisé est un nom de domaine qui peut contenir des caractères Unicode/ISO10646. Le standard correspondant est défini dans RFC 3492[10]. L'autorité d'un pays qui enregistre les noms de domaines sous sa responsabilité, doit définir elle-même les caractères permis qu'elle peut gérer. En fait, elle attribue un équivalent en ASCII, appelé punycode, qui sera utilisé dans la résolution de nom. Exemple à عنوان.ويب.ma correspondra xn-mgb9azzbi.xn-ngb7dg.ma. Ce sera aux agents utilisateurs de faire cette transformation IDN vers punycode.

Pour que les IRIs marchent, il faut: 1) que le format de document accepte des caractères non latins dans les adresses. C'est le cas de HTML, XML, ... 2) Que l'agent utilisateur accepte la saisie de ces caractères dans la zone adresses. 3) Que le protocole d'échange (e.g. HTTP, FTP...) puissent porter des informations de ce type et 4) que le serveur supporte des noms de fichiers en UTF-8 (eg. Linux, MacOS-X, Windows).

Les IRIs sont actuellement à leur phase de prototypes. Il existe quelques sites qui marchent avec des IRIs (http://____.w3.mag.keio.ac.jp/ est un site test).

Pour certaines langues, il y a des critères supplémentaires à considérer, e.g. la voyellisation en arabe. Faut-il l'intégrer ou pas? Sinon, comment distinguer

قرآن .com de قرآن.com ? . Il faut aussi définir les critères de comparaisons pour des mots avec des caractères composés.

5. CONCLUSION

Dans ce bref papier nous avons introduit quelques éléments essentiels pour comprendre comment la toile mondiale, le World Wide Web, est réellement mondiale. Ces éléments sont déjà des standards (ou en cours) principalement développés par le *World Wide Web Consortium*, W3C[1], Unicode[2] et l'IETF [11]. Nous avons parlé de Unicode comme moyen de numériser les différents alphabets du monde et des technologies du Web pour créer diffuser de l'information multilingue et accommoder Internet aux différentes cultures.

L'implémentation et l'usage de ces standards ne sont cependant pas encore très généralisés dans la pratique. Mais ces technologies constituent à l'avenir le meilleur moyen pour représenter et manier les données des différentes langues.

Références

- [1] W3C, *World Wide Web Consortium*, (<http://www.w3.org>)
- [2] Le consortium Unicode, (<http://www.unicode.org>)
- [2a] ISO10646, *Universal Character Set ISO*, (<http://www.iso.org>)
- [3] M. Dürst, F. Yergeau, R. Ishida, T. Texin, *the Character Model For the World Wide Web*, recommendation W3C, (<http://www.w3.org/TR/Charmod>)
- [4] W3C XML, *The Extensible Markup Language*, (<http://www.w3.org/TR/XML>)
- [5] RFC3066, *Internet Engineering Task Force*, RFC 3066, (<http://www.ietf.org/rfc/rfc3066.txt>)
- [6] R. Ishida, *Tutorial: Character Sets and Encodings in XHTML, HTML, CSS*. (<http://www.w3.org/International/tutorials/>)
- [7] ISO, *International Standard Organisation*, (<http://www.iso.org>)
- [8] RFC 3987, *Internet Engineering Task Force*, (<http://www.ietf.org/rfc/RFC3987.txt>)
- [9] Michet Suignard, IDN, *International Domain Names and Applications*, (<http://www.verisign.com/>)
- [10] RFC 3492, *Internet Engineering Task Force*, (<http://www.ietf.org/rfc/RFC3492.txt>)
- [11] IETF, *Internet Engineering Task Force*, (<http://www.ietf.org>)

Autres références

<http://www.w3.org/International>

<http://www.i18nguys.com>

<http://www.opentag.com/xmli18nfaq.htm>

LES NORMES DU TRI, DU CLAVIER ET UNICODE

Mohamed OUTAHAJALA

INTRODUCTION

L'évolution des technologies de l'information est en train de modifier en profondeur l'activité humaine. Les nouvelles technologies abolissent graduellement les frontières traditionnelles, créent de nouveaux espaces économiques et culturels, entraînent le développement de nouveaux secteurs d'activité. Cette évolution ne peut cependant masquer une obligation : la nécessité de respecter les langues, les cultures et les usages propres à chaque société. Cet impératif devient encore plus aigu avec l'augmentation rapide du nombre d'utilisateurs des technologies et l'émergence des inforoutes.

Selon un principe reconnu internationalement, tout utilisateur de technologies de l'information doit avoir accès à une information qui respecte les conventions en vigueur dans son pays ou son territoire. En d'autres mots, les technologies ne doivent pas représenter de nouvelles contraintes pour l'utilisateur; elles doivent plutôt offrir de nouvelles possibilités qui permettent à chaque individu de développer ses habiletés et d'exploiter son plein potentiel. Les outils informatiques doivent continuellement être adaptés pour les rendre conformes aux systèmes d'écriture et aux conventions culturelles en usage dans chaque société.

Depuis plusieurs années, des organismes internationaux et nationaux s'emploient à définir des normes et standards pour orienter le développement technologique à l'échelle internationale. L'ISO (Organisation internationale de normalisation) et la CEI (Commission électrotechnique internationale) forment un système spécialisé dans la normalisation internationale. Les comités membres nationaux de l'ISO et de la CEI participent au développement de normes internationales dans le cadre de comités. Le SNIMA (service de normalisation industrielle marocaine) est un membre de l'ISO.

Dans cet article on survolera les principes du standard Unicode, ensuite on discutera des problèmes et solutions du classement des chaînes de caractères et à la fin on présentera la norme internationale de disposition des claviers des équipements informatiques ISO/CEI 9995.

I. Unicode

A. INTRODUCTION A LA PROBLEMATIQUE DU CODAGE

Unicode spécifie un numéro unique pour chaque caractère, quelle que soit la plate-forme, quel que soit le logiciel et quelle que soit la langue. Fondamentalement, les ordinateurs ne comprennent que les nombres. Ils codent les lettres et autres caractères sous formes de nombres. Avant l'invention d'Unicode, plusieurs systèmes de codage de caractères ont été créés. Pas un seul d'entre eux n'était satisfaisant : par exemple, l'Union Européenne a besoin de plusieurs systèmes de codage pour couvrir toutes ses langues d'usage. Même pour une seule langue comme le français, aucun système de codage ne couvrait toutes les lettres, les signes de ponctuation et les symboles techniques en usage courant.

Ces systèmes de codage sont souvent incompatibles entre eux. Ainsi, deux systèmes peuvent utiliser le même nombre pour deux caractères différents ou utiliser différents nombres pour le même caractère. Les ordinateurs, et plus particulièrement les serveurs, doivent supporter plusieurs systèmes de codage de caractères, ce qui crée un risque de corruption des données à chaque transition.

B. UNICODE TRANSFORME TOUT

Unicode spécifie un numéro unique pour chaque caractère, quelle que soit la plate-forme, quel que soit le logiciel et quelle que soit la langue. Le standard Unicode a été choisi par des pionniers technologiques tels que Apple, HP, IBM, Microsoft, Oracle, SAP, Sun, Sybase, Unisys et bien d'autres. Unicode est exigé par de nombreux standards récents tels que XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc. Le développement d'Unicode est synchronisé avec celui de la norme ISO/CEI 10646, la version 4.0 d'Unicode est identique code pour code avec l'ISO/CEI 10646:2003 mais contient de nombreux éléments supplémentaires d'implantation. Unicode est utilisé dans de nombreux systèmes d'exploitation, dans tous les navigateurs récents, et dans de nombreux autres produits. L'apparition du standard Unicode, ainsi que la disponibilité d'outils le gérant, sont parmi les faits les plus marquants de la globalisation récente du développement logiciel.

II. Classement des chaînes de caractères

Pourquoi les codages en soi, les comparaisons caractère par caractère et les programmes de tri commerciaux ne sont-ils pas appropriés au tri et qu'est-ce qui doit être fait pour résoudre ce problème? Dans cette partie on discutera des problèmes de classement des chaînes de caractères et dans un souci de clarté on donnera l'exemple de l'écriture latine.

1. En utilisant le codage de la norme ISO 646, le tri, dans toutes les langues utilisant l'écriture latine, incluant l'anglais, ne produit pas la séquence traditionnelle du dictionnaire, ce qui est le minimum qu'un utilisateur ordinaire peut attendre. Si on trie la liste «*august*», «*August*», «*container*», «*coop*», «*co-op*», «*Vice-president*», «*Vice versa*» nous aurons le résultat suivant : «*August* », «*Vice versa* », «*Vice-president*», «*august*», «*co-op* », «*container* », «*coop* » Ce qui est évidemment erroné.

2. En transformant les minuscules en majuscules et en enlevant les caractères spéciaux, on obtient une liste plus acceptable, bien qu'imprévisible. Par exemple: trier la liste «*August*», «*august*», «*coop*», «*co-op*» donne le résultat suivant: «*August* », «*august* », «*coop*», «*co-op*». Trier la même liste à partir d'un ordre initial différent, soit par exemple, «*august*», «*August*», «*co-op*», «*coop*» donne un résultat différent du précédent, soit: «*august*», «*August*», «*co-op*», «*coop*»

3. Si on inclut les caractères accentués, en utilisant par exemple le codage de la norme ISO 8859-1, les problèmes rencontrés dans les étapes i et ii sont amplifiés bien qu'ils partagent les mêmes causes.

La seule solution possible réside dans la décomposition des données initiales tout en respectant l'ordre lexicographique traditionnel et en assurant une prévisibilité absolue. Ce qui nécessite quatre niveaux de décomposition:

1. Le premier niveau de décomposition permet de rendre l'information à trier indépendante de la casse et des signes diacritiques en plus d'éliminer les caractères n'ayant aucun ordre préétabli dans quelque culture que ce soit.

2. Le deuxième niveau de décomposition différencie les quasi-homographes, chaînes qui diffèrent seulement par leurs signes diacritiques. En français les principaux dictionnaires suivent la règle suivante: les accents sont ignorés lors du tri sauf dans les cas d'égalité homographique. Dans ces cas, la dernière différence détermine l'ordre correct de deux mots, une priorité différente étant attribuée à chaque type d'accent. Par exemple, «*coté*» devrait être placé après «*côte*» mais avant «*côté*».

3. Le troisième niveau de décomposition différencie les quasi-homographes dont la seule différence réside dans la hauteur de casse.

4. Le quatrième niveau de décomposition brise les dernières égalités entre quasi-homographes qui ne diffèrent que par la présence de caractères spéciaux. Cette différenciation est essentielle pour assurer la prévisibilité absolue des tris et pour trier des chaînes composées uniquement de caractères spéciaux.

Ces quatre niveaux de décomposition peuvent être structurés en utilisant une clé à quatre niveaux. Cette clé sera constituée par la concaténation des sous-clés de la plus haute importance à la plus basse.

<K-tamatart> <K-tamatart>;<BASE>;<MIN>;<K-tamatart> ; LETTRE TIFINAGHE YAKW

...

B. LE CLASSEMENT DES CHAINES DE CARACTERES ET LES APPLICATIONS

La relation d'ordre voulue n'est pas nécessairement la même pour toutes les applications. Un certain nombre de caractéristiques débordent de ce que les implantations standardisées offrent, et tombent dans le champ de ce qu'on appelle la classification, par exemple :

- ▀ Plusieurs chaînes de caractères représentent des nombres, ou encore, “ nichent “ des nombres. Plusieurs utilisateurs voudraient voir « appendice a12 » apparaître après, plutôt qu'avant « appendice a9 ». Mais puisque 1 précède 9, la fonction classique de tri « strcmp » fera un choix non intuitif, ici « appendice a12 » sera placé avant « appendice a9 ».
- ▀ Il est parfois requis d'ordonner les chaînes de chiffres comme si leur valeur numérique avait été écrite en toutes lettres.
- ▀ Certaines applications ont besoin de voir l'espacement et la ponctuation prises en considération seulement en dernier lieu, lorsque l'ordre n'est pas établi par ailleurs, et évidemment, l'espacement et la ponctuation ignorés initialement. D'autres applications ont des exigences différentes en ce qui concerne le traitement du trait d'union.
- ▀ Certains utilisateurs voudraient voir quelques mots grammaticaux (e.g. les articles, déterminants, pronoms, prépositions) complètement ignorés, à moins de chaînes par ailleurs identiques.
- ▀ Dans les annuaires de liste de noms propres, on souhaite voir les initiales du prénom associées à ce prénom, indépendamment de leur ponctuation. Un niveau élevé de connaissances du contexte intervient pour déterminer si certains mots sont préfix du nom propre, ou le nom propre lui-même. Par exemple, « Le Blanc » est intégralement un nom propre tandis que « Le Tseung » est un nom propre suivi d'un prénom. L'équivalence abrégative des mots comme « Mc » et « Mac », « Mt » et « Mount » et « St » et « Saint » doit être prise en considération.

