L'édition des textes multilingues

Abdel-Malek Boualem, Stéphane Harié
Laboratoire Parole et Langage
Université de Provence & CNRS
29, avenue Robert Schuman, 13621 Aix-en-Provence Cedex 1, France
e-mail: malek@lpl.univ-aix.fr

Introduction

Dans un précédent article [BOUA95a], nous présentions les difficultés de conception et de réalisation d'outils pour l'édition et le traitement de textes multilingues. Nous mentionnions que si des solutions commençaient à se mettre en place pour des langues européennes, la conception d'outils pour d'autres familles de langues était encore à un stade peu avancé. Nous présentions le prototype d'un éditeur multilingue sur lequel nous avions précédemment travaillé [BOUA90] et que nous avions intégré dans un environnement de traduction automatique du français vers l'arabe [BOUA93]. Cependant, cet éditeur présentait des faiblesses au niveau du codage des caractères et des documents, de l'incompatibilité des formats d'échanges des données textuelles et au niveau de l'environnement logiciel non portable.

Cet article développe les difficultés de la mise en place d'outils pour le traitement de textes multilingues et présente l'éditeur **MtScript** développé dans le cadre du projet MULTEXT [MUL96]. **MtScript** permet de combiner de nombreux types d'écritures dans un même document : latin, arabe, cyrillique, grec, hébreu, chinois, japonais, coréen, etc. (figure 1). Les fonctions d'édition de **MtScript** permettent d'insérer ou de supprimer des zones de texte même en écritures à sens opposés. De plus, **MtScript** permet d'identifier les langues utilisées dans un texte multilingue, de leur associer des règles de saisie au clavier et de traiter différents types de codage des caractères (sur un ou plusieurs octets). Enfin, **MtScript** a été développé dans un environnement portable (C, Tcl/Tk) et est basé sur les normes internationales de codage. La version 1.1 de **MtScript** (binaire pour Solaris et Linux) peut être téléchargée gratuitement sur le WWW à l'URL :

http://www.lpl.univ-aix.fr/projects/multext/MtScript/

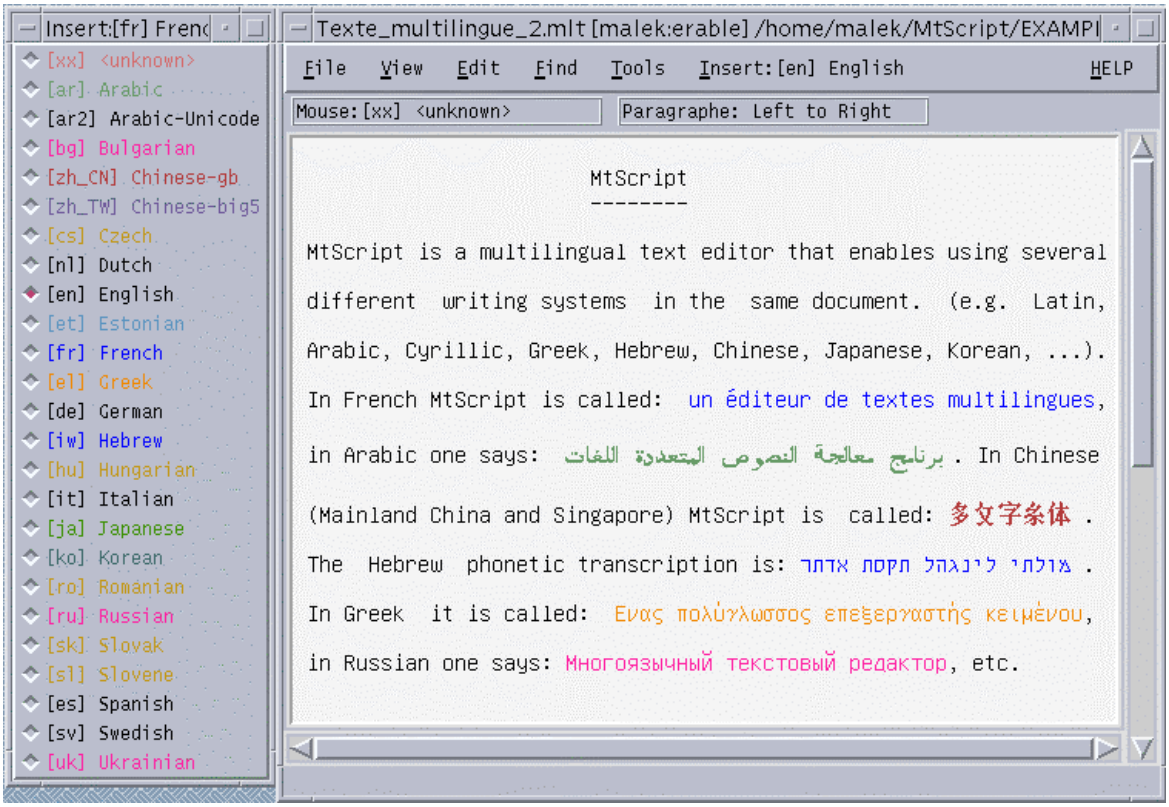

Figure 1. Un écran de MtScript

II. Difficultés de traitement des textes multilingues

Il existe un nombre de plus en plus important d'applications nécessitant des modules d'édition de textes multilingues tels que le traitement de texte, les interfaces aux bases de données, la micro-édition, etc. Dans le domaine de la traduction automatique ou assistée par ordinateur, un éditeur de textes multilingues est un outil fondamental pour les phases de pré-édition des textes sources et de post-édition des textes cibles [BENT91]. D'autre part, les éditeurs de textes multilingues sont d'une grande utilité dans le domaine de l'internationalisation et de la localisation des logiciels et de leur documentation. Ce nouveau domaine est né suite à l'émergence des nouvelles technologies et à la mondialisation du marché des technologies de l'information. Un certain nombre d'organismes et de projets travaillent de près ou de loin dans ce domaine : **CEC, CEN, LRE Glossasoft, Esprit, Eureka, JSA, Linux International, Unicode, TEI**, etc.

Le traitement de langues non basées sur l'écriture latine pose un certain nombre de difficultés. Par exemple, l'arabe s'écrit de la droite vers la gauche, le chinois contient des milliers d'idéogrammes; ce qui rend impossible leur codage sur un seul octet, en thaï et

dans certaines langues indiennes, la succession des caractères dans le mot ne correspond pas à leur succession phonétique, un caractère peut même en entourer d'autres, en coréen, les caractères s'agglutinent en syllabes. Les difficultés de réalisation d'éditeurs de textes multilingues se situent à différents niveaux : saisie, codage, édition, impression et échange de données (figure2).

