QUEEN
QA
76.6
.L33
1987

Guide conceptuel pour la réalisation technique des logiciels de langue française à portée multilingue

Version revue et corrigée
Janvier 1987

Étude réalisée
dans le cadre des travaux
du comité Canada-Québec
sur le développement du logiciel
d'expression française

Canada

Québec

Cette étude a été produite dans le cadre des travaux du Comité Canada/Québec sur le développement du logiciel d'expression française. L'auteur est M. Alain LaBonté du Bureau central de l'informatique du ministère des Communications du Québec. A collaboré à cette étude pour les aspects normatifs, M. Jean-Yves Fortin, de la Direction de l'industrie informatique et du développement économique du ministère des Communications du Canada.

Édition réalisée à la Direction des communications du ministère des Communications du Québec

© Gouvernement du Québec
Gouvernement du Canada

Dépôt légal - 1er trimestre 1987
Bibliothèque nationale du Québec
Bibliothèque nationale du Canada

ISBN 2-550-17029-6

Queen
QA
76.6
L33
1987

Guide conceptuel pour la réalisation technique des logiciels de langue française à portée multilingue

Version revue et corrigée
Janvier 1987

Industry Canada
Library Queen
AVR APR 28 1998
Industrie Canada
Bibliothèque Queen

Étude réalisée
par Alain LaBonté
dans le cadre des travaux
du comité Canada-Québec
sur le développement du logiciel
d'expression française

COMMUNICATIONS CANADA
SEP 20 1990
LIBRARY – BIBLIOTHÈQUE

Canada Québec

QA
76.6
L33
1987

DD 7014940
DL 9942823

Table des matières.

- Page -

1. Introduction 3

2. Considérations sur les jeux de caractères. 4

2.1 Le répertoire des graphèmes utilisés. 4
2.2 Les méthodes de représentation. 5
2.3 Les opérations textuelles critiques 7

2.3.1 Les validations 7
2.3.2 La recherche et les comparaisons. 7
2.3.3 Les conversions de caractères. 8
2.3.4 Les tris. 9

3. Particularités linguistiques et culturelles des données. 10

3.1 Les coupures syllabiques. 10
3.2 Les règles grammaticales dans les messages variables. 10
3.3 Les types de données spéciales. 11
3.4 Une fonction globale très importante. 11

4. La modularité des éléments textuels. 12
4.1 Les messages. 12
4.2 Les réponses anticipées 12
4.3 Les noms de variables 13
4.4 Les langages informatiques et les noms de commandes 13
4.5 Les traitements dépendants de la représentation interne 13

5. Claviers, écrans, imprimantes et leurs interfaces. 14

6. La documentation et la terminologie utilisée 15

7. Les normes nationales et internationales 17
7.1 Normes touchant les jeux de caractères. 17
7.2 Normes touchant les techniques d'extension de codes 19
7.3 Normes concernant les langages de programmation 19
7.4 Normes relatives à la reconnaissance formelle des caractères. . . . 20
7.5 Normes relatives aux claviers 20
7.6 Normes à incidences culturelles 21

8. Conclusion 22

Annexe. Bibliographie 23

1. Introduction.

Au cours de la dernière année, les ministères des Communications du Québec et du Canada se sont entendus pour harmoniser leurs efforts dans le domaine des politiques de développements du logiciel d'expression française.

Pendant cette période, les réflexions des divers intervenants ont conduit à une série de constatations et de recommandations, dont l'une traite de l'intérêt de tirer profit du caractère français des logiciels québécois pour encourager le développement de logiciels à portée multilingue.

Il va sans dire que les firmes se spécialisant dans cette direction devraient avoir de meilleures chances de concurrencer les firmes américaines sur le marché international qu'il est possible de le faire aujourd'hui, compte tenu des moyens mis en place, des capitaux en jeu et du bassin de population domestique considérable dont bénéficient déjà nos voisins du Sud.

Le Ministère des Communications du Canada possède une longue expérience dans l'élaboration et la gestion de normes nationales et internationales en matière d'informatique. Il est représenté au niveau de l'ISO (International Standard Organization) et du CCITT (Comité consultatif international télégraphique et téléphonique) qui sont des organisations mondiales de normalisation, la dernière étant spécialisée en téléphonie et télégraphie. Ces entités tentent de prévoir les aspects linguistiques et culturels dans les normes touchant le monde de l'informatique. Mais elles ne se sont pas encore penchées comme tel sur une véritable architecture universelle des langues nationales.

Le Ministère des Communications du Québec fait partie depuis peu de l'organisation européenne SEAS (SHARE European Association), qui a produit en septembre 1985 un «livre blanc» sur les caractères nationaux, les langues et les claviers dans un environnement IBM. Cette association, formée d'utilisateurs de grands systèmes IBM, revendique de cette dernière une architecture des langues nationales. Le Bureau central de l'informatique participe à cet effort de revendication, ayant lui-même à son actif une bonne expertise sur le sujet. Il est aussi représenté depuis peu au sein d'un comité canadien de normalisation des claviers, où il défend très activement les besoins des francophones en matière d'entrée de texte et de données.