C. LE CLASSEMENT DES CHAINES DE CARACTERES TIFINAGHES UTILISANT LES POLICES DE CARACTERES LOCALES

L'application présenté par la suite fait le tri des documents Word utilisant les polices de caractères tifinaghes qui n'utilisent pas la plage de l'Unicode réservée pour le tifinaghe, néanmoins se sont les polices les plus répandues et les plus utilisées. Cette application permet de trier soit tout le document en cliquant sur *Trier* document soit de *Trier* sélection pour trier une partie du document :

◌ΘΖΖΞΕ [abqqid] m. sg. ; Pl. ΞΘΖΖΞΕΙ. E.A. 8ΘΖΖΞΕ ♦ 1. Tache, ♦ 2. Le péché
◌ΧΗ [agl] v. t. ; I. ◌ΧΗ, ◌Χ'Η, II. 8ΧΗ, III. 8ΧΞΗ, IV. ++◌ΧΗ, ++◌Χ'Η. ♦ Pendre,
suspendre. +8ΧΗ +ΓΨΟ+ +οΚΓΓ8Η+ Χ 8Θ◌ΧΗ. (La femme a suspendu la
baratte au crochet). → ◌Θ◌ΧΗ (n. crochet). Passif : ++ϝ◌ΧοΗ, +Π◌ΧΗ (être
suspendu). Récipr. : Γϝ◌ΧοΗ (se suspendre réciproquement).
◌ΗΘ [als] v. t. indirect ; I. ◌ΗΘ, II. 8ΗΘ, III. 8ΗΞΘ, IV. ++◌ΗΘ. ♦ 1. répéter. ♦ 2.
Raconter, narrer. → 1. ◌ΗΗΘ (répétition). Passif : ++ϝ◌ΧοΗ, +Π◌ΧΗ (être
suspendu). Récipr. : Γϝ◌ΧοΗ (se suspendre réciproquement).
◌Γ◌Π◌Η [amawal] n. m. sg. ; Pl. ΞΓ◌Π◌ΗΙ. E.A. 8Γ◌Π◌Η, ♦ Lexique.
◌Γ◌888 [amazuz] adj. et n. m. sg. ; Pl. ΞΓ888 / ΞΓ◌888Ι, Fé.m. +◌Γ◌8888+.
E.A. 8Γ◌888 ♦ 1. Tardif. ◌Θ/Χ◌Ο ◌Γ◌888 (Le maïs tardif). ♦ 2. Benjamin,
cadet, dernier né. ΗΗ◌ Ξ++Γ8/ 8Γ◌888 Λ 8Γ◌Θ ΨΟ +Ξ/ΓΗ. (Le cadet de la
famille va à l'école avec son frère). Ant. : ◌Γ888
ΨΟ [yr] prép. ♦ Vers (orientation). ΞΛΛ◌ ΨΟ +Ξ/ΓΗ. (Il est parti à (vers)
l'école). Syn. Θ. V. Ψ8Ο, Ψ◌Ο, ΨΟ
888 [zil] v. i. ; I. 888, II. 888, III. 888, IV. ++888. ♦ Être beau, bon, agréable.
◌Θ◌888 ◌Λ Ξ888 (ce pain est bon). → 1. +Ξ888+, ◌888Ξ (beauté). Syn. ◌8888,
◌ϝϝ◌, Η888Ξ, Γ◌◌, Θ◌Θ. V. ◌8888, ◌ϝϝ◌, Η888Ξ, Γ◌◌, Θ◌Θ.

On remarque que la liste est triée dans le bon ordre.

Afin de faire ce tri, l'application fait un prétraitement qui consiste en:

- ▶ L'élimination des caractères indésirables tel que le « ^ », en effet si
 l'opérateur de saisie appuie pour une longue durée sur la touche « ^ » ce
 caractère est ajouté au document même si ça n'apparaît pas dans le texte
 écrit en tiffinaghe.
- ▶ Le reclassement des caractères ASCII de sorte à avoir un bon classement
 pour des glyphes tiffinaghes, en affectant des poids appropriés pour les
 caractères tiffinaghes

Le tableau suivant donne les glyphes tiffinaghes, leur correspondance en latin, le code ASCII correspondant et le nouveau poids attribué à la lettre pour avoir un classement correct des chaînes de caractères.

Iskkiln	Correspondance latine	ASCII	Poids affecté
o	a	97	1
⊖	b	98	2
⌘	g	103	3
⌘ ^u	å	229	4
∧	d	100	5
E	ä	228	6
⊗	e	101	7
⌘	f	102	8
⌘	k	107	9
⌘ ^u	æ	230	10
⊖	h	104	11
∧	p	112	12
∩	o	111	13
⌘	x	120	14
∩	q	113	15
⌘	i	105	16
I	j	106	17
∩	l	108	18
∩	m	109	19
I	n	110	20
⊗	u	117	21
O	r	114	22
Q	ë	235	23
∩	v	118	24

IV. Les claviers et méthodes de saisies

Afin d'écrire on utilise un jeu de caractères qui permet de lire du chinois même au Maroc ou du tfinaghe même en Inde (jeu universel de caractères), Cependant on a besoin d'un moyen pour permettre à l'utilisateur de réagir avec le système. L'application de l'ISO/CEI 9995 dans la conception des claviers va fournir une interface uniforme et prévisible entre l'utilisateur et les machines de bureau en divisant le clavier en zones et modules et en affectant des fonctions aux touches. L'une des tâches principales a été de prendre en charge les jeux plus élaborés ou multiples de caractères requis par les diverses applications auxquelles sont destinés les claviers de nos jours. On est parvenu à ce résultat en affectant plusieurs caractères graphiques ou fonctions de commande à chacune des touches d'un clavier, principalement dans le module alphanumérique.

La norme ISO/CEI 9995 : Technologies de l'information - Disposition des claviers conçus pour la bureautique comprend les parties suivantes :

- ▶ Partie 1 : Principes généraux pour la disposition des claviers
- ▶ Partie 2 : Module alphanumérique
- ▶ Partie 3: Dispositions complémentaires de la zone alphanumérique du module alphanumérique
- ▶ Partie 4: Module numérique
- ▶ Partie 5: Module d'édition
- ▶ Partie 6: Module de fonctions
- ▶ Partie 7: Symboles employés pour la représentation de fonctions
- ▶ Partie 8: Affectation de lettres aux touches d'un clavier numérique

Dans ce qui suit nous allons présenter les principes généraux les plus importants pour la disposition des claviers

A. DOMAINE D'APPLICATION

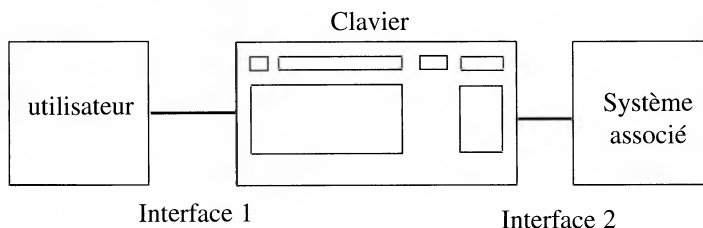
La norme ISO/CEI 9995 spécifie les différentes caractéristiques des claviers d'équipements informatiques par exemple:

- a) les ordinateurs personnels, les stations de travail, les terminaux d'ordinateurs, les terminaux à écran de visualisation (TEV), les machines à écrire, etc., dotés d'un clavier alphanumérique ;
- b) les calculatrices, les téléphones et les guichets automatiques qui possèdent un pavé numérique.

Cette norme spécifie l'affectation des fonctions (caractères graphiques, fonctions de commande ou les deux) aux touches.

B. DIVISION DU CLAVIER

Pour les besoins de la présente norme, le clavier est considéré comme un élément intermédiaire entre un utilisateur et un système de traitement de l'information.



Les fonctions du clavier peuvent se résumer comme suit :

- l'utilisateur active une ou plusieurs touches (interface 1) ;
- les signaux correspondants sont transmis au système de traitement de l'information (interface 2).

Pour les besoins de la présente norme internationale, le clavier est considéré comme étant divisé logiquement en groupes et niveaux, et physiquement en modules et zones.

a. DIVISION LOGIQUE

Les caractères graphiques ou les fonctions de commande auxquels une touche permet d'accéder sont logiquement divisés en groupes et niveaux. La fonction « majuscules » traditionnelle a été étendue pour permettre d'accéder à ces différents groupes et niveaux. Un ou plusieurs mécanismes de sélection permet à l'utilisateur de choisir parmi les groupes et les niveaux disponibles. Deux types de mécanismes sont reconnus dans le cas présent :

- ▲ la sélection de groupe ;
- ▲ la sélection de niveau.

Les deux fonctions peuvent s'appliquer simultanément. Le groupe est hiérarchiquement supérieur au niveau, et plusieurs niveaux peuvent être définis dans un groupe.

Sélection de groupe	Sélection de niveau	Groupe et niveau actifs
Aucune	Aucune	Groupe 1 niveau 1
(Groupe 1 = implicite)	Sélection du niveau 2	Groupe 1 niveau 2
	Sélection du niveau 3	Groupe 1 niveau 3
Oui	Aucune	Groupe n niveau 1
(Vers groupe n	Sélection du niveau 2	Groupe n niveau 2
	Sélection du niveau 3	Groupe n niveau 3

Division logique en groupes et niveaux

Les groupes peuvent contenir des ensembles complets ou distincts de fonctions. Un clavier peut comporter un nombre quelconque de groupes, ce nombre étant seulement limité par des raisons d'ordre pratique. Au sein de chaque groupe, les fonctions (caractères graphiques et/ou fonctions de commande) sont organisées sur trois niveaux au maximum.

b. DIVISION PHYSIQUE

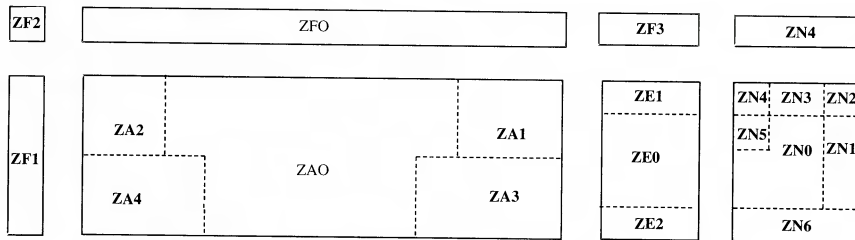
Le clavier est physiquement divisé en modules eux-mêmes divisés en zones dans lesquelles sont disposées les touches. Dans la norme ISO/CEI 9995, les modules du clavier sont identifiés et la forme générale ainsi que la disposition relative des modules sont spécifiées.

Les différentes fonctions pouvant être exécutées par un clavier se divisent en quatre catégories, disposées en quatre modules, comme suit :

- ▶ module alphanumérique, zones ZA0 - ZA4 ;
- ▶ module d'édition, zones ZE0 - ZE2 ;
- ▶ module de fonctions, zones ZF0 - ZF4 ;
- ▶ module numérique, zones ZN0 - ZN6.

Chaque module peut être considéré comme un noyau central (zone 0) entouré d'autres zones permettant de soutenir des touches de fonction ou d'autres touches associées.

La disposition générale des modules et des zones est représentée ci-dessus.

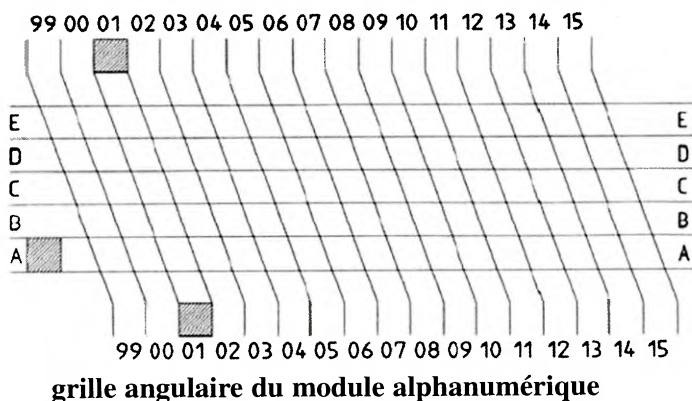


Disposition des modules et zones

C. PRINCIPE DE GRILLES

Le système de numérotation spécifié dans l'ISO/CEI 9995 est associé à un ensemble de diagrammes de disposition, basé chacun sur une grille (intersection de rangées et colonnes). Chaque grille indique l'emplacement relatif des touches dans la disposition d'un module de clavier.