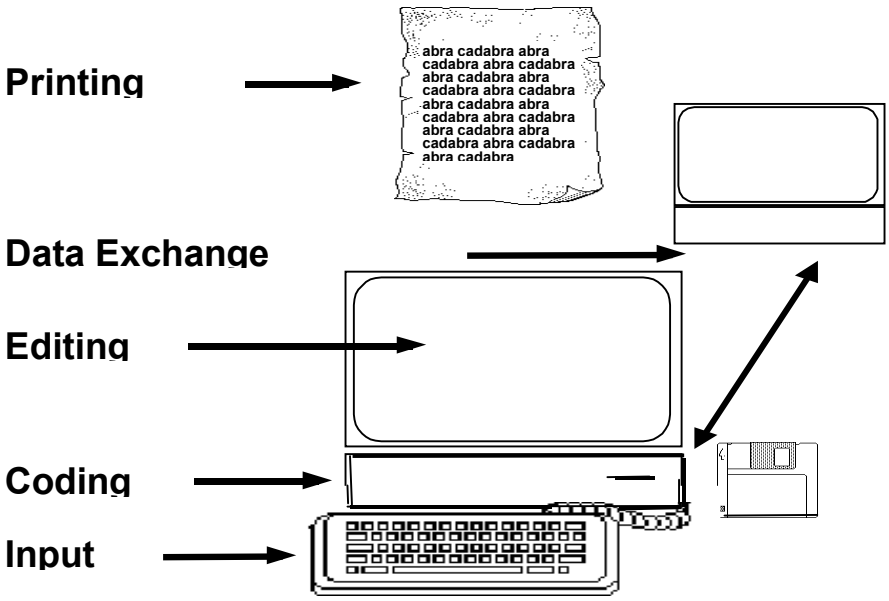

Figure 2.

II.1. Saisie de textes multilingues

Si beaucoup de claviers ne représentent que les caractères graphiques de l'ASCII (ou **ISO 646**), certains claviers localisés (ou adaptés) comportent des touches pour des caractères accentués ou des caractères spéciaux. Par exemple, les claviers français comportent généralement les touches correspondant aux caractères "**à ç é è ù**", mais les caractères comportant un accent circonflexe ou un tréma (**ê ï …**) sont saisis au moyen de deux frappes successives. En outre, il n'y a généralement pas de touche unique permettant de réaliser sur un clavier français des caractères existants dans d'autres langues européennes, tels que "**ñ**" ou "**ò**". Dans un contexte fortement multilingue, on ne peut d'ailleurs guère imaginer un clavier qui contienne tous les caractères possibles. L'adjonction de langues comme le chinois (plus de 6000 idéogrammes) ou l'arabe (environ 4 fois 28 lettres + 10 voyelles) rend nécessaire la mise au point de règles et de programmes spécifiques de saisie.

Les solutions proposées par les constructeurs d'ordinateurs sont souvent hétérogènes. Il existe en théorie une norme de saisie pour les claviers à 48 touches (**ISO/IEC 9995-3**) au moins pour l'écriture latine, mais elle n'est guère respectée. Un ensemble de méthode de saisie des caractères du répertoire de l'**ISO 10646** ont été récemment proposées [LABO95]:

entrée par code hexadécimal, par composition, etc. Toutefois, ces méthodes imposent une mémorisation quasi-impossible des codes par l'utilisateur et/ou un nombre important de frappes pour un même caractère. Il est donc nécessaire de développer des méthodes de saisie plus intuitives et minimisant si possible le nombre de frappes par l'utilisateur.

II.2. Codage

II.2.1. Codage des caractères

Les constructeurs d'ordinateurs et les concepteurs de logiciels utilisent de nombreux codes de caractères spécifiques et non compatibles :

- *MS-Windows character set for Western Europe MS CP1252,*
- *Dec Multinational Character Set,*
- *International IBM PC character set IBM CP850,*
- *Macintosh Extended Roman character set,*
- *Hewlett-Packard ROMAN8,*
- *etc.*

Cependant, des versions successives de normes de codage de caractères ont été élaborées au niveau international. Elles sont déjà utilisées sur certaines plates-formes. En particulier, la série de normes **ISO 8859-*** propose des jeux de caractères pour les alphabets latins (6 variantes adaptées à des régions différentes), cyrillique, arabe, grec et hébreu. Plus récemment (1993), la norme **ISO 10646** (*Universal multiple-octet coded character set* ou UCS) a proposé un jeu de caractères “universel" regroupant tous les jeux de caractères de l'**ISO 8859**, ainsi que le chinois, le coréen, le japonais, l’alphabet phonétique international (API), etc. Dans sa forme présente (**ISO 10646-1**), l'UCS utilise un codage sur 16 bits, qui correspond en réalité à la norme **UNICODE** et sera étendu à 32 bits dans les additions futures, ce qui permet un codage quasi illimité de caractères [JAMG95]. Toutefois, les environnements matériels et logiciels ne sont pas encore prêts à l'implémentation de jeux de caractères sur plusieurs octets, bien que la situation évolue rapidement (WINDOWS-NT, AT&T Bell Plan 9 et Apple QuickDraw GX). D'autre part, le langage **SGML** (Standard Generalized markup Language) est de plus en plus utilisé pour le codage des documents et des caractères de différentes langues à l'aide d'entités. SGML devient une véritable norme pour l'échange des documents multilingues.

II.2.2. Codage des systèmes d'écriture

Dans un texte multilingue il convient de coder non seulement les caractères individuels, mais aussi les systèmes d'écriture ou les scripts (notion plus générale que celle des langues) : script latin, grec, cyrillique, sémitique, etc. Dans le cas d'un codage sur un octet (par exemple la série ISO 8859-*), il est nécessaire de marquer le passage d'un jeu de caractères à l'autre (par exemple, passage de grec au cyrillique). Ce marquage peut être fait à l'aide d'un codage tel que celui proposé par la norme **ISO 2022** qui fournit des séquences

d'échappement <SI> (shift-in) et <SO> (shift-out) qui permettent le passage entre "jeu principal" et "jeu complémentaire". Ces mécanismes sont toutefois limités et des difficultés se posent, en particulier lors du mixage dans un même document de caractères codés sur un octet et sur plusieurs octets (par exemple mixage entre jeux **ISO 8859-*** et **GB-2312-80** ou **BIG5-0** pour le chinois, **JISX0208-1983-0** pour le japonais ou **KSC-5601-1987-0** pour le coréen).