Ce sont les fruits de cette expérience et de ces réflexions qui sont présentés ici de façon condensée à l'intention des citoyens québécois et canadiens désireux de réfléchir à leur tour à ce sujet avant de passer à l'action, c'est-à-dire de produire des logiciels exportables de bonne qualité. Nous espérons que la forme du document sera suffisamment claire et concise pour faire oeuvre utile dans le domaine de la conception de bons logiciels avant tout en français pour les besoins québécois et canadiens, mais par surcroît, facilement traduisibles pour le marché international.

2. Considérations sur les jeux de caractères.

2.1 Le répertoire des graphèmes utilisés.

Il va sans dire que le soutien d'un produit dans une langue nécessite de faire les bons choix dans l'établissement du répertoire de caractères d'un pays, et ce, avant de se pencher sur leur représentation informatique. Il serait malheureux qu'un développement technologique amène la radiation de plusieurs siècles d'histoire en quelques mois. C'est pourtant ce à quoi on en est presque venu avec l'arrivée des machines à écrire, qui ont empêché l'utilisation des majuscules accentuées en français alors que celles-ci ont toujours eu dans cette langue pleine valeur orthographique: on doit considérer comme faute d'orthographe toute absence d'accentuation, que ce soit en majuscules ou en minuscules [1]. Et que dire de l'apparition des terminaux informatiques où ont dominé les majuscules non accentuées pendant près d'un quart de siècle? Il est facile au commun des mortels de répondre à cette question.

Les logiciels conçus pour une langue donnée, ou pour un territoire national donné, doivent donc pratiquement comprendre le répertoire complet des caractères en usage par la population visée.

Dans un contexte multilingue, plusieurs communications utilisant divers supports informatiques, s'effectuent entre différents pays, et il est idéalement préférable, à un deuxième niveau, de prévoir l'utilisation d'un répertoire de caractères étendu comprenant l'ensemble des caractères communs à ces pays, ce qui revient à dire, en termes pratiques, que l'ensemble des caractères latins affectés de leurs signes diacritiques et quelques autres symboles ou graphèmes d'usage national et international doivent être prévus par les concepteurs. Ceci nécessite au moins 180 caractères de plus que les caractères latins non accentués. C'est un idéal pour lequel la technologie actuelle nous oblige souvent à des compromis (comparer, par exemple, le répertoire des caractères représentables dans la norme Télétex de base, qui comprend plus de 256 graphèmes différents, au répertoire de caractères EBCDIC étendu d'IBM, qui comprend 190 caractères [avec un soutien presque entier du français], ou avec l'ASCII étendu standard du PC, qui ne permet l'accentuation que de trois majuscules françaises).

Les compromis sont possibles en utilisant des jeux de caractères régionaux, comme dans le cas du produit L'ACCENT, interface commercialisée par IBM au Canada pour rendre disponible un soutien des majuscules accentuées sur PC. Mais il faut être conscient que dans ces contextes, l'exportabilité des logiciels utilisant ces solutions régionales en souffrira, à moins qu'on les préserve entièrement de la dépendance conceptuelle avec un répertoire de caractères donné, national ou international.

À un troisième niveau, le soutien des langues orientales non alphabétiques (en particulier le chinois) nécessite une conception totalement différente et nous ne nous attaquerons pas à ce problème ici en raison de sa trop grande complexité, qui déborde la portée de ce guide.

Finalement le choix du répertoire de caractères relève de la conception même des systèmes d'exploitation ainsi que des grands modes de communication de l'information véhiculés par d'autres logiciels de base. Le choix du système d'exploitation comme support primaire de développement est donc important dès la conception de logiciels à portée multilingue. En effet, bien que les répertoires de graphèmes puissent varier pour un même système d'exploitation et que certains systèmes permettent de dessiner les caractères manquants (le Macintosh, entre autres, fait partie de cette catégorie), la communauté des utilisateurs d'un système a généralement tendance à adopter comme norme de facto un répertoire donné, ce qui est un facteur de communication important.

Notons, pour les besoins de développement de logiciels de langue française que le français comprend, au sens strict et au minimum, les 26 lettres de l'alphabet, les 10 chiffres, les symboles de ponctuation communs à toutes les langues européennes, de même que 13 lettres accentuées. À cela il faut ajouter les ligatures «ae» et «oe» [2] qui n'ont pas seulement un usage typographique mais servent à distinguer les diphtongues (comme dans «oeuf») des hiatus simples comme dans «coercitif». Il faut encore ajouter les guillemets anguleux (ou français) ouverts et fermés, à distinguer des guillemets anglais («quotes» ou «"»).Ceci nécessite donc un répertoire de graphèmes d'au-delà de 128 caractères.