Les rangées sont identifiées par des lettres de A à E, la rangée A correspondant à la barre d'espacement et la rangée E correspondant à la rangée la plus éloignée de l'utilisateur assis devant son clavier. Les colonnes sont identifiées par des chiffres variant de gauche à droite.



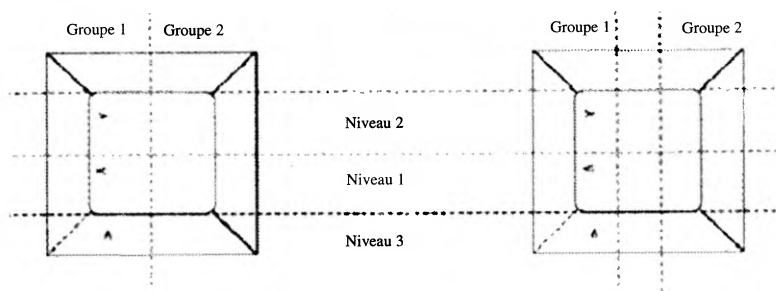
Chacun des autres modules possède sa propre grille.

D. EMPLACEMENT DES SYMBOLES SUR LE DESSUS DES TOUCHES

L'emplacement des symboles correspondant à des fonctions de commande ou à des caractères graphiques affectés à différents niveaux d'un groupe s'effectue à l'aide des différentes rangées de la colonne attribuée à ce groupe. L'ISO/CEI 9995 autorise au maximum trois niveaux à l'intérieur d'un même groupe. Dans cette norme il existe deux scénarios d'emplacement :

1. Présentation des légendes - trois niveaux sur le dessus d'une touche

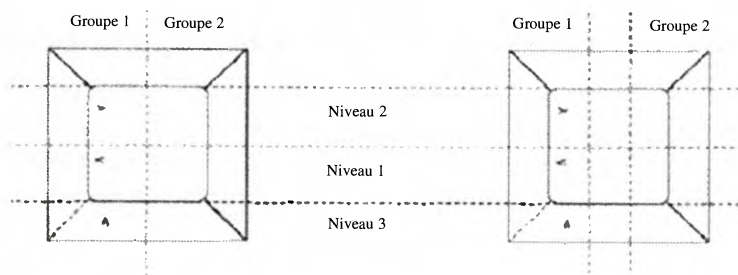
Les symboles représentant l'affectation de niveau 1 doivent être placés dans le tiers central de la colonne du groupe, sur le dessus de la touche. Les symboles correspondant à l'affectation de niveau 2 sont à placer dans le tiers supérieur de la colonne du groupe, sur le dessus de la touche. Les symboles représentant l'affectation de niveau 3 doivent alors être placés dans le tiers inférieur de la colonne du groupe, sur le dessus de la touche. Lorsqu'un même symbole représente les deux premiers niveaux (comme le symbole graphique M qui correspond à la fois à la lettre majuscule M et à la lettre minuscule m), il peut être agrandi et occuper les deux tiers supérieurs de la colonne du groupe, sur le dessus de la touche.



Trois niveaux sur le dessus d'une touche

2. Présentation des légendes - deux niveaux sur le dessus d'une touche, et niveau 3 sur la face de la touche

Les symboles représentant l'affectation de niveau 3 doivent être placés sur la partie avant de la touche, dans la même colonne que les symboles correspondant à l'affectation de niveaux 1 et 2. Les symboles représentant l'affectation de niveau 1 sont à placer dans la moitié inférieure de la colonne du groupe, au dessus de la touche. Les symboles correspondant à l'affectation de niveau 2 doivent être placés dans la moitié supérieure de la colonne du groupe, au-dessus de la touche. Lorsqu'un même symbole représente les deux premiers niveaux (comme le symbole graphique M qui correspond à la fois à la lettre majuscule M et la minuscule m), il peut être agrandi et occuper la moitié supérieure de la colonne du groupe, au dessus de la touche.



Deux niveaux sur le dessus d'une touche, et le 3ème sur la face de la touche

Concernant la taille minimale requise pour les symboles de dessus de touche, les dimensions et surfaces des dessus de touche, l'inclinaison du clavier le déplacement et la force de frappe et la distance entre les touches, toutes ses propriétés sont indiquées dans l'ISO 9241-4.

Conclusion :

Le domaine des technologies de l'information connaît un développement accéléré. Les choses bougent très vite. Les concepteurs, les distributeurs et les diffuseurs doivent se tenir continuellement au fait des dernières avancées, afin d'adapter continuellement les outils qu'ils mettent à la disposition du public. Plusieurs outils de recherche et, en particulier, le réseau Internet offrent maintenant l'accès instantané à l'information et permettent de suivre, presque au jour le jour, l'évolution des travaux nationaux et internationaux de normalisation.

Ces progrès constituent autant de pas vers la mise en place de moyens de communication qui respectent les codes et les pratiques en usage dans chaque société, et qui contribuent à leur épanouissement. Tel est le but qui est visé par ces projets de norme.

Ressources

- ISO/CEI CD 14651 Classement international et comparaison de chaînes de caractères utilisant le jeu universel de caractères codés pour plusieurs octets
- ISO/CEI 9995 - Technologies de l'information - Disposition des claviers conçus pour la bureautique
- http://hapax.qc.ca/ISO_CEI_9995.html

Ouvrages techniques

- LaBonté, A. (Comité Canada-Québec sur le développement du logiciel d'expression française), Guide conceptuel pour la réalisation de logiciels de langue française à portée multilingue, 1987, 23 p.
- LaBonté, A., Règles de classement alphabétique en langue française et procédure informatisée pour le tri, Québec, ministère des Communications, 1988, 6 p.

LES GRAPHES UNL : UN CONCEPT UNIFICATEUR POUR L'INTEGRATION DE L'AMAZIGH DANS DES DOCUMENTS MULTILINGUES

Ali RACHIDI

Ecole Nationale de Commerce et de Gestion

B.P. 37/S hay Salam - Agadir, Maroc

Laboratoire de Traitement d'Images et Systèmes d'Information (LTISI)

Faculté des Sciences, Université Ibn Zohr, Agadir, Maroc

Email : rachidi.ali@caramail.com

Tél. : (212) 62 28 19 75 Fax : (212) 48 22 57 41

Résumé

L'intégration des technologies de l'information et de communication (TIC) à l'apprentissage de la langue Amazigh est absolument nécessaire pour que l'Amazigh ait droit de cité plein et entier sur le Web et dans le monde informatisé.

Dans ce contexte, il est de plus en plus nécessaire de créer des documents multilingues qui intègrent l'Amazigh. La coédition d'un texte en langue naturelle et de sa représentation dans une forme interlingua (IL) semble le moyen le meilleur et le plus simple de partager la révision du texte vers plusieurs langues et :

- ✓ de répercuter du texte et les modifications de L0 vers l'IL,
- ✓ de régénérer vers L1,... Ln depuis l'IL.

Il faudra cependant permettre des améliorations manuelles, car la forme interlingua ne sera pas toujours présente, ou pas assez améliorable par défaut d'expressivité, et les générateurs ne seront jamais parfaits.

Pour plusieurs raisons, les graphes UNL (Universal Networking language) sont les meilleurs candidats. L'idée actuelle est de faire de la traduction manuelle collaborative sur le Web à l'aide d'une mémoire de polyphrases multilingues (MPM) outils en construction par l'équipe de Mr C. Boitet (GETA, CLIPS, IMAG à Grenoble, France), puis d'intégrer le résultat (les phrases en Amazigh) dans le document en UNL-XML. Enfin, il faudra donc construire un déconvertisseur UNL-Amazigh et un enconvertisseur.

Mots Clefs : Communication multilingue, graphe UNL, représentation

Introduction

Ce séminaire s'inscrit dans un large mouvement international qui vise à ce que chaque peuple puisse disposer de tous les moyens pour communiquer dans sa langue. A l'époque, affirmer ou défendre une langue passait par d'autres moyens : fixer une orthographe, construire des dictionnaires monolingues ou bilingues, recueillir des traditions orales ou encore élaborer des polices d'imprimeur.

Aujourd'hui, le développement des ordinateurs personnels et celui des réseaux font de l'informatique un moyen pour écrire et communiquer au même titre que le papier l'est depuis Cai Lun et l'imprimerie depuis Gutenberg. Mais les langues ne sont pas égales devant le processus d'informatisation et les populations parlant des langues mal dotées ont un accès limité à ces nouveaux moyens, limitation pouvant aller d'une simple gêne à une incapacité totale. L'Amazigh fait partie de ces langues peu dotées informatiquement. Par conséquent, des recherches scientifiques et linguistiques sont lancées dans ce sens pour améliorer la situation actuelle. Parmi les volets, qui occupent une position prioritaire, est de concevoir et réaliser des documents multilingues qui intègrent la langue Amazigh. En effet, et dans la pratique courante, un document multilingue consiste en plusieurs fichiers monolingues parallèles, qui peuvent être de la documentation technique aussi bien que des fichiers d'aide, des fichiers de messages, ou simplement de l'information thématique mise sur la toile et destinée à une audience multilingue (médecine, cuisine, chanson, traditions...).

La tâche est difficile, même pour un document géré de façon centralisée ([1] [2]). D'habitude, on le crée d'abord dans une unique langue source, puis on le traduit dans plusieurs langues cibles (au moins une trentaine dans le cas de firmes comme IBM, HP ou Microsoft). Il faut avoir un moyen de garder trace des modifications, qui peuvent parfois être faites en différents lieux sur des versions en différentes langues. Dans ce cas, des traducteurs humains doivent retraduire les segments nouveaux ou modifiés, ou les réviser s'ils sont traduits par un système de TA de qualité. Ce que nous souhaiterons, c'est faire en sorte que le travail de révision puisse être partagé entre les langues, quels que soient le domaine et le contexte.

Il est clairement impossible de refléter les changements sur un fichier en langue L0 dans les fichiers en langues L1,... Ln automatiquement et fidèlement, sans une structure intermédiaire pour faire le pont, car il faudrait au moins un aligneur parfait à granularité très fine dans le cas simple d'un changement d'article ou de nom.

L'approche la meilleure et la plus simple nous semble être d'utiliser un interlingua formel IL et :

- ▶ • de répercuter les modifications de L0 vers l'IL,
- ▶ • de régénérer vers L1,... Ln depuis l'IL.

Il faudra cependant permettre des améliorations manuelles, car la forme interlingua ne sera pas toujours présente, ou pas assez améliorable par défaut d'expressivité, et les générateurs ne seront jamais parfaits.

On choisit UNL ([3] [4] [6] [7]) comme interlingua pour différentes raisons :

- ▶ il est spécialement conçu pour le traitement linguistique et sémantique par ordinateur ;
- ▶ il a été dérivé avec beaucoup d'améliorations du langage pivot de H. Uchida utilisé dans ATLAS-II de Fujitsu ([8]), toujours évalué comme le système de TA anglais-japonais de meilleure qualité, avec une très grande couverture (586.000 entrées par langue) ;
- ▶ les participants du projet UNL⁽¹³¹⁾ ont construit des "déconvertisseurs" d'UNL vers environ 12 langues, parmi elles, on cite l'arabe, l'indonésien, l'italien, le français, le russe, l'espagnol et le thaï;
- ▶ bien qu'ils soient de nature formelle, les graphes UNL (voir ci-dessous) sont assez simples à comprendre avec peu de formation et peuvent être présentés de façon localisée à des utilisateurs "naïfs" en traduisant les symboles (relations sémantiques, attributs) et les lexèmes du langage UNL par des symboles et des lexèmes de leur langue ;
- ▶ le projet UNL a défini un format "UNL-html" intégré à html pour des fichiers contenant un document multilingue complet aligné au niveau des énoncés, et a produit un "visualiseur" qui transforme un fichier dans ce format en autant de fichiers html que de langues, et les envoie à n'importe quel navigateur Web.