Unifiant tous ces jeux de caractères en un jeu unique, l'**UCS** résout une partie du problème, puisqu'il n'est plus nécessaire de définir des mécanismes de changement de jeux de caractères. Toutefois, le problème n'est pas totalement résolu car l'UCS ne fournit pas de codage explicite des systèmes d'écriture, qui est nécessaire en particulier pour déterminer la direction d'écriture. Il est à noter que des protocoles d'écriture bidirectionnelle ont été proposés par le consortium **UNICODE**.

II.2.3. Codage des langues

Les traitements linguistiques dans un texte multilingue (segmentation, analyse morpho-lexicale, etc.) nécessitent l'identification des langues utilisées. La connaissance du jeu de caractères ou du système d'écriture ne suffit pas à identifier la langue dans laquelle est écrite une portion de texte : un document codé en ISO 8859-1, par exemple, peut aussi bien être écrit en français, en anglais, en espagnol ou même dans une combinaison de ces langues.

Il existe des normes de codification des noms des langues :

- **ISO 639-1988** : code à 2 lettres alphabétiques pour 140 langues (par exemple, "**en**" pour English, "**fr**" pour French, etc.),
- **ISO 639-2** : code à 3 lettres alphabétiques, alpha-3, en cours de développement ("**eng**" pour English, "**fre**" ou "**fra**" pour French, etc.).

Toutefois, dans le codage interne d'un document, ces codes ne peuvent pas être utilisés tels quels et il n'y a pas à l'heure actuelle de norme établie pour des séquences d'échappement qui permettraient de représenter le passage d'une langue à l'autre, bien qu'une proposition existe consistant à utiliser des codes de séquences de contrôle **ISO/IEC 6429** avec une conversion numérique des codes alphabétiques ci-dessus [LANG93]. Le marquage des langues est également en cours de définition dans le langage HTML utilisé sur le *World Wide Web* [YERG95].

II.3. Édition de textes multilingues

La plupart des langues s'écrivent horizontalement de la gauche vers la droite. Certaines langues comme l'arabe ou l'hébreu s'écrivent de la droite vers la gauche. D'autres langues comme le chinois ou le japonais peuvent même s'écrire du haut vers le bas (dans le cas de textes anciens). La cohabitation de langues à écritures opposées dans un même document

et particulièrement dans une même ligne de texte pose des difficultés lors de l'insertion ou la suppression de zones de texte. À l'extrême, l'exemple évoqué par J.Becker (figure 3) montre que le réarrangement des mots est nécessaire pour maintenir la cohérence sémantique de la phrase. Cet aspect représente l'une des grandes difficultés dans la conception d'éditeurs de textes multilingues.

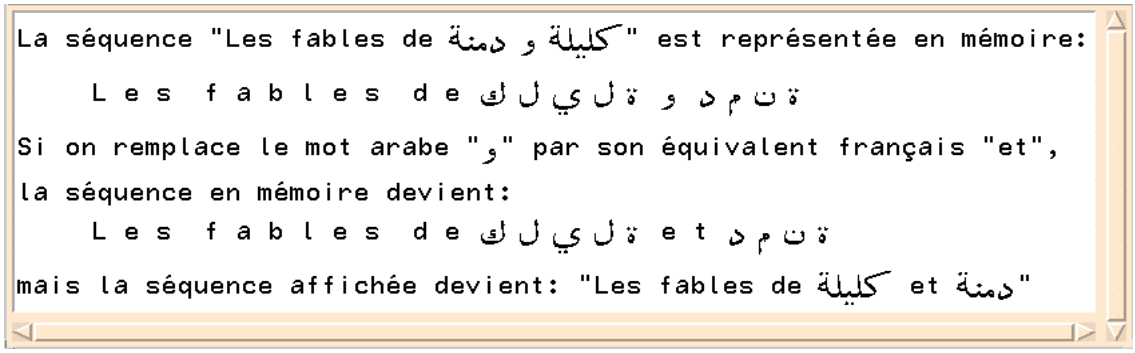

Figure 3. Edition dans un texte mixte

II.4. Impression de textes multilingues

Les nouvelles techniques d'impression incluant des représentations Bitmap ou PostScript ne sont pas tout à fait généralisées aux caractères non latins. L'impression de textes multilingues souffre du manque de polices de caractères non latins pour les imprimantes (essentiellement en PostScript). Toutefois, des travaux sont menés dans ce domaine comme ceux du projet OMEGA [YHJP95] pour la mise en place de fontes TeX et ceux de C.Bigelow et K.Holmes [CBKH95] dans la création d'une police de caractères *Unicode Lucida Sans* pour visualiser et imprimer des documents électroniques multilingues. La production de polices de caractères (PostScript, etc.) pour les nouvelles normes de codage permettrait sans doute une impression multilingue de qualité.

II.5. Échange d'informations multilingues

Avec la prolifération d'Internet, l'échange des documents multilingues dans un format électronique devient de plus en plus nécessaire. Jusqu'à une date récente, seuls les caractères graphiques invariants de l'**ISO 646-IRV** (ASCII) permettaient de véhiculer et restituer des textes électroniques "sans corruption" et il était donc nécessaire d'utiliser des mécanismes d'encodage tels que ***uuencode*** ou ***binhex***. Cependant, la situation est en voie d'amélioration : des normes ont été adoptées pour Internet qui permettent de transporter des caractères codés sur 8 bits de manière "clean" via le protocole TCP/IP (par exemple des applications comme *TELNET* et *FTP* sont "8-bit clean"). De plus, l'extension **MIME** (*Multi-purpose Internet Mail Extensions* : **RFC-1521** and **RFC-1522**) permet l'échange de données proprement en toutes circonstances par compactage et décompactage appropriés. Par ailleurs, le codage standard émergent pour l'échange de documents est SGML (et dans certains cas la TEI). Toutefois, ces "normes" ne sont pas encore définies de façon universelle et quelques problèmes de transmission subsistent. Enfin, notons que la garantie

de l'échange des octets sans corruption (sans perte du 8ème bit en particulier) n'est pas suffisante pour le transfert de données multilingues; il est nécessaire que les deux acteurs de la transaction, l'expéditeur et le destinataire, aient les mêmes mécanismes de codage des caractères, des systèmes d'écriture et des langues.