2.2 Les méthodes de représentation.

Une fois choisis le système d'exploitation et le répertoire de graphèmes utilisés, il faut déterminer comment on représentera et on traitera cette information, soit pour le stockage informatique, soit pour la transmission télématique.

Traditionnellement, les environnements IBM ont utilisé un code à 8 bits pour représenter de façon interne les jeux de caractères et pour les transmettre. Les tables utilisées par IBM sont identifiées sous le nom de EBCDIC («Extended Binary Coded Decimal Interchange Code»). Toutes ces tables ont en commun les caractères latins non accentués, les chiffres et un certain nombre de symboles spéciaux servant à la ponctuation et à d'autres usages.

En utilisant un code à 8 bits, la convention EBCDIC permet la représentation simultanée de 256 caractères, ce qui est suffisant pour représenter le répertoire des caractères du français et en général de toute autre langue européenne prise individuellement. Il est cependant insuffisant pour les représenter toutes à la fois. C'est pourquoi IBM a introduit la notion «d'usage national» réservé à un certain nombre de positions dans la table, d'où en pratique l'existence de plusieurs tables EBCDIC.

Ces diverses tables sont actuellement incohérentes, c'est-à-dire que les mêmes graphèmes dans différentes tables ont été représentés par des codes différents, ce qui pose de sérieux problèmes au niveau des communications internationales (ex.: le «ü» allemand comporte 4 codes différents, selon qu'on est au Canada, en Allemagne, au Danemark ou en Suède): on en arrive à une situation où un caractère présent dans un nom de famille, par exemple, est mal transmis dans un autre pays alors qu'il serait physiquement possible de le reproduire dans le pays récepteur. Inversement, aux mêmes codes peuvent correspondre différents graphèmes, comme le proverbial exemple connu du «$» américain qui est transmis comme «£» au Royaume-Uni.

Le développement de logiciels multilingues doit tenir compte de cette réalité dans les environnements IBM. L'architecture DCA d'IBM, sans régler ce problème d'office, permet au moins de savoir, lorsque des documents sont transmis, à quelle table on a affaire. L'évolution à attendre dans cet environnement sera vraisemblablement due aux efforts de SEAS, organisation européenne qui réclame d'IBM une «architecture des langues nationales (NLA)»[3]. Le Ministère des Communications du Québec (Bureau central de l'informatique) appartient à cette organisation et se préoccupe activement du suivi de ces revendications. Notons au passage qu'il existe une ébauche de norme de l'ISO traitant de la structure des documents bureautiques (voir section 7, norme préliminaire DP 8613-1985). Cette ébauche est le pendant de l'architecture DCA d'IBM, mais tout comme cette dernière, elle ne s'attaque pas à une véritable architecture linguistique des systèmes.

Ailleurs que dans les environnements IBM, le reste de l'industrie informatique occidentale, largement dominé par la technologie américaine, a traditionnellement utilisé un code à 7 bits appelé USASCII («U. S. American Standard Code for Information Interchange»). Ceci rend le problème un peu plus compliqué. En effet, un code à 7 bits ne permet la représentation simultanée dans une table que de 128 symboles différents.

Ces dernières années, plusieurs firmes ont développé parallèlement des extensions à 8 bits de ce jeu de caractères, permettant ainsi de représenter différents symboles nationaux et internationaux. Ces développements, propres à chaque firme, ont créé une diversité très complexe dont il est maintenant difficile de se débarrasser, à l'heure où les normes internationales ont été définies. En effet, malgré qu'il existe maintenant une norme internationale, adaptée au Canada [4], définissant un jeu de caractères ASCII à 8 bits (soutenant presque intégralement le répertoire des graphèmes du français), les normes de facto dans ce domaine (par exemple, le jeu de caractères du PC) ont plus de force et sont difficiles à déloger, malgré leur caractère incomplet.

Ce qui vient aussi compliquer les choses en ASCII, c'est que quelques firmes ont choisi l'approche du soutien de graphèmes internationaux par des jeux de caractères alternatifs à 7 bits, où des séquences de codes de contrôle déterminent dans quelles tables se situent les chaînes de caractères qui suivent. Cette dernière méthode est nettement à déconseiller pour le stockage et le traitement informatique de texte, puisqu'elle entraîne d'innombrables possibilités d'erreurs et d'embêtements lors des opérations de comparaison, de recherche ou de conversion nécessaires dans tout logiciel manipulant des données textuelles. Dans les versions de UNIX antérieures à UNIX V version 3, il semble qu'une telle méthode était l'approche la moins ardue pour soutenir d'autres langues que l'anglais, puisque les codes ASCII à 7 bits étaient les seuls qui étaient utilisés dans l'ensemble des importants utilitaires de ce système d'exploitation transportable (à partir de la version 3, les utilitaires ont été réécrits pour éliminer le filtrage des caractères qui empêchait la reconnaissance de jeux de caractères ASCII étendus à 8 bits).