La suite est organisée comme suit. D'abord, on présentera l'UNL comme langage de représentation unifié. Ensuite, on exposera le format UNL-HTML comme l'un des formats des documents multilingues stockés sur un serveur sous forme d'une collection de fichiers multilingues. Enfin, on essaye de donner une idée sur ce qui sera le déconvertisseur Unl-Amazigh et l'enconvertisseur associé en passant sur un certain nombre de concepts et définitions.

(131) - <http://unl.ias.unu.edu>

Langage UNL (Universal Networking Language)

1. Définition

L'NUL est l'acronyme de ' Universal Networking Language'. C'est un langage artificiel qui réplique, dans le Web, les fonctions des langages naturels dans la communication humaine. Il permet aussi aux ordinateurs du monde entier de inter communiquer, et enfin, avoir une infrastructure linguistique pour distribuer, recevoir et comprendre les informations multilingues. Par conséquent, il permet aux différentes populations du monde d'exprimer toutes les connaissances communiquées par les langages naturels

2. Caractéristiques

L'UNL représente trois éléments : un projet international, un langage artificiel et un format de document multilingue.

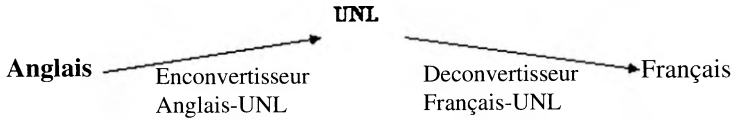
2.1. Le projet

Le projet UNLP consiste à mettre en place un système de traduction automatique universel. Il est défini comme un projet de "métalangage numérique" pour l'encodage, le stockage, la recherche et la communication d'informations multilingues indépendamment d'une langue source - et donc d'un système de pensée.

Il est mené sous l'égide de l'Université des Nations Unies (UNU, Tokyo). Plusieurs équipes réparties dans le monde travaillent sur ce projet (Japon, Inde, France, Espagne, Italie, Chine, Russie...). L'intérêt de cette collaboration est de travailler avec les mêmes règles et codes et d'obtenir ainsi un résultat vraiment universel, c'est-à-dire compréhensible par toutes les cultures.

2.2. Le langage artificiel

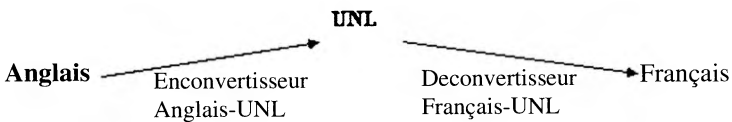
L'UNL est un langage compris par les ordinateurs. Il est universel, c'est à dire que toute langue naturelle (parlée par des êtres humains) peut être traduite en UNL. De même, l'UNL peut être traduit en toute langue naturelle. Pour passer d'une langue naturelle à l'UNL, on utilise un enconvertisseur et pour passer de l'UNL à une langue naturelle, on utilise un deconvertisseur.



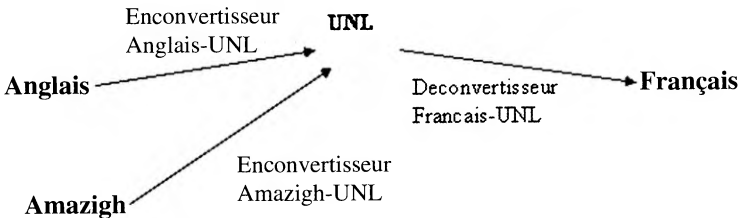
Principe de base de la traduction par UNL

- ▶ Un enconvertisseur permet le passage d'une langue naturelle à l'UNL
- ▶ Un deconvertisseur passe de l'UNL à une autre langue naturelle

Avantages :

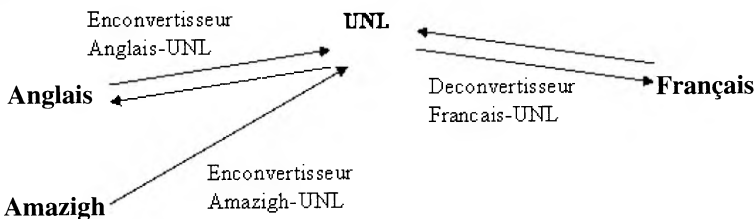


Grâce à un enconvertisseur anglais-UNL et un deconvertisseur UNL-Français, on peut traduire automatiquement de l'Anglais au Français.



L'ajout d'un seul enconvertisseur Amazigh-Unl permet de passer de l'Amazigh au français

(Ainsi, une équipe Amazighienne n'a pas besoin de connaître le français pour faire progresser la traduction Amazigh-Français).



Enfin, deux équipes, française et anglaise, peuvent faire avancer les traductions Français-Anglais et indirectement Amazigh-Anglais. Et si on a le deconvertisseur

UNL-Amazigh, la communication entre l'Amazigh, le Français et l'Anglais sera complète.

Tout travail national aide à la communauté internationale.

2.3. Le format de documents multilingues

Un même document UNL contient les textes en plusieurs langues ainsi qu'en UNL. Cependant, l'utilisateur ne verra que les langues le concernant, par exemple la langue source et la langue traduite. Le document est balisé, c'est-à-dire que des groupes de mots sont entourés de chaînes de caractères représentant un code compréhensible par les ordinateurs (comme <html> par exemple). On traitera UNL-HTML en détail dans la prochaine section.

3. Principes

L'UNL est composé de mots (Universal Words ou UW), de relations définies entre les mots et d'attributs. Les UW représentent le vocabulaire. Les relations entre UWs et les attributs représentent la syntaxe et la sémantique (donner un sens).

Comme l'anglais est connu par tous les développeurs informatiques, les UW sont définies par des termes ou des abréviations anglaises. Cependant, ceci n'est qu'une règle de commodité, les UWs pourraient être définies numériquement.

Les développeurs UNL francophones ont féminisé UW, car le terme français le plus voisin est "unité lexicale".

3.1. Unités lexicales (UWs)

3.1.1. Définition

Les UWs représentent donc le vocabulaire de l'UNL. En d'autres termes, chaque UW représente un concept ou un ensemble de concepts, un concept correspondant à une idée précise. Une UW est formée d'un mot anglais suivi d'une liste de contraintes, qui précise le sens de l'UW.

La syntaxe d'une UW est :

<UW> := <Entrée> [liste des contraintes>] (entrée est un mot anglais)

Exemples :

- ▶ Le mot Français « danse » existe au moins sous deux concepts, "je danse" ou "j'aime cette danse".

Dans le premier cas, il sera défini ainsi :

- dance(icl>do) c'est-à-dire comme un verbe d'action, défini par l'attribut "do" (faire en anglais).

Dans le second cas, il est défini :

- dance(icl>thing) c'est-à-dire comme un nom commun définissant un objet.
- ▀ Le mot Amazigh « tawada » (TAWADA) sera défini : walking(icl>thing) et pas seulement « walking » puisque le mot désigne plusieurs choses.

En augmentant le nombre d'attributs, on définit très précisément le sens de chaque UW. Exemple, le terme “marquer” pour marquer un but dans un match de football est défini : score(icl>event,agt>human,fld>sport). Ce mot est un verbe d'événement (icl>event), l'objet qui agit, l'agent, est un être humain (agt>human), le champ concerné est le sport (fld>sport). On remarque que dans la définition, le sens du mot est donné ainsi que les autres types de mot avec lesquels il peut interagir. Ceci pour éviter des phrases du genre : “Le chien a marqué un but de la main.”.

3.1.2. Types des UWs

Il existe six types :

- ▀ UW de base est une entrée (mot en anglais) sans contraintes. Elle signifie tous les concepts correspondants en anglais. Elle est utilisée pour structurer la base de connaissances et comme moyens pour établir les correspondances entre les mots de différentes langues une fois que des correspondances spécifiques n'existent pas. Exemple : go, take, state, ...
- ▀ UW restreinte est une entrée suivie d'une liste de contraintes. Exemple: 1) state(icl>express), 2) state(icl>country), 3) state(icl>government), etc.
- ▀ UW importée est une entrée qui n'est un mot anglais mais importée pour pouvoir exprimer des phrases en d'autres langues. Exemple : 1) samba(icl>dance), 2) (apidus) : ahidous(icl>dance).
- ▀ UW temporaire est une entrée qu'on ne doit pas intégrer dans la base de connaissances ou dans le vocabulaire des UWs. Exemple : formule, nom propre, etc.
- ▀ UW composée est une entrée temporaire, mais elle est un ensemble de relations binaires regroupées pour exprimer un concept complexe. Exemple : « women who wear big hats in movie theaters (icl>person) » peut être décomposée en plusieurs UWs.
- ▀ UW nul est une chaîne de caractères vides pour exprimer l'absence d'un argument nécessaire d'une relation. Elle peut être considérée comme temporaire. Exemple : « Med was killed ». le sujet est absent. Pour exprimer l'idée, on utilise : 1) agt(kill(agt>thing, obj>thing)).@entry .@past, ‘ ’) 2)obj(kill(agt>thing,obj>thing)).@entry.@past, Med.@topic)

3.2. Relations entre les UWs

Les relations entre les UWs sont définies par des chaînes de trois caractères ou moins. Ce sont des relations binaires (entre deux UWs) et elles permettent de préciser quelle UW peut être utilisée avec quelle UW. L'UNL traite les textes phrase par phrase; ainsi les relations entre les UWs sont définies à chaque phrase. Une phrase est représentée sous la forme d'un graphe ou d'une liste.

Exemple : la phrase « I ran in the park yesterday » sera représentée comme suit:

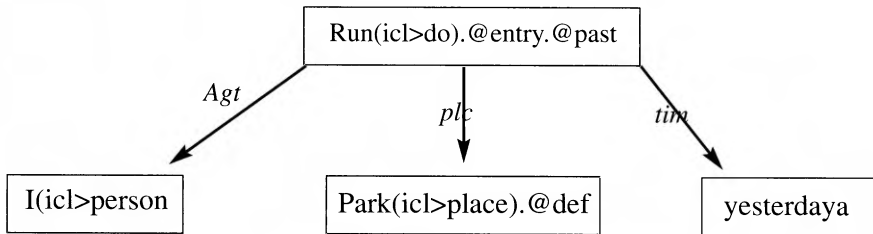
Relations :

Agt(run(icl>do).@entry.@past,i(icl>person))

Plc(run(icl>do).@entry.@past,park(icl>place).@def)

Tim(run(icl>do).@entry.@past,yesterday)

Graphe



Une phrase UNL est centrée autour du verbe principal de la phrase, ici “run”. Les relations entre les mots sont ensuite définies à partir de ce verbe.

Les ensembles en gras sont inclus dans la définition des UWs, indépendamment de la phrase (exemple : “run(icl>do)”, le mot “run” est défini comme un verbe d’action par le code “do” (“faire” en anglais)). Les UWs utilisées ont des attributs :

@entry signifie que l’UW est une entrée ou une UW principale de la phrase ;

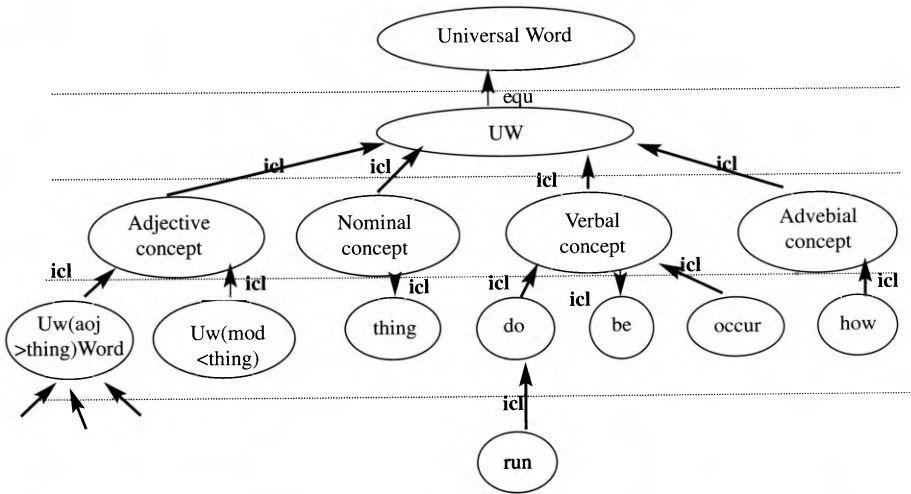
@past signifie que l’UW est au passé ;

@def signifie que l’UW est défini (the park).