III. Les outils existants :

Les travaux pour l'élaboration d'outils d'édition de textes multilingues ont souvent été menés sous forme d'études expérimentales isolées aboutissant à des produits parfois incompatibles, difficiles à exploiter et non conformes aux normes de codage des caractères et des documents. En outre, les solutions proposées ne concernent, dans la plupart des cas, que les langues à alphabet latin et ne peuvent être adaptées à d'autres familles de langues. Les premières approches dans la conception d'éditeurs de textes multilingues furent proposées par Xerox (1ère et de 2ème génération d'outils "Star" [BECK84] et "ViewPoint Documenter" [BECK87]). D'autres outils se sont spécialisés dans des domaines particuliers comme la traduction assistée par ordinateur, par exemple les logiciels "TED" de Ink-Languages, "IDOS-A/II" de la société Integro et "TSS" de la société Alps. Ces logiciels ont été conçus de manière limitée à quelques langues européennes. Parmi les outils récents de traitement de textes multilingues, nous trouvons le logiciel “Universal Word” développé par Wysiwyg Corporation [UNIW96] qui intègre un grand nombre de langues et le logiciel "WinText" développé sous l'environnement Apple Macintosh par la société WinSoft. Celui-ci permet la fusion de plusieurs langues dans le même document même en écritures opposées. Dans l'environnement Windows, nous trouvons le logiciel de traitement de texte "Word" de Microsoft utilisant les interfaces multilingues TwinLink et TwinBridge. Dans le monde des stations de travail (sous Unix et sous d'autres systèmes), l'environnement TEX et son extension multilingue (ArabTeX, etc.) est quasi présent dans le milieu scientifique. Un grand nombre de polices de caractères au format TEX pour une variété de systèmes d'écriture ont été conçues. L'inconvénient de cet environnement est qu'il n'est pas wysiwyg[1] ou du moins dans certaines plates-formes. Les textes sources sont codés à l'aide de balises de structure, ils doivent être compilés pour générer la version finale dans un format visualisable (PostScript, etc.). Parmi les projets à vocation universelle, OMEGA comprend un certain nombre d'extensions de TEX qui améliorent ses possibilités de traitement multilingue. Il utilise la norme ISO 10646/UNICODE comme base de codage des caractères, mais accepte d'autres codages car il inclut un mécanisme de conversion de textes vers cette norme. Des algorithmes puissants permettent d'interpréter la composition ou la translitération des caractères non latins, de manipuler des codes de caractères différents et de générer les variantes graphiques correctes telles que les ligatures ou les formes contextuelles des caractères (typographie). Des efforts considérables sont également fournis pour adapter l'éditeur GNU Emacs (très largement utilisé dans le milieu scientifique) à d'autres langues autres que l'anglais (éditeur MULE). D'autres travaux plus

---

[1] What You See Is What You Get (synonyme d'un environnement où les informations sources ainsi que les informations finales sont représentées de la même façon).

récents dans le domaine du multilinguisme sont également en pleine effervescence. Nous citerons les activités du laboratoire CRL [CRL96] pour le développement d'outils dans une variété de domaines (traduction multilingue, extraction de textes, dictionnaires multilingues, etc.), les activités de la société Accent [ACC96] pour le développement de navigateurs WWW multilingues, les activités de l'institut Technion en Israel dans le traitement de textes bidirectionnels [BERRY92]. Contrairement à la plupart des travaux existants dans le domaine des éditeurs multilingues, **MtScript** est compatible avec les normes de codage des caractères et des documents (ISO-8859, Unicode, SGML) et il a été développé dans un environnement (Unix, C, Tcl/Tk) paramétrable, évolutif et portable vers les environnement Windows et Macintosh, en outre il présente une interface wysiwyg.

IV. Description de l'éditeur MtScript

IV.1. Caractéristiques principales

L'éditeur **MtScript** a été développé dans l'environnement C, Tcl/Tk, qui présente les avantages suivants :

- Tcl: langage de script (les commandes sont interprétées de façon interactive),
- facilité de la manipulation des données textuelles (caractères, fontes, mots, etc.),
- possibilité d'affecter des attributs aux caractères,
- possibilité de gérer des événements de X-Window (souris, clavier, etc.),
- convivialité de l'interface (X-Window, boutons, widgets, etc.)
- possibilité de la portabilité sur plusieurs autres environnements (Windows, etc.).

MtScript.1.1 fonctionne actuellement sous l'environnement Unix/X-Windows qui offre un support multilingue et des ressources pour la manipulation des attributs visuels des applications (fontes, couleurs, etc.). Pour rendre **MtScript** paramétrable au niveau du système, les jeux de caractères sont liés aux fontes à travers des fichiers d'alias du type "font.alias". Ceci permet à l'utilisateur de redéfinir des attributs des caractères (taille, couleur, etc.) et facilite la portabilité du logiciel à d'autres environnements.

MtScript est un éditeur de texte incluant toutes les caractéristiques d'un éditeur monolingue standard ainsi que des caractéristiques multilingues :

- mixage d'écritures en sens opposés sur la même ligne de texte,
- insertion et suppression de caractères dans les deux sens d'écriture,
- fonctions d'édition sur des portions multilingues de texte (copier / couper / coller),
- reconnaissance de la langue d'une portion donnée de texte, etc.

MtScript est indépendant de toutes langues. Les langues traitées sont des paramètres externes représentés par des fichiers de règles d'écriture et de translitération et des polices de caractères. L'adaptation de l'éditeur à une nouvelle langue nécessite “simplement” de fournir les polices de caractères adéquates ainsi que les règles d'écriture et de translitération.

IV.2. Représentation interne

Dans sa version actuelle, **MtScript** utilise les jeux de caractères suivants :

- iso8859-1, 2, 3 et 4 (Alphabets latins)
- iso8859-5 (Cyrillique)
- iso8859-6 (Arabe)
- iso8859-7 (Grec)
- iso8859-8 (Hébreu)
- gb2312-80 et big5-0 (Chinois)
- jisx0208-1983-0 (Japonais)
- ksc5601-1987-0 (Coréen)

Dans les prochaines versions, nous envisageons d'adopter l'UCS (ISO 10646) qui inclut d'autres systèmes d'écriture et un grand nombre de caractères absents dans les autres normes (par exemple, les ligatures typographiques, voire linguistiques **œ** et **Œ** qui sont considérées en Français comme étant des éléments textuels distincts des lettres qui les composent).

L'étiquetage des jeux de caractères et des langues est effectué à l'aide d'un fichier appelé **"feuille de style"** associé à chaque texte multilingue et contenant une instanciation des attributs des caractères. Ces attributs décrivent pour chaque portion de texte la langue, la police de caractères, le jeu de caractère, le style, la taille, la couleur, les tabulations, etc. Ces attributs sont associés à une position dans le texte exprimée par des numéros de lignes et de caractères. La figure 4 montre une partie de la "feuille de style" associée au texte de la figure 1. Nous développons actuellement un format d'échange SGML/HTML qui utilisera la balise <LANG> proposée par la norme HTML 3.0.

```
{mtscript_version 1.2}
{default_style
  { -width 80}{ -height 40}
  { -tabs {52.0 104.0 156.0 208.0 260.0 312.0 364.0 416.0 468.0 520.0 572.0 624.0 676.0 728.0 780.0}}
  { -wrap char}}
{newpage}
...
{ar {-foreground PaleGreen4 -font arabic} {13.22 13.63}}
{ar2 {-foreground Black -font arabic_unicode} {}}
{zh_CN {-foreground brown -font gb2312_1980} {15.53 15.63}}
{zh_TW {-foreground MediumPurple4 -font big5_0} {}}
{en {-foreground black -font iso_8859_1} {1.0 11.32 11.65 13.22 13.63 15.53 15.63 17.41 17.63 19.26 19.64
21.22
21.53 22.0 37.0 41.0}}
{fr {-foreground black -font iso_8859_1} {11.32 11.65 22.0 37.0}}
{el {-foreground DarkOrange2 -font iso_8859_7} {19.26 19.64}}
{iw {-foreground blue -font iso_8859_8} {17.41 17.63}}
{hu {-foreground DarkGoldenrod -font iso_8859_2} {}}
{ja {-foreground ForestGreen -font jisx0208_1983_0} {}}
{ko {-foreground DarkSlateGray -font ksc5601_1987_0} {}}
{ru {-foreground DeepPink1 -font iso_8859_5} {21.22 21.53}}
...
```

Figure 4. Extrait de la feuille de style associée au texte multilingue de la figure 1

IV.3. Saisie

MtScript utilise des modules de saisie basés sur les caractères qui se retrouvent sur la plupart des claviers, à savoir ceux de l'ISO 646-IRV. Les modules de saisie sont de 2 types:

- **Module de saisie alphabétique** pour les langues comme le français, l'arabe, le russe.
- **Module de saisie phonétique**[2] pour les langues à idéogrammes ou à phonèmes (chinois, Alphabet Phonétique International, etc.).

[2] *Le module de saisie phonétique du chinois a été développé et intégré à la version MtScript.2.0*

IV.3.1. Saisie alphabétique

Le module de saisie alphabétique est un programme unique pour toutes les langues alphabétiques. Bien entendu, les langues non latines sont saisies au clavier selon les règles de translitération standards correspondantes. Le module de saisie utilise un fichier de **règles d'écriture** et un fichier de **règles de translitération** par langue (figure 5).

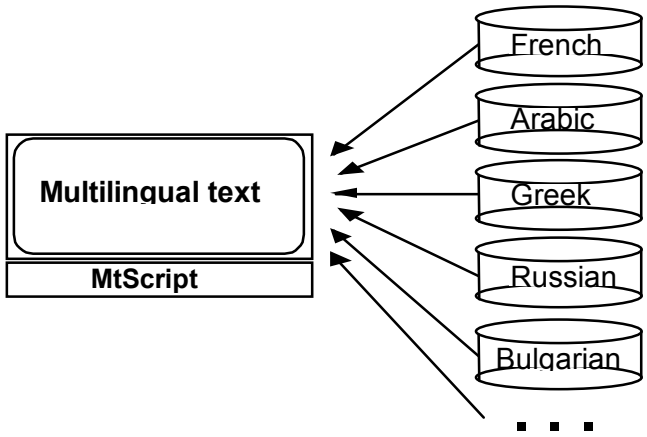

Figure 5. Règles d'écriture et règles de translitération des langues

Chaque fichier de règles d'écriture contient des classes de caractères et des règles d'écriture:

- les **classes de caractères** organisées en fonction de leur comportement commun, par exemple:
  - lettres minuscules accentuables,
  - lettres minuscules non accentuables
  - lettres majuscules accentuables
  - caractères invariants utilisés pour la saisie des accents,
  - chiffres,
  - etc.
- les **des règles d'écriture** spécifiques à la langue, par exemple:
  - français: **e** + **'** => **é** ; **c** + **,** => **ç** ; e + ESC + ' => e' ; etc.
  - grec: sigma => **σ** (début et milieu de mot) ou **ς** (fin de mot)
  - allemand: **s** + **s** => **ß** ; etc.
  - etc.

Les règles d'écriture des différentes langues sont exprimées dans un formalisme intuitif basé sur un mécanisme d'**automates à états finis** (nous envisageons d'utiliser un autre formalisme basé sur les expressions régulières et symboliques). Les fichiers de règles sont compilés et convertis sous forme de tables exploitables par les programmes de saisie. Des règles sont fournies par défaut pour chaque langue, mais elles peuvent être redéfinies à volonté par l'utilisateur, en fonction d'habitudes ou de besoins spécifiques, ou de particularités de certains claviers. Les règles par défaut sont basées sur les principes suivants:

- Les **caractères accentués** sont saisis par deux caractères, selon les règles déclarées pour la langue. Ainsi, dans les règles du français, **e** + **'** donne **é**, mais la même combinaison donne **e'** en anglais. La génération de **e'** en français (qui est une séquence beaucoup moins fréquente que **é**), passe par une séquence d'échappement: **e** + **ESC** + **'** => **e'**.

- Les **caractères non latins** sont saisis en respectant le plus possible les habitudes et conventions des langues concernées et les normes de translittération quand elles existent. Les tables de translitération sont représentées dans un fichier de règles de translitération associé à chaque langue (par exemple **ISO 233-1984/1993** pour l'arabe, **ISO 259-1984** pour l'hébreu, **ISO/R 843-1968** pour le grec, etc.). Ainsi, on tapera "**a**" pour "α", "**b**" pour "β", "**s**" pour "سـ", etc.