Les éléments en italique représentent les relations entre les mots dans la phrase (Agt, tim, plc)

Ce graphe sera attaché à une base de connaissances UNL (UNL KB) qui contient les définitions et les différentes catégories de mots d’une langue naturelle. Par exemple, dans l’exemple précédent, on utilise le mot « do » (entrée de base) qui est un élément de la base UNL KB. Cette base est un graphe sémantique composé de toutes les relations binaires directes entre UWs. La structure générale de l’hierarchie des UWs dans la base UNL KB, qui est sous forme de niveaux, est la suivante ⁽¹³²⁾ :

(132) - <http://www.undl.org>



Après avoir détaillé les principes de langage UNL, on passera pour présenter le format d'un document multilingue UNL-HTML comme signalé précédemment.

Format et document UNL-HTML

1. balises et syntaxe

Le document UNL [5] a le format suivant :

<document UNL>	::= "[D:" <dinf> "]" { "[P:" <nombre> "]" { "[S:" <nombre> "]" <phrase> "[/S]" }... "[/P]" }... "[/D]"
<dinf>	::= <nom de document > "," <nom de propriétaire> ["," <id. de document > "," <date> "," <E-mail >]
<nom de document >	::= "dn=" <chaîne de caractères>
<nom de propriétaire>	::= "on=" <chaîne de caractères>
<id. De document >	::= "did=" <chaîne de caractères> /* définit par le système*/
<date>	::= "dt=" <chaîne de caractères> /* définit par le système*/
<E-mail>	::= "mid=" <chaîne de caractères> /* définit par le système*/
<phrase>	::= "{org:" <l-tag> ["=" <code>] "}" <source de la phrase> "{/org}" "{unl" [":" <uinif>] "}" <expression de UNL> "{/unl}" "{<l-tag> ["=" <code>] [":" <sinif> "]" <phrase générée> "{/" <l-tag> "}"
<l-tag>	::= "ab" "cn" "de" "el" "es" "fr" "id" "hd" "it" "jp" "lv" "mg" "pg" "ru" "sh" "th" "am" /* symboles des langues*/
<code>	::= <chaîne de caractères>
<source de la phrase>	::= <chaîne de caractères>
<phrase générée>	::= <chaîne de caractères>
<uinif>	::= <nom de système> "," <nom d'éditeur poste> "," <fiabilité> ["," <date> "," <E-mail >]

<sinf> ::= <nom de système> “,” <nom d’éditeur poste> “,” <fiabilité>
 [“,” <date> “,” <E-mail>]
 <nom de système> ::= “sn=” <chaîne de caractères>
 <nom d’éditeur poste> ::= “pn=” <chaîne de caractères>
 <fiabilité> ::= “rel=” <entier>

La signification des balises utilisées est la suivante:

[D:<dnf>] Indique le début d’un document
 [/D] Indique la fin de document
 [P:<nombre>] Indique le début d’un paragraphe
 [/P] Indique la fin d’un paragraphe
 [S:<nombre>] Indique le début d’une phrase et son numéro
 [/S] Indique la fin d’une phrase
 {org:<l-tag
 =<code>} Indique le début d’une phrase d’origine, le
 langage et le code peuvent être omis.
 {/org} Indique la fin d’une phrase d’origine
 {unl:<uinf>} Indique le début d’une expression d’UNL
 associée à une phrase ou information
 nécessaire, “:<uinf>” peut être omis.
 {/unl} Indique la fin d’une expression d’UNL d’une
 phrase.

2. Exemple

<HTML><HEAD><TITLE>

Exemple EL/UNL

</TITLE></HEAD><BODY>

[D:dn = Séminaire CEISIC sur UNL, on = UNL francais, mid =
 premier.auteur@here.com]

[P]

[S:1]

(1) {org:el} I ran in the park yesterday. {/org}

(2){UNL}

agt(run(icl>do).@entry.@past, i(icl>person))

plc(run(ixl>do).@entry.@past, park(icl>lance).@def)

tim(run(icl>do).entry.@past, yesterday)

{/UNL}

(3){de dtim = 20020130-2035, deco = man}

Ich life gestern im park. {/de}

(4){es dtim = 20040130-2031, deco = UNL-SP}

Yo corri ayer en el parquet. {/es}

(5){fr dtim = 20040131-0805, deco = UNL-FR}

j’ai couru dans le parc hier. {/fr}

(6){am dtim = 20040131-0900, deco = UNL-AM}

```
rulv asnat g tmardult . {/am} [/S]
[S:2]
.... [/S] [/P] .... [/D]
</BODY></HTML>
```

- (1) texte d'origine
- (2) traduction en UNL
- (3) traduction allemande
- (4) traduction espagnole
- (5) traduction française
- (6) traduction amazighe

Les balises permettent à l'utilisateur de choisir ce qu'il désire visualiser. Si un utilisateur ne souhaite voir que la langue source et la traduction Amazigh, l'ordinateur ne rendra visible que les textes encadrés par les balises {org:el}{/org} et {Am dtime}{/Am}, ce qui donnera :

```
I ran in the park yesterday.
rulv asnat g tmardult
```

Enconvertisseur et déconvertisseur UNL-Amazigh

Chaque équipe de l'UNL travaille sur le déconvertisseur et enconvertisseur de sa propre langue, en augmentant le nombre d'UWs et en précisant les relations entre les UWs. C'est en augmentant le nombre de phrases analysées que les progrès de la traduction automatique se réalisent. Toutes les UWs sont regroupées dans une base de données, ainsi que les relations possibles entre les UWs. La base de données d'UWs sera donc terminée quand tous les concepts mondiaux seront entrés dans cette base, c'est-à-dire jamais... En effet, la langue évoluant sans cesse, de nouveaux concepts apparaissent tous les jours. Le but de l'UNL n'est pas de réaliser une traduction parfaite mais compréhensible, avec le moins d'ambiguïté; le réel danger d'une traduction étant l'ambiguïté, mais non la faute grammaticale ou d'orthographe.

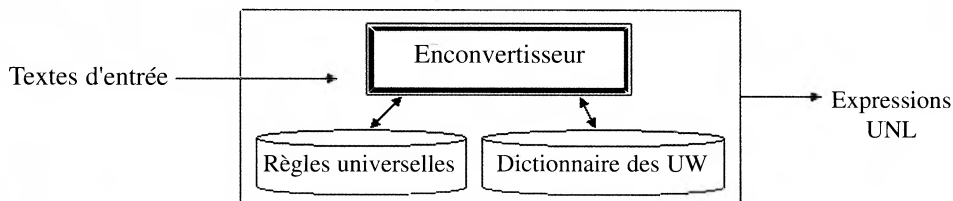
1. Enconvertisseur Amazigh-UNL

L'enconversion n'est pas (si on veut de la qualité pour du tout venant) une analyse classique. C'est une méthode de fabrication de graphes UNL qui suppose une bonne part d'interaction, avec plusieurs possibilités :

- ◆ analyse classique multiple suivi d'une désambiguïstation interactive en langue source
- ◆ entrée sous langage contrôlé,
- ◆ encore plus séduisant (et encore pas clair, au niveau recherche pour l'instant), entrée directe via une interface graphique reliée à la base lexicale (dictionnaire) et à la base de connaissances.

En effet, c'est un analyseur indépendant de langues, qui fournit une structure pour une analyse morphologique, syntaxique et sémantique simultanée en utilisant un ensemble de règles universelles et le dictionnaire des UW. Le processus d'un enconvertisseur amazigh futur est le suivant :

Chaque texte d'entrée est scanné de gauche à droite. Par la suite, chaque morphème correspondant aux initiales de texte est extrait de dictionnaire des UWs et il devient morphème candidat. Les règles universelles (grammaticales) seront appliquées sur ces morphèmes selon l'ordre de priorité pour construire l'arbre syntaxique et le réseau sémantique de la phrase. Le résultat de ce processus est un réseau sémantique exprimé dans le format UNL. Le schéma suivant illustre la structure d'un enconvertisseur Amazigh-UNL.



Structure d'enconvertisseur Amazigh

Ce qu'il faut faire d'une manière explicite, c'est de définir tous les mots Amazigh dans le dictionnaire des UW et d'intégrer les règles grammaticales et sémantiques de l'Amazigh dans la base des règles universelles.

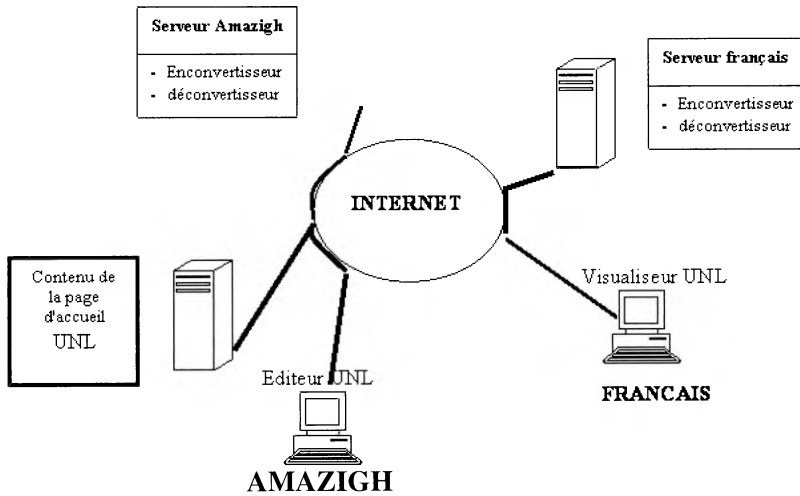
2. déconvertisseur UNL-Amazigh

Le déconvertisseur est un générateur indépendant de langues naturelles qui fournit une structure pour une génération morphologique et syntaxique. Il peut déconvertir les expressions UNL à une variété de langues naturelles, en utilisant un certain nombre de données linguistiques telles que le dictionnaire, les règles grammaticales et le dictionnaire co-occurrence de chaque langue. Choses qui restent à concevoir et à réaliser par les chercheurs de la langue Amazigh.

Une question se pose : comment UNL peut être utilisé via l'Internet? Pour répondre à cette question, on expliquera le processus dans la section suivante qui détaille le système UNL.

Système UNL

Le système UNL consiste en un ensemble de serveurs, d'éditeurs et de visionneurs UNL-langues naturelles. Prenons l'exemple d'un éditeur Amazigh et d'un visualiseur français comme illustré dans la figure suivante:



Structure de système UNL

Lorsqu'on développe une page Web en Amazigh, l'éditeur UNL reconnaît le contenu comme Amazigh et envoie le contenu au serveur Amazigh pour enconvertir le texte. Lorsque le texte est enconverti en UNL, le serveur Amazigh envoie le résultat à l'éditeur UNL. Les développeurs de page Web peuvent ensuite intégrer UNL dans leur page web.

Lorsque le visualiseur français essaie de lire cette page Web, il reconnaît le format UNL-HTML et envoie le contenu au serveur Français pour une déconversion en français qui se charge d'envoyer par la suite le résultat au visualiseur français.

Le serveur Amazigh contiendra deux fichiers : le premier est un dictionnaire qui liste la correspondance entre les UW et les mots de l'Amazigh. Le deuxième contiendra les règles grammaticales de l'Amazigh.

Conclusion

Le meilleur moyen de promouvoir la culture et la langue Amazigh a été d'adapter les techniques d'informations et de communication à l'apprentissage de la langue. Informatiser l'Amazigh demande un effort de plusieurs chercheurs et l'utilisation des leviers permettant d'obtenir rapidement des logiciels de qualité. Dans ce contexte, les documents multilingues resteront un moyen de diffusion du savoir Amazigh afin de valoriser cette culture sur le Web et dans le monde informatisé. En effet, il est nécessaire de développer la plate forme logicielle pour intégrer l'Amazigh dans les documents multilingues. Le meilleur moyen est de développer un serveur Amazigh qui intègre l'enconvertisseur et le deconvertisseur UNL puisque l'UNL est un système de traduction automatique universel soutenu par communauté internationale.