- Les **formes variantes** telles que les deux sigmas du grec (σ en début et milieu de mot ou ς en fin de mot), le double **s** allemand (β) ou les variantes positionnelles des lettres arabes sont générées de façon dynamique en fonction de leur position dans le mot, sans que l'utilisateur ait à intervenir. Le cas de la langue arabe est particulièrement intéressant [BOUA95b]; l'alphabet contient 28 lettres, dont la plupart s'écrivent sous 4 formes différentes selon leur position dans le mot (figure 6). **MtScript** gère totalement l'affichage des variantes positionnelles, y compris lors de l'insertion ou la suppression de caractères ou de mots.

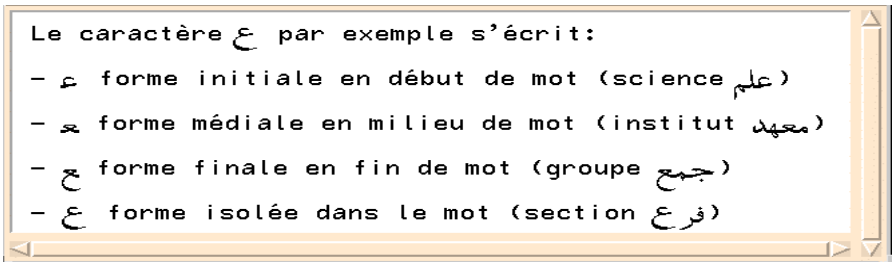

Figure 6. Variantes positionnelles de l'arabe.

IV.3.2. Saisie phonétique

Certaines langues, comme le chinois, s'écrivent à l'aide d'un nombre important d'idéogrammes. Le chinois traditionnel, par exemple, inclut plus de **60000 idéogrammes** représentant chacun un concept particulier. Dans les années 80, des versions d'idéogrammes simplifiées et différentes ont été adoptées par la Chine Populaire d'une part et par Taiwan et Hong Kong de l'autre part. Diverses méthodes de saisie existent, telles que la saisie par le code des caractères à deux octets (code **GB-2312-80**, etc.) ou bien la saisie **Pinyin** consistant en un codage phonétique des idéogrammes en caractères latins (420 syllabes accompagnées de tons différents, jusqu'à 5 tons par syllabe). La version **MtScript.2.0** intègre un module de saisie des idéogrammes chinois pour les deux principales normes de codage GB-2312-80 et BIG-5 et inclut différentes méthodes de saisie des idéogrammes (transcriptions phonétiques en Pinyin, radicaux, codes des quatre coins, etc.).

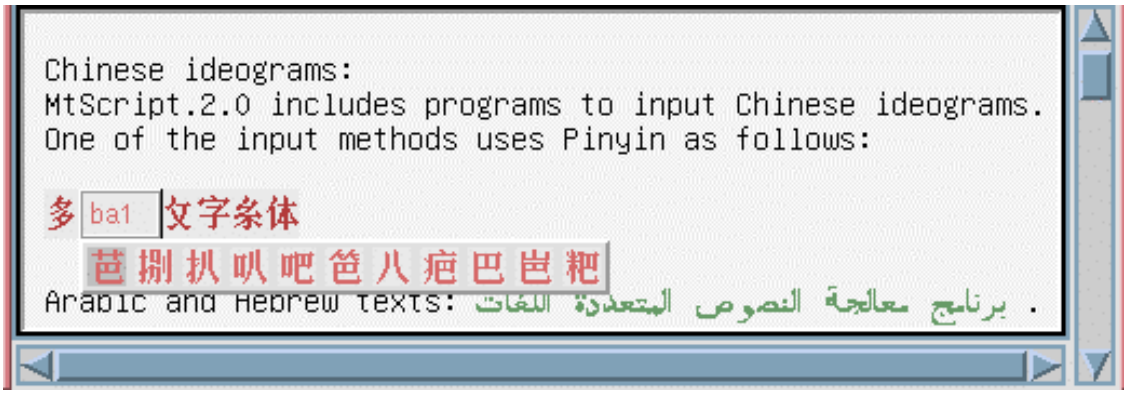

Figure 7. Saisie des idéogrammes chinois en Pinyin

IV.4. Affichage et restitution des caractères

Comme il a été dit plus haut, le problème fondamental de l'affichage de textes multilingues est la coexistence d'écritures en sens opposés sur la même ligne de texte. Les fonctions d'insertion et de suppression de caractères doivent tenir compte de leur sens d'écriture, selon des règles parfois complexes. **MtScript** donne la possibilité à l'utilisateur de définir de façon interactive le sens d'écriture principal pour une zone de texte ou paragraphe, gauche-droite ou droite-gauche. L'autre sens sera le sens secondaire. Le curseur se déplace alors seulement dans le sens principal. Lorsqu'une séquence de caractères est entrée dans une langue utilisant le sens secondaire, le curseur reste fixe et les caractères s'écrivent en mode insertion (figure 8).

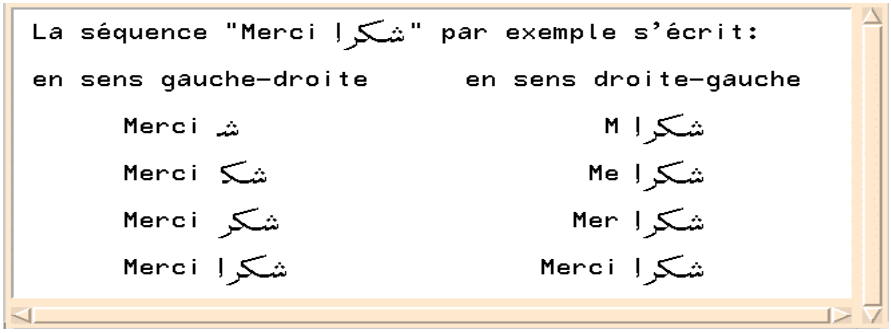

Figure 8. Directions d'écriture mixtes

Pour ce qui est de la justification des textes multilingues incluant une écriture bidirectionnelle, la version actuelle de **MtScript** ne permet que la justification à gauche ou bien à droite des textes. La justification des textes des deux côtés est un problème complexe non encore résolu. En effet, contrairement aux textes à base d'alphabet latin qui peuvent être justifiés par l'utilisation d'**extra-espaces** du fait que les caractères ne sont pas liés entre eux, la justification de textes arabes, par exemple, nécessite l'utilisation de **fontes dynamiques** où les caractères peuvent être contractés et décontractés de façon dynamique en fonction de la longueur des lignes de textes. L'exemple de la figure 9 montre que le caractère "Ba" (à droite) du mot "KaTaBa" (écrire) doit être adapté à la longueur de la ligne de texte (un caractère spécial "-" peut aussi être utilisé pour rallonger le dessin des caractères).