Références bibliographiques

- [1] Al Assimi A.-B. (2000) Gestion de l'évolution non centralisée de documents parallèles multilingues. Nouvelle thèse, UJF, Grenoble, 31/10/00, 200 p.
- [2] Al Assimi A.-B., & Boitet C. (2001) Management of Non-Centralized Evolution of Parallel Multilingual Documents. Proc. of Internationalization Track, 10th International World Wide Web Conference, Hong Kong, May 1-5, 2001, 7 p.
- [3] Blanc E. (2001) From graph to tree : Processing UNL graph using an existing MT system. Proc. of First UNL Open Conference - Building Global Knowledge with UNL, Suzhou, China, 18-20 Nov. 2001, UNDL (Geneva), 6 p.
- [4] Boguslavsky I., Frid N., Iomdin L., Kreidlin L., Sagalova I., & Sizov V. (2000) Creating a Universal Networking Language Module within an Advanced NLP System. Proc. of COLING-2000, Saarbrücken, 31/7-3/8/2000, ACL & Morgan Kaufmann, 1/2, pp. 83-89.
- [5] Christian BOITET, TSAI Wang-Ju, La coédition langue_UNL pour partager la révision entre les langues d'un document multilingue : un concept unificateur, Traitement Automatique des langues Naturelles TALN'2, Nancy, 24-27/06/2002 P. 275-286.
- [6] Sérasset G., & Boitet C. (1999) UNL-French deconversion as transfer & generation from an interlingua with possible quality enhancement through offline human interaction. Proc. of MT Summit VII, Singapore, 13-17 September 1999, Asia Pacific Ass. for MT, pp. 220-228.
- [7] Sérasset G., & Boitet C. (2000) On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter. Proc. of COLING-2000, Saarbrücken, 31/7-3/8/2000, ACL & Morgan Kaufmann, 2/2, pp. 768-774.
- [8] Uchida H. (1989) ATLAS. Proc. of MTS-II (MT Summit), Munich, 16-18 août 1989, pp. 152-157.

CE QUE VOUS DEVEZ SAVOIR AU SUJET DU BIDI ET DU BALISAGE AU SEIN DES BLOCS

Richard Ishida, W3C
Traduit par: Patrick Andries

Ce tutorial décrit d'abord quelques principes de base de l'algorithme bidirectionnel. Il se penche ensuite sur quelques cas de figure fréquents où l'algorithme bidi a besoin d'information complémentaire qu'il faut fournir à l'aide de balises ou de caractères de commande.

Bien que ce tutorial essaie d'être neutre en matière de balisage, la plupart des exemples utilisent XHTML car ce langage est sans doute le mieux connu. Pour des conseils sur un langage de balisage particulier, consulter les notes marginales.

Définitions

Les **textes bidirectionnels** sont courants dans les écritures droite-à-gauche comme l'arabe, l'hébraïque, la syriaque et la thâna. De nombreuses langues s'écrivent à l'aide de ces écritures.

Toute incise de texte d'une écriture gauche-à-droite ainsi que tous les chiffres s'écrivent de gauche à droite au sein d'un texte écrit globalement de droite à gauche. (Le texte français de ce tutorial comprend, bien sûr, également du texte bidirectionnel quand il contient des exemples arabes ou hébreux.)

Nous utiliserons le terme **bidi** dans le sens de « bidirectionnel ». De même, nous utiliserons **DàG** et **GàD** comme formes abrégées de « droite-à-gauche » et « gauche-à-droite » respectivement.

Ordre visuel et ordre logique

L'ordre visuel des caractères s'utilisait couramment pour représenter l'hébreu en HTML dans les anciens navigateurs qui ne mettaient pas en œuvre l'algorithme bidi d'Unicode. Dans une certaine mesure, il persiste encore aujourd'hui, par inertie en ordre visuel, le premier caractère stocké en mémoire est le caractère qui apparaît à l'extrême gauche de l'écran.

Prenons un texte hébreu à directionnalité mixte

W3C, _____

Le tableau ci-dessous illustre les représentations en mémoire des caractères de ce même texte codé, d'une part, en ordre logique et, d'autre part, en ordre visuel. L'ordre de stockage représente également l'ordre de saisie ; l'hébreu visuel devait donc être tapé en commençant par la fin logique du texte !

Ordre logique	Ordre visuel
05E4 _ LETTRE HÉBRAÏQUE PÉ	0057 W LETTRE MAJUSCULE LATINE W
05E2 _ LETTRE HÉBRAÏQUE AÏN	0033 3 CHIFFRE TROIS
05D9 _ LETTRE HÉBRAÏQUE YOUD	0043 C LETTRE MAJUSCULE LATINE C
05DC _ LETTRE HÉBRAÏQUE LAMÈD	0020 ESPACE
05D5 _ LETTRE HÉBRAÏQUE WAW	002C , VIRGULE
05EA _ LETTRE HÉBRAÏQUE TAV	05DD _ LETTRE HÉBRAÏQUE MÉM FINAL
0020 ESPACE	05D5 _ LETTRE HÉBRAÏQUE WAW
05D4 _ LETTRE HÉBRAÏQUE HÈ	05D0 _ LETTRE HÉBRAÏQUE ALEF
05D1 _ LETTRE HÉBRAÏQUE BÈT	05E0 _ LETTRE HÉBRAÏQUE NOUN
05D9 _ LETTRE HÉBRAÏQUE YOUD	05D9 _ LETTRE HÉBRAÏQUE YOUD
05E0 _ LETTRE HÉBRAÏQUE NOUN	05D1 _ LETTRE HÉBRAÏQUE BÈT
05D0 _ LETTRE HÉBRAÏQUE ALEF	05D4 _ LETTRE HÉBRAÏQUE HÈ
05D5 _ LETTRE HÉBRAÏQUE WAW	0020 ESPACE
05DD _ LETTRE HÉBRAÏQUE MÉM FINAL	05EA _ LETTRE HÉBRAÏQUE TAV
002C , VIRGULE	05D5 _ LETTRE HÉBRAÏQUE WAW
0020 ESPACE	05DC _ LETTRE HÉBRAÏQUE LAMÈD
0057 W LETTRE MAJUSCULE LATINE W	05D9 _ LETTRE HÉBRAÏQUE YOUD
0033 3 CHIFFRE TROIS	05E2 _ LETTRE HÉBRAÏQUE AÏN
0043 C LETTRE MAJUSCULE LATINE C	05E4 _ LETTRE HÉBRAÏQUE PÉ

L'ordre visuel n'a jamais vraiment été utilisé en arabe. Sans doute la cursivité des lettres arabes qui sont toutes reliées a-t-elle poussé les développeurs arabes à privilégier la méthode de stockage en ordre logique.

L'ordre visuel d'un texte courant oblige l'auteur à désactiver le repli automatique de lignes, à insérer manuellement les passages à la ligne et à aligner à droite le texte situé à l'intérieur des éléments de type bloc et des cellules de tableau. Il faut ensuite coder le texte dans un codage qui évitera l'exécution de l'algorithme bidi d'Unicode dans les navigateurs récents. Voici un exemple écrit en HTML :

```
<table width="50%"><tr><td align="right" nowrap>
,INRIA- _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ W3C<br> W3C- _ _ _ _ _ _ _ _ _ _
.ERCIM- _ , _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ <br>
_ _ _ _ _ ,INRIA _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ <br>
.2003 _ _ _ _ _ 1 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ .ERCIM
</td></tr></table>
```

(Cet exemple est en fait assez propre. On trouve parfois des choses moins présentables comme des paragraphes alignés à droite dont chaque ligne est entourée de balises <nobr>.</nobr>. Si la fenêtre d'affichage est trop étroite, le début de chaque ligne, à droite de l'écran, disparaît.)

L'ordre visuel produit un code très fragile et difficile à entretenir. Ainsi, outre la difficulté de taper de l'hébreu à reculons, si l'on veut ajouter quelques mots à la deuxième ligne de ce paragraphe, il faudra réajuster tous les passages à la ligne subséquents. Il faut également ajouter (et ensuite entretenir) du balisage supplémentaire pour les hyperliens ou les mises en exergue dont le texte s'étend sur plus d'une ligne.

Note: L'ordre visuel peut également poser des problèmes à un niveau supérieur car il force, par exemple, l'inversion manuelle de l'ordre des colonnes d'une table quand on traduit son contenu dans une autre langue. Il faudra également réajuster les passages à la ligne si la géométrie de la page devait changer. Etc.

L'ordre logique est une bien meilleure méthode car le texte est stocké en mémoire dans l'ordre où il est normalement saisi (et prononcé). L'algorithme bidirectionnel d'Unicode réordonne le texte pour l'afficher dans le bon ordre.

La création de longs paragraphes de texte courant qui se replie automatiquement en fonction de la largeur de l'élément bloc qui les contient devient alors aisée. Il est alors nettement plus facile d'utiliser des outils comme les lecteurs d'écran.

L'algorithme bidi opère sur du texte stocké en ordre logique. Si vous préférez utiliser l'ordre visuel, vous pouvez tout de suite arrêter de lire ce tutorial (bien que sa lecture pourrait vous faciliter grandement la vie)

Fonctionnement de l'algorithme bidi

Nous allons maintenant introduire quelques concepts de base importants. Si cela vous semble fastidieux, prenez patience et persévérez car, sans une bonne compréhension de ces concepts, vous serez perdu quand il vous faudra écrire des textes bidi balisés.

Propriété directionnelle des caractères

Nous avons déjà vu qu'une suite de caractères latins est rendue (c.-à-d. affichée) de gauche à droite (comme cette page en est témoin). Par contre, l'algorithme bidi rendra une suite de caractères fortement DàG (arabes par exemple) en les affichant de droite à gauche.

Ceci ne fonctionne que parce qu'Unicode associe à chaque caractère une propriété directionnelle. La plupart des lettres ont un type directionnel GàD fort, c'est le cas des lettres latines, grecques ou cyrilliques par exemple. Les lettres des écritures bidirectionnelles ont, par contre, un type DàG fort.

Quand du texte au sein d'un élément bloc mélange plusieurs directionalités, l'algorithme bidi affiche chaque suite contiguë de caractères de même directionnalité

sous la forme d'un passage directionnel distinct. L'exemple ci-dessous comprend trois passages directionnels :

bahreïn مصر koweit

L'ordre d'affichage des passages directionnels dépend de leur contexte général. Pour l'exemple ci-dessus, dont le contexte général est GàD puisque il s'inscrit dans un texte en français, on lira d'abord «bahreïn» puis « مصر » (DàG) et enfin «koweit». Aucun balisage ou stylage n'est nécessaire pour obtenir ce résultat.

Contexte directionnel

Le résultat de l'algorithme bidirectionnel dépend du contexte directionnel global du paragraphe, du bloc ou de la page auquel il s'applique.

En XHTML, si on ne précise aucun attribut dir sur la balise html, ce contexte directionnel est alors implicitement GàD. En revanche, en ajoutant dir="rtl" à la balise html, tous les éléments du document héritent d'un contexte DàG. L'utilisation de l'attribut dir sur un élément bloc particulier permet de modifier le contexte directionnel en vigueur dans cet élément.

Caractères neutres

Les espaces et les signes de ponctuation n'ont pas de formes GàD ou DàG dans Unicode car ils peuvent être utilisés avec n'importe quelle écriture. C'est pourquoi on les appelle des caractères neutres.

C'est ici que les choses deviennent intéressantes. Quand l'algorithme bidi rencontre des caractères à propriété directionnelle neutre (comme les espaces et la ponctuation), il détermine la suite des choses en fonction des caractères voisins.

Un caractère neutre situé entre deux caractères à forte directionnalité DàG sera traité comme un caractère DàG, ce qui prolonge le passage directionnel. C'est pourquoi les trois mots arabes de la phrase GàD suivante se lisent de droite à gauche (on lit d'abord le mot arabe مفتاح, puis معايير et enfin الويب).

Le titre est مفتاح معايير الويب en arabe.

Remarquez qu'à ce stade tout balisage ou stylage est superflu. L'exemple ci-dessus ne comprend toujours que trois passages directionnels.

Les choses se corsent quand une espace ou un signe de ponctuation se trouve entre deux caractères à directionnalité forte différente. Le caractère neutre prend alors la directionnalité globale du paragraphe ou du contexte. Si plusieurs caractères neutres se trouvent entre les deux caractères de types forts différents, on associe à tous la même directionnalité résultante.

Ceci crée donc une frontière entre des passages directionnels.

Quand l'algorithme a besoin d'un petit coup de pouce

Dans la majorité des cas, l'algorithme bidi traitera parfaitement les textes sans recours à du balisage ou à tout autre mécanisme, en autant qu'on aura précisé la directionnalité globale du document. Mais il est peu probable, toutefois, que vous vous en tiriez toujours à si bon compte.