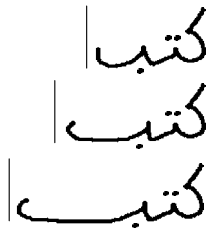

Figure 9. Fontes dynamiques pour la justification des textes arabes

Quelques travaux pour résoudre le problème de la justification des textes bidirectionnels (incluant l'arabe ou l'hébreu) ont été proposés notamment ceux de D.M.Berry [BERRY89], [BERRY90] et [BERRY92] et de D.E.Knuth et P.MacKay [KNMC87]. Nous travaillons actuellement sur une synthèse de ces travaux dans la perspective de définir des mécanismes de justification bidirectionnelle des textes multilingues.

IV.5. Echanges de textes multilingues

Dans la version actuelle de **MtScript**, les textes sont codés en ISO-8859-*, GB-2312-80, JISX0208 et KSC-5601-1987-0 et chaque texte est associé à un fichier de style. Celui-ci contient les attributs des portions de textes (langue, police de caractères, etc.). Ainsi, **MtScript** peut éditer des textes **importés** s'ils sont codés dans l'un des standards ci-dessus. Si le texte multilingue importé contient plus d'une langue, il est nécessaire que des

informations soient fournies indiquant la langue de chaque partie du texte. D'un autre côté, les textes produits par **MtScript** peuvent être **exportés** (même à travers Internet via des protocoles tels que MIME) vers d'autres éditeurs multilingues s'ils supportent les mêmes normes de codage des textes. Cependant, compte tenu que les fichiers de styles associés à **MtScript** ne peuvent être interprétés par les autres éditeurs, les informations relatives aux attributs des textes (en particulier la langue) doivent être fournies d'une façon "standard". Dans ce contexte, le métalangage SGML (et le langage HTML pour le Web) présente sans doute une solution efficace s'il est adopté par l'ensemble des fournisseurs et des utilisateurs d'applications et de ressources multilingues.

V. Développements futurs :

A l'heure actuelle le prototype **MtScript** existe, il inclut la plupart des fonctionnalités visées. Il a été distribué dans une version compilée[3] pour Unix (Solaris et Linux) et a été téléchargé et utilisé par un grand nombre d'utilisateurs dans plus de 32 pays. Un grand nombre d'entre eux ont mis en évidence la compatibilité de **MtScript** avec les normes de codage des caractères et de translitération, la simplicité de son utilisation (wysiwyg) et son aspect paramétrable. Certains utilisateurs ont redéfini des règles d'écriture ou de translitération pour les adapter à leurs langues et d'autres ont lié **MtScript** avec des applications spécifiques.

Cependant, la version actuelle de **MtScript** présente un certain nombre de limites qui sont étudiées à différents niveaux. Chaque texte est associé à un fichier de style qui n'est interprétable que par **MtScript**; ce qui limite considérablement ses possibilités s'échange d'informations même s'il respecte les normes standards de codages des caractères. SGML et la TEI offrent des possibilités de codage des textes multilingues et sont en phase de devenir des standards pour l'échange d'informations multilingues [IDVE95].

MtScript.2.0 qui est en cours d'amélioration inclut de nouvelles fonctionnalités:

- programme de saisie du chinois codé en GB-2312-80 et en BIG-5,
- possibilité de lier **MtScript** à des commandes externes d'Unix (Shell, etc.),
- possibilité d'utiliser le programme de correction orthographique d'Unix (ispell),
- possibilité d'impression de textes latins (disponibilité de polices PostScript gratuites),
- intégration de nouvelles langues,
- codage SGML de textes (module non encore terminé).

D'autres fonctionnalités pourront aussi être envisagées:

- intégration du codage ISO 10646 / UNICODE,

---

[3] Il n'est pas exclu que d'autres distribution de MtScript soient effectuées via une licence publique de GNU.

- intégration des fonctionnalités de représentation de HTML qui permettraient l'enrichissement des textes et l'utilisation de **MtScript** en association avec un navigateur WWW,

- intégration du protocole MIME pour coupler **MtScript** aux logiciels du courrier électronique,

- justification des textes multilingues bidirectionnels,

- impression de textes autres que latins (relative à la disponibilité des polices PostScript).

Par ailleurs, il serait utile d'intégrer les voyelles dans l'écriture arabe, celles-ci sont des signes placés au-dessus ou bien en dessous des lettres. Bien que les textes arabes diffusés dans la presse et les journaux ne soient généralement pas voyellés, les voyelles jouent un rôle important dans le traitement automatique de l'arabe (figure 10).

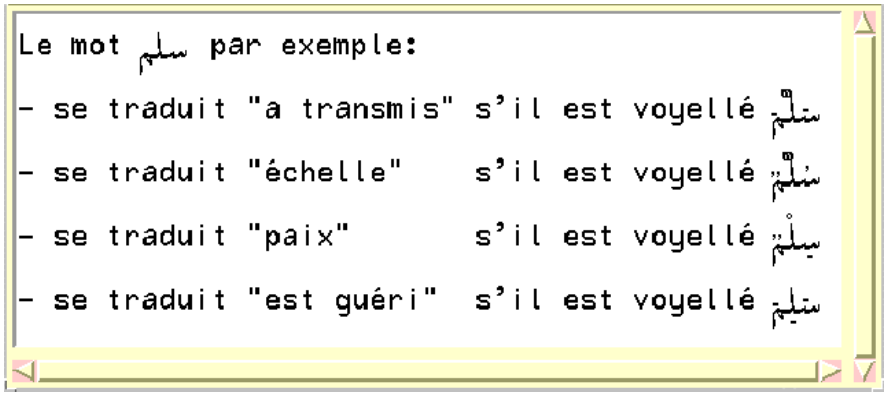

Figure 10. Voyelles en arabe

La version actuelle de **MtScript** utilise une police de caractères non voyellés issue du code Metafont présenté par Y.Haralambous à partir du format TEX. Mais nous intégrons actuellement une version plus complète de cette fonte contenant la plupart des caractères requis par UNICODE et incluant les voyelles ainsi que les formes contextuelles des caractères.

Conclusion :

On assiste aujourd'hui à une évolution des technologies de l'information entraînant un besoin croissant en outils multilingues. Les échanges d'informations entre partenaires de langues différentes sont de plus en plus fréquents. L'éditeur de textes multilingues est un élément d'articulation dans cette internationalisation des supports de l'information. **MtScript** a été développé dans la perspective de répondre à des besoins en outils de codage et de traitement de documents multilingues, tant pour des langues européennes que des langues non européennes. Cet outil permet de saisir, éditer, mémoriser et échanger des textes multilingues. Il ouvre des perspectives sur un certain nombre d'applications nécessitant des traitements de textes multilingues telles que la segmentation et l'analyse morphologique de textes, la traduction automatique, les dictionnaires multilingues ainsi que la localisation des logiciels (et de leur documentation) dans différentes langues.

Remerciements :

Ce travail a bénéficié du financement de la Commission Européenne dans le cadre du projet Multext. Diverses personnes ont contribué à l'amélioration du logiciel. Les auteurs tiennent en particulier à remercier Jean Véronis, Directeur de l'équipe de traitement automatique des langues au laboratoire LPL et coordinateur du projet Multext, Greg Priest-Dorman pour ses tests approfondis de MtScript, Emmanuel Flachaire pour la compilation sous Linux (Intel) et Nancy Ide pour son aide sur la documentation anglaise. Nous remercions également Mark Leisher du laboratoire CRL (New Mexico State University) pour ses commentaires et sa collaboration par de nombreuses ressources, ainsi que les lecteurs anonymes pour leurs commentaires détaillés.

Références Bibliographiques

[ACC96] http://www.accentsoft.com

[BECK84] **J. Becker**, "The multilingual word processing", *Pour La Science*, Septembre 1984, 66-67.

[BECK87] **J. Becker,** "Arabic word processing", *Communications of the ACM*, volume 30, number 7, Juillet 1987, 600-610.

[BENT91] **P. M. Benton,** "The Multilingual Edge", *BYTE*, Mars 1991, 124-132.

[BERRY89] **Z.Becker, D.Berry,** "triroff, an adaptation of the device-independant troff for formatting tri-directional text", *ELECTRONIC PUBLISHING*, Vol.2(3), October 1989, 119-142.

[BERRY90] **U.Habusha, D.Berry,** "vi.iv, a bi-directional version of the vi full-screen editor", *ELECTRONIC PUBLISHING*, Vol.3(2), May 1990, 65-91.

[BERRY92] **J.Srouji, D.Berry,** "Arabic formatting with ditroff/ffortid", *ELECTRONIC PUBLISHING*, Vol.5(4), December 1992, 163-208.

[BOPI95] **L. Bourbeau, F.** Pinard, "Normalisation et internationalisation: inventaire et prospective des normes clefs pour le traitement informatique du français". *Progiciels BPI.* , Montréal, Canada, 1995.

[BOUA90] **A.M. Boualem,** "The multilingual terminal", rapport de recherche interne, INRIA Sophia Antipolis, Janvier 1990, 1-4.

[BOUA93] **A.M. Boualem,** "ML-TASC: Système de traduction automatique multilingue dans un environnement à syntaxe contrôlée", SS'93, *7th annual High Performance Computing Conference*, Alberta, Canada, Juin 1993, 537-544

[BOUA95a] **A.M. Boualem,** "Multilingual text editing", *SNLP'95, The 2nd Symposium on Natural Language Processing*, NECTEC, C&C, Bangkok, Août 1995, 336-342.

[BOUA95b] **A.M. Boualem,** "Arabic Language Processing", *SNLP'95, The 2nd Symposium on Natural Language Processing*, NECTEC, C&C, Bangkok, Août 1995, 95-102.

[CBKH95] **C. Bigelow, K. Holmes,** “The design of a UNICODE font”, version française dans le *Cahier GUTenberg* n°20, Mai 1995, 81-102.

[CRL96] http://crl.nmsu.edu

[IDVE95] **N. Ide, J. Véronis,** *The Text Encoding Intiative: Background and Context*, Kluwer Academic Publishers, Dordrecht, 1995.

[JAMG95] **J. André, M. Goossens,** "Codage des caractères et multi-linguisme: de l'ASCII à UNICODE et ISO/IEC-10646", *Cahier GUTenberg* n°20, Mai 1995, 1-54.

[KNMC87] **D.E.Knuth, P.MacKay,** "Mixing Right-to-left Texts with Left-to-right Texts", *TUGBoat*, 8(1), 1987, 14-25.

[LABO95] **A. Labonté,** "Input methods to enter characters from the repertoire of ISO/IEC 10646 with a keyboard or other input devices". *ISO/CEI JTC1/SC18/GT9 Working Draft*, Février 1995.

ftp://ftp. funet. fi/pub/doc/charsets/ucs-input-methods

[LANG93] *Language Coding Using ISO/IEC 6429*. Draft circulated in January 1993 by the European Standardization Organization CEN technical committee TC304. Available electronically at:

http://www.stonehand.com/unicode/standard/tc304.html/

[MUL96] http://www.lpl.univ-aix.fr/projects/multext/

MULTEXT est le nom générique d'un ensemble de projets coordonnés par le laboratoire "Parole et Langage" du CNRS & Université de Provence pour la mise en place de méthodes standards pour la représentation d'informations linguistiques et d'outils pour le traitement de près de 15 langues: LRE-MULTEXT (*Linguistic Research and Engineering* Program), MULTEXT-EAST (*Copernicus* Program), MULTEXT-CATALOC (Program of *Langues Régionales et Minoritaires* de la DGXXII), ALAF Research Shared Action (Alignement of African and French Languages, AUPELF•UREF).

[UNIW96] http://www.wysiwyg.com

[YERG95] **F. Yergeau, G. Nicol, G. Adams, M. Duerst,** "Internationalization of the Hypertext Markup Language". *Internet Draft draft-ietf-html-i18n-02*, November 1995.

http://www.ics.uci.edu/pub/html/draft-ietf-html-i18n-02.txt

[YHJP95] **Y. Haralambous, J. Plaice,** "Ω, une extension de TEX incluant UNICODE et des filtres de type Lex", *Cahier GUTenberg* n°20, Mai 1995, 55-79.