Neutres mal placés

Ajoutons un signe de ponctuation à la fin de la phrase arabe. En absence de balisage, on verra

Le titre est « **امفتاح معايير الويب** » en arabe.

Les guillemets sont bien placés, mais le point d'exclamation est à la mauvaise place. Il devrait apparaître à la fin du texte arabe, c'est-à-dire à sa gauche. Le titre est **امفتاح معايير الويب** en arabe. Grâce à notre compréhension de l'algorithme bidi, il est facile de comprendre pourquoi ceci se produit. En effet, le point d'exclamation se trouvant en mémoire entre la dernière lettre DàG « _ » (à gauche) et la lettre GàD « e » (du mot « en »), c'est le contexte global (ici GàD) qui détermine sa directionnalité. Remarquons qu'il importe peu que l'on trouve à cet endroit deux caractères de ponctuation et deux espaces, car ils sont tous neutres et sont donc affectés de la même façon. Le point d'exclamation étant considéré GàD, il se joint au passage directionnel qui comprend le texte « en arabe ».

Comment faire alors pour remettre la ponctuation à sa place ? Dans ce genre de situation, vous aurez sans doute entouré la citation arabe de balises pour la désigner comme une citation ou pour y adjoindre une information linguistique. Dans ce cas-ci, il existe une solution simple: ajouter un attribut à la balise pour en préciser la directionnalité DàG.

Voilà à quoi cela pourrait ressembler en XHTML :

```
Le titre est «&nbsp;<span dir="rtl" lang="ar" xml:lang="ar">!
مفتاح معايير الويب</s en arabe.
```

Observez bien que la balise span se trouve à l'intérieur des guillemets car ces derniers font partie du texte français.

Une autre solution serait de faire suivre le point d'exclamation d'un caractère invisible à forte directionnalité DàG. De la sorte, le point d'exclamation serait interprété comme DàG et ferait dès lors partie du passage directionnel arabe.

Or, il se trouve qu'Unicode contient un tel caractère - le caractère U+200F, appelé MARQUE DROITE-À-GAUCHE (MDAG). Sa contrepartie GàD existe également, il s'agit de U+200E MARQUE GAUCHE-À-DROITE (MGAD).

Ce caractère étant invisible, il vaut mieux le saisir sous la forme d'un appel de caractère numérique (‏) ou - si possible - d'un appel d'entité (par exemple

‏ en XHTML). Dans l'exemple suivant, on a ajouté ‏ après le point d'exclamation, le résultat est alors correct : Le titre est « مفتاح معايير الويب ! » en arabe.

Neutres en pleine crise d'identité

Dans notre prochain exemple, l'ordre de la liste est incorrect car les deux premiers mots arabes devraient être intervertis et la virgule qui se trouve entre eux et qui fait partie du texte français devrait apparaître à la droite immédiate du premier mot.

Les noms de ces États en arabe sont respectivement le مصر والبحرين et الكويت
Le résultat escompté était :

Les noms de ces États en arabe sont respectivement الكويت و مصر والبحرين

Ce résultat inattendu s'explique par la présence de chaque côté de la virgule de caractères à forte directionnalité droite-à-gauche (DàG). En présence de ces caractères, l'algorithme bidirectionnel considère la virgule neutre comme faisant partie du texte arabe alors qu'elle fait partie du texte français et devrait marquer la frontière entre deux passages directionnels arabes.

Alors que, dans la section précédente, le caractère neutre pensait faire partie du contexte global et qu'il n'en faisait pas partie, dans ce cas-ci le neutre pense faire partie du passage contra-directionnel alors qu'il fait partie du contexte global ! Personne n'a dit que la vie était simple...

Une solution élégante consiste à insérer à côté de la virgule un autre caractère Unicode invisible, cette fois-ci il s'agit de la MARQUE GAUCHE-À-DROITE. Ceci place notre virgule entre deux caractères à directionnalité forte, l'un DàG et l'autre GàD, ce qui la force à prendre la directionnalité du contexte global, c'est-à-dire le flot GàD français. Ceci sépare les mots arabes pour former deux passages distincts qui sont affichés en ordre GàD conformément à la direction dominante du paragraphe.

À nouveau, il se peut que vous préféreriez utiliser un ACN (‎) ou un appel d'entité

(comme ‏) si possible.

Entourer la virgule de balises dans ce cas-ci reviendrait à écraser une mouche avec un marteau.

Encore une fois, tous ensemble !

Les exemples que nous avons montrés jusqu'à présent étaient en français et avaient donc une directionnalité globale GàD. Les mêmes principes s'appliquent aux textes DàG écrits dans des langues comme l'hébreu ou l'arabe. Un exemple supplémentaire nous aidera à mieux comprendre.

Voici ce qu'on voudrait voir, dans un paragraphe dont la direction de base est réglée à DàG

.ERCIM - _____) (World Wide Web Consortium) W3C

Malheureusement, sans aide l'algorithme bidirectionnel crée un véritable fouillis:

.ERCIM - _____ (W3C (World Wide Web Consortium

On peut croire qu'il sera extrêmement compliqué de régler ce gâchis, mais en fait la solution est des plus simples. Il suffit d'insérer une MDAG après « W3C » et le tour est joué. C'est vraiment aussi simple !

Si vous n'êtes pas convaincu, voici l'explication. La MDAG après « W3C » sépare ce morceau de texte GàD du passage directionnel entre parenthèses qui le suit. Souvenez-vous que les passages directionnels s'afficheront ici de droite à gauche, qui est la direction générale du paragraphe. Le « W3C » ayant été saisi en premier lieu, il apparaît le plus à droite. La parenthèse se trouve maintenant entre des caractères de forte directionnalité différente et prend donc la directionnalité globale du paragraphe. Elle suit donc. Puis vient le passage directionnel GàD continu, c'est-à-dire tout ce qui se trouvait à l'intérieur des parenthèses.

Note: Afin de mieux comprendre la suite de caractères des exemples ci-dessus, il vaudrait peut-être la peine de déplacer le curseur à travers la source de ce texte dans un éditeur plutôt que de simplement examiner l'écran.

(Le changement de glyphe de la parenthèse située le plus à droite est automatique. Le glyphe utilisé pour ces « caractères miroités » se transforme automatiquement en fonction de leur directionnalité. Il s'agit cependant du même caractère.)

Emboîtement des passages directionnels

L'algorithme bidi d'Unicode et les marques directionnelles font l'affaire quand on n'est en présence que d'un seul niveau de texte à directionnalité contraire (appelé **parfois palier directionnel**). En présence de deux ou plus de deux niveaux imbriqués (paliers) de texte directionnel, une autre solution s'impose. Mais d'abord un exemple de texte mal ordonné :

Le titre est « _____, W3C » en hébreu.

L'ordre des deux mots hébreux est correct, mais le terme «W3C» devrait apparaître du côté gauche de la citation et la virgule devrait se trouver entre le texte hébreu et « W3C ». En d'autres mots, le résultat escompté est :

Le titre est « W3C, _____ » en hébreu.

Note : Dans certaines versions des navigateurs de Netscape et Mozilla, le quatrième élément de la deuxième rangée apparaîtra sous la forme de 1234. Les deux chaînes représentent le même nombre. Notez que Mozilla 1.4 considère à nouveau la forme européenne des chiffres comme l'œil implicite à utiliser.

Ceci ne se produit que dans des textes DàG.

Complicé ? Ne vous en faites pas, d'habitude l'algorithme bidi s'occupera de tout pour vous. Je n'ai inclus cette section que pour ceux qui remarqueraient la différence et se demanderait il ce qui se passe.

Enfin, notez que d'autres caractères neutres accolés à un nombre, comme les symboles de devises monétaires, sont considérés comme faisant partie du nombre plutôt que comme des neutres.

Désactiver l'algorithme

Il peut arriver que vous vouliez empêcher l'algorithme bidi d'effectuer son travail de réordonnement sur un passage de texte. Il faut alors entourer ce passage de balisage supplémentaire.

En XHTML 1.0, on utilise pour ce faire l'élément `bdo`. En XHTML 2, l'élément `bdo` sera probablement remplacé par les valeurs `rlo` et `lro` ajoutées à l'attribut `dir`, ce qui permettra de l'appliquer à n'importe quel élément. À nouveau, il existe des caractères de commande Unicode qui permettent d'obtenir le même résultat, mais puisqu'ils créent des régions aux frontières invisibles, ils ne sont pas recommandés.

Les exemples de cet article qui montrent l'ordre des caractères en mémoire utilisent la balise `bdo` pour obtenir cet effet. Ainsi, pour illustrer la suite de caractères correspondant à :

W3C, _____

il suffit d'utiliser le balisage suivant en XHTML 1.0 :

```
<p><bdo dir="ltr">W3C, _____</bdo></p>
```

Le résultat, écrit de gauche à droite, est :

_____, W3C

Pour approfondir

- ❖ Unicode et ISO 10646 en français <http://hapax.qc.ca>
- ❖ Bidi techniques in HTML (en dévt)
- ❖ <http://www.w3.org/International/geo/html-tech/tech-bidi.html>
- ❖ The Unicode Bidirectionnel Algorithm <http://www.unicode.org/reports/tr9/>
- ❖ Non-Latin scripts tutorial <http://people.w3.org/rishida/scripts/tutorial/>

❖ Other W3C I18N resources relating to Bidirectional text

<http://www.w3.org/International/resource-index.html#bidi>

Ce document est une traduction. En cas de divergences ou d'erreurs, la **dernière version originale** en anglais fait autorité. Comme indiqué ci-dessous, les **droits d'auteur** reviennent au W3C.

Traduit de la version anglaise datée du 30 janvier 2004. La traduction a été modifiée pour la dernière fois le 2005-02-18 14:50 TU.

Version anglaise en-ligne:

<http://www.w3.org/International/articles/inline-bidi-markup/Overview.en.html>

Version française en-ligne:

<http://www.w3.org/International/articles/inline-bidi-markup/Overview.fr.html>

Copyright (c) 2005 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.

<http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>.

Tables Rondes

L'après midi de la deuxième journée du Workshop « La typographie entre les domaines de l'art et de l'informatique » s'est caractérisée par la tenue de deux tables rondes selon le planning suivant :

15h -16h : « Aspects typographiques et écriture cursive de l'alphabet amazighe »

16h30- 17h30 : « Langue amazighe et technologie de l'information et de la communication »

Afin d'être objectif, nous avons essayé d'être fidèles, dans la mesure du possible, à l'esprit du débat en reproduisant pratiquement telles quelles les questions et réponses soulevées.

Table ronde sur les « Aspects typographiques et écriture cursive de l'alphabet amazighe »

Coordinateur : Frank Jalleau

Rapporteur : Youssef Ait Ouguengay

Mr. Franck Jalleau, le coordinateur de la séance, a ouvert le débat sur la cursivité de l'écriture amazighe en exprimant le souhait de voir des interventions de pédagogues amazighes au cours de la table ronde.

Il revient sur la question de la cursivité pour donner son opinion. Il signale qu'il faut tout d'abord un travail d'analyse des cahiers des élèves. Selon lui, l'élève passe par deux étapes dans son éducation :

1. apprendre et s'approprier la langue : acte de mémorisation,
2. l'enfant accélère sa façon d'écrire, l'écriture se personnalise et ensuite elle devient fluide; c'est à ce moment que ça devient très intéressant pour le travail d'analyse du typographe.

Mr. Lahbib fouad, qui a une grande expérience, dans l'animation des ateliers de créations artistiques pour enfants, a observé la façon dont les enfants écrivent leurs noms. Il a relevé que le phénomène de la cursivité n'a jamais été observé chez un enfant. Selon lui, les enfants écrivent exactement comme dans le manuel. Par contre, leurs façons de résoudre les problèmes de l'écriture sont très intéressantes. La première remarque est que la taille des lettres n'est pas très intéressante pour eux (apparence de la descendante, et de l'ascendante). Il a fait remarquer qu'il faut beaucoup de temps pour étudier les écritures des enfants.

Mr. Lahbib relate son expérience avec des adultes analphabètes dans les ateliers de tissage des motifs de décoration. Il a trouvé que les décorations ont une grande similitude avec les caractères tifinaghes et que les gens trouvent une grande facilité pour la représentation des caractères tifinaghes (noms en tifinaghe) comme décoration des tissus.

Mr. Jalleau est revenu ensuite sur le facteur « temps » et a signalé qu'il ne faut pas anticiper et qu'il faut se donner le temps nécessaire. Pour accélérer le processus, il suggère d'injecter des contraintes d'écritures au fur et à mesure, mais il ajoute que cette injection doit se faire en douceur et que toute erreur est dangereuse pour l'évolution de l'écriture.

Mr Andries a demandé si une méthodologie graphique est utilisée. Il a fait entendre que jusqu'à maintenant il n'y a pas de leçon d'écriture, mais des leçons de la langue.

Mr. Jalleau intervient pour ajouter qu'il faut une période de transition pour régler le « ductus ». Pour éclaircir son idée, il a signalé que les tableaux de caractères exposés lors du séminaire ne sont que de la stylistique et de la fantaisie. L'écriture pour lui est quelque chose de plus sérieux. Il faut, pour lui, oublier qu'un texte est écrit en Garamond ou autre ; il faut que la reconnaissance des caractères soit la plus naturelle possible et sans distinction du style. Ce dernier détail vient par la suite.

Un intervenant du public présent lors de cette séance, signale que Mr. Jalleau met beaucoup l'accent sur l'outil papier alors que le Tifinaghe s'adapte à des supports plus larges que le papier.

Il faut donc passer par la phase d'apprentissage le plus longtemps possible ensuite, selon Mr. Jalleau, il faut une phase de collecte puis une phase de classement des collectes.

Mr. Logié nous a fait part de son expérience avec ses élèves. Il a fait des essais avec ses élèves en leur faisant approprier les signes et les sons du Tifinaghe. Il suit pour cela une démarche d'art plastique, qui est nécessaire quand on se ramène à la typographie. Il reconnaît que les élèves ne sont pas entrain d'apprendre l'écriture mais la typographie. Ceci constitue, pour lui, un point de départ !

Mr. Tounsi explique qu'il faut laisser les lettres se former et « s'échapper » afin de donner des formes stables et un « ductus ». C'est après cela, que vient l'étape de classement.

Mr. Yacoubi s'est demandé : c'est quoi l'élément moteur qui peut amener une collecte et un classement dans tous le Royaume ?

Selon Mr Jalleau, ceci dépend de la puissance d'organisation existante. La question peut être résolu par correspondance et par création de projets universitaires fédérés par des comités d'organisation.

Mr. Zenkouar soulève le point sur ce qu'il faut collecter, et signale qu'il faut rassembler tous les manuscrits : ce qui est bon et ce qui ne l'est pas, et que les croisements entre ses différentes collectes formeront le noyau de départ pour une écriture plus cursive. Mr. Jalleau a ajouté que ceci doit se faire à une condition nécessaire : c'est de ne pas tomber dans le piège de l'artificiel.

Mr. Ishida pose deux questions :

1. les enfants sont-ils les meilleurs candidats ?
2. est-ce que ceci va couler dans le Tifinaghe actuel ?

Mr. Jalleau répond:

1. la réponse évidente de la première question est pour lui non, car pour lui ce ne sont pas les élèves qui vont inventer le « modèle » cursive, il faut pour cela des spécialistes ; pour lui il faut se poser des questions pédagogiques pour que le modèle soit pédagogiquement correct et accepté, ce qui constituera un travail de professionnels,
2. Sa réponse pour la deuxième question est que l'influence sur l'écriture de Tifinaghe vient d'abord de la part des maîtres qui seront dirigés au fur et à mesure dans leurs travaux avec les enfants.

Mr. Jalleau a fait remarquer qu'il y a plus de 4000 polices de caractères latines. Les polices de caractères, pour lui, dépendent amplement du contexte social. Pour lui, l'évolution de la société et de l'écriture ne font qu'un. Comme exemple, il évoque le cas différent, du chinois et de l'hébreu dans les écritures manuscrites et imprimées. Mais les imprimés actuels ne sont pour lui que de la typographie (Titrage, fontes fantaisistes,...).

Mr Andries a soulevé le problème de la lecture de Tifinaghe : il a fait allusion au fait qu'en latin les polices de caractères destinés à l'édition du texte sont différents de ceux utilisés pour l'affichage publicitaire, etc. Mr. Andries s'est posé la question pour le Tifinaghe : est-ce que l'écriture « bâton » actuelle du Tifinaghe ne dérange pas en lecture ?

Mr. Lahbib a répondu que ce n'est qu'une question d'habitude, et que ces aspects d'écritures dépendent de la culture. Mr. Lahbib a ajouté que dans le cas du Tifinaghe, au contraire du latin, les formes de lettres sont jolies et fluides dans la lecture.

Mr. Jalleau est revenu pour dire que la typographie de la graphie n'est pas un problème, le vrai problème est dans l'écriture elle-même.

Mr. Zenkouar a signalé ensuite que le Tifinaghe est déjà très bon et satisfaisant dans son état actuel, et que le besoin actuel est en termes de polices de caractères et de créations typographiques.

Mr. Jalleau a ajouté que pour ce faire, il faut le caractère adéquat pour chaque utilisation et a indiqué qu'un panneau d'affichage n'est pas comme un livre !

Mr. Logié a posé les questions suivantes:

Quelles polices et pourquoi ?

Avec quels locuteurs, on a à faire ?

Mr. Logié a proposé une démarche pour la quête d'une écriture amazighe cursive, en effet, il a proposé de :

1. travailler sur un « cahier des normes » pour le dessin des caractères, qui raconte la réalité et l'histoire de chaque caractère et définit les grandes lignes des caractéristiques de l'écriture amazighe.
2. faire un « cahier de pré-connaissances », ce qui constitue un travail de typographes professionnels.

Mr. Jalleau a ajouté que le travail de typographe se situe entre la culture et l'industrie, et que son travail est de fabriquer la matière première et puis à en extraire l'essence.

Mr. Ishida a clôturé le débat avec la remarque suivante : le caractère doit respecter la culture de la société concernée, et il faut laisser le choix aux propriétaires de la langue.

NB :

Le ductus : En calligraphie, l'ordre précis des mouvements successifs de la main pour réaliser les différentes séquences du tracé de chaque lettre.

Ascendante : Partie verticale des longues du haut ; on parle aussi de haste, hampe, ou jambage supérieur.

Descendante : Partie verticale des longues du bas, descendant en dessous de la ligne du pied ; on parle aussi de jambage inférieur.

- Table ronde sur les « Langue amazighe et technologies d'information et de communication »

Coordinateur : Richard Ishida
Rapporteur : Mohamed Outahajala

Monsieur Ishida entame la table ronde par la présentation du sujet « Langue amazighe et technologies d'information et de communication ». Ensuite il lance le débat en posant les questions suivantes :

Qu'est ce qui manque pour que les technologies montent au Maroc? En particulier pour la culture amazighe ? Est-ce un UNL(1) amazighe ou autre?

- Mr Zenkouar souligne qu'après l'introduction des tfinaghés dans le système universel de codage Unicode, d'autres normes, telle que les normes du tri et du clavier suivront. Cependant il faut que les grandes firmes prennent en considération ces normes. Mr Zenkouar note également que le CEISIC (centre des études informatiques et des systèmes d'informations) travaille actuellement en collaboration avec le SNIMA (Service de Normalisation Industrielle Marocaine) pour mettre au point les différentes normes en relation avec les technologies de l'information. Pour terminer son intervention il pose la question suivante:

Quels sont les problèmes qui peuvent se poser pour les pages Web amazighes?

- Mr Andries propose à l'IRCAM d'amorcer le processus d'édition en ligne d'ouvrages en tfinaghés.
- Mr Rachidi mentionne qu'il faut cibler la grande population et pose la question sur les dispositions à mettre en place pour cet objectif ? Est-ce le Web, les CD multimédia, une autre solution?
- Mr Tounsi, répond à la question concernant le Web. Il explique que le Web est basé sur XML(2) et que ce dernier est basé sur Unicode. Par conséquent, il fait remarquer qu'il n'y aura pas de problèmes pour les pages Web tfinaghés écrites en Unicode ainsi que pour toutes les applications qui utilisent ce codage.
- Mr Andries met en relief le fait que les applications viennent après le codage des caractères Unicode.
- Mr Ishida, pour sa part, met le point sur la nécessité de prendre des contacts à l'échelle internationale avec Microsoft et avec tous les distributeurs de logiciels en général et les logiciels libres en particulier.

- Mr Zenkouar informe qu'il est actuellement en contact avec les responsables de Microsoft Afrique du nord pour intégrer l'amazighe à leurs produits.
- Mr Bououd de l'Université de Fès a souligné qu'il faut toucher également l'oral dans cet effort d'accès de l'amazighe aux nouvelles technologies.
- Mr Houssa Yacoubi soulève la problématique du passage de l'oral à l'écrit, alors que la société est analphabète.
- Mr Rachidi soulève, pour sa part, la question des stratégies à suivre et des expériences similaires afin de cibler le grand public.
- Mr Boudar donne l'exemple de la mission « alif lam mim » (émission de la première chaîne nationale) comme exemple de mission télévisée ciblant le grand public.
- Mr Paput explique que la télévision et la radio sont des outils non fiables car elles s'appuient sur d'autres critères, en particulier publicitaires et économiques.
- Mr Rachidi ajoute que la télévision manque d'interactivité.
- Mr Ishida demande aux présents si Unicode est enseigné aux universités et aux écoles.
- Sur ce point, Mr Andries propose de mettre des cours d'Unicode sur le site de l'IRCAM afin d'aider les développeurs à adopter ce codage et à l'utiliser.
- Mr Logié explique qu'il y a un besoin de formateurs en Unicode et donne l'exemple de l'école des beaux arts de Casa.
- Mr Zenkouar informe qu'un projet sur la vulgarisation d'Unicode est en cours et annonce également qu'un projet sur la cursivité est en cours.
- Mr Andries affirme qu'après la 1ère ébauche du février 2004, les membres de l'Unicode ont jugé nécessaire d'ajouter les autres variétés du tfinaghe, néanmoins pour le clavier chaque pays peut adopter son clavier national. Mr Andries suggère comme projet une application de recherche exploitant les entrées lexicales.
- Mr Rachidi propose d'intégrer l'amazighe dans le projet Papillon(3) et fait remarquer que ce projet nécessite d'avoir les catégories grammaticales des lexies.
- Mr Bououd mentionne qu'il y a plusieurs dictionnaires classés selon les racines.
- Mr Zenkouar informe que tout dictionnaire standard doit passer par le moulin de l'IRCAM et attire l'attention des conférenciers sur la nécessité d'une coopération avec les chercheurs marocains et avec les chercheurs des autres pays pour promouvoir les technologies de l'information en relation avec la langue amazighe.

Monsieur Ishida conclut la table ronde par la nécessité, dans l'état actuel de la culture amazighe, de collaborer avec Microsoft Afrique du Nord et avec les développeurs des systèmes ouverts.

NB :

- (1) : UNL : Universal Networking language, voir la communication de Mr Ali RACHIDI
- (2) : XML eXtensible Markup Language (langage extensible de balisage)
- (3) : Projet papillon lancé en janvier 2000 par une coopération entre le GETA-CLIPS et le

National Institute of Informatics (NII) japonais avec le support de l'Ambassade de France à Tokyo. Ce projet a pour but de créer une base lexicale multilingue contenant entre autres l'allemand, l'anglais, le français, le japonais, le malais, le lao, le thaï, le vietnamien et le chinois.

Parmi les choses qui distinguent l'être humain des autres créatures est le fait qu'il écrit. L'écriture possède l'intéressante particularité de conserver l'oral. Elle permet ainsi aux civilisations et aux cultures de se transmettre à travers les âges et de subsister quelles que soient les vicissitudes de l'époque.

L'écriture informatique utilise les codages des caractères ; ceux-ci utilisent à leurs tours les fontes et les polices. Ces deux aspects sont liés mais répondent à des besoins différents. Le codage est une conceptualisation informatique de l'écriture. Il a pour but de caractériser l'information dans ses composants élémentaires. Un codage est une table de caractères. Un caractère étant une notion abstraite, immatérielle. Le document électronique est une prolongation de cet aspect qui aboutit à un codage international de tous les glyphes reconnus des écritures. Cet effort les introduira dans l'ordinateur et les inscrira dans ces métamorphoses nécessaires aux langues et à leur maintien dans le monde de l'écriture [...].

Le but de ce séminaire international est de trouver des solutions aux problèmes qui se posent à l'introduction des caractères, en particulier ceux de l'alphabet amazighe, dans les outils informatiques, tant du codage que de la beauté des glyphes et de l'art de les dessiner.