Une autre méthode de représentation des caractères accentués est utilisée dans la transmission Télétex, définie en 1980 par le CCITT [5]. Selon cette méthode, un graphème accentué est représenté et transmis à l'aide d'une suite de 2 caractères ASCII de 8 bits, l'un représentant le caractère diacritique (ou accent), le suivant représentant le caractère latin non accentué. Dans un environnement non-IBM, cette méthode de représentation est recommandable pour la transmission de caractères multilingues, l'idéal étant l'application intégrale de la norme Télétex. C'est une méthode rigoureuse qui permet d'assurer un lien cohérent entre appareils autrement incompatibles.

Pour le stockage et le traitement interne de caractères dans une machine donnée, la représentation à l'aide d'un jeu de caractères à 8 bits, où chaque caractère accentué est représenté de façon distincte, est à coup sûr la meilleure approche, en attendant l'apparition hypothétique de jeux de caractères à 16 bits dans un avenir imprévisible.

2.3 Les opérations textuelles critiques.

Dans un contexte multilingue, plusieurs opérations impliquant des données textuelles sont critiques, à cause de la spécificité des répertoires de graphèmes propres à chaque langue. Ces opérations sont tout particulièrement les validations alphabétiques, la recherche et les comparaisons alphabétiques, les conversions de caractères et les tris. Nous décrivons brièvement ici les points dont doivent particulièrement tenir compte les concepteurs lors du développement d'un logiciel à portée multilingue. Les tests devraient aussi être orientés dans ce sens pour assurer la conformité aux particularités linguistiques.

2.3.1 Les validations.

Certains langages informatiques (en particulier COBOL) possèdent des énoncés, commandes ou verbes de validation alphabétique. On doit s'assurer que les interpréteurs ou compilateurs impliqués effectuent correctement ces validations avant de les utiliser. Ce n'est pas le cas avec les différentes versions de COBOL actuellement sur le marché, qui ne reconnaissent aucun caractère accentué comme alphabétique, ce qui est majeur dans le cas de validations de fichiers d'adresses ou de noms propres dans les bases de données les plus communément utilisées.

2.3.2 La recherche et les comparaisons.

La recherche dans un texte d'une chaîne de caractères comportant des graphèmes accentués est assez délicate, compte tenu des environnements actuels et futurs. En effet, pour les fins de la recherche, au sens strict, un caractère non accentué n'est pas égal au même caractère accentué.

Cependant, tout comme il peut être intéressant d'effectuer une recherche en considérant ou non comme égales les minuscules et les majuscules dans un texte, pour discriminer plus ou moins, ou encore d'effectuer des recherches en ne considérant que des mots complets au lieu de parties de mots (ex.: la recherche du mot «fort» ne permet pas de pointer le mot «fortement», tandis que la recherche de la chaîne «fort» permet de le retrouver: il s'agit de deux opérations de recherche distinctes), il peut aussi être utile d'avoir une opération de recherche qui considère égales des chaînes comportant ou non les caractères accentués inclus dans la clé de recherche, dans un contexte où l'usage (ex.: la recherche de la chaîne «cle» pourrait permettre de retrouver indifféremment les mots «clé» et «clef», deux façons orthographiquement correctes de désigner le même objet) ou une simple erreur d'orthographe aurait pu faire oublier l'entrée de l'accent, que ce soit en majuscules ou en minuscules.

Ceci ne doit toutefois pas faire perdre de vue que l'on doit prévoir des mécanismes spéciaux pour la recherche précise de caractères accentués.

D'autre part, il faut prévoir un traitement spécial pour les diphtongues: la ligature «oe» peut se retrouver dans un texte sous forme d'un ou deux caractères, en majuscules ou en minuscules, selon l'usage ou les contraintes techniques.

En plus du français où les lettres doubles sont peu utilisées, notons que dans certaines langues comme le danois et l'allemand, certains graphèmes modernes ont une valeur double: par exemple, il est équivalent en danois d'écrire Århus et Aarhus. Une recherche correcte devrait considérer comme égales la lettre Å et les lettres Aa en danois. Chaque langue possède ses particularités et les opérations de recherche devraient donc toujours être incluses dans un module unique isolé du reste du logiciel pour permettre une adaptation rapide et facile selon différents usages nationaux.

2.3.3 Les conversions.

Outre les conversions requises entre divers jeux de caractères, en particulier pour la communication textuelle entre appareils incompatibles, il peut arriver qu'il y ait à effectuer des conversions temporaires dues à des limitations matérielles. Par exemple, en allemand, il peut être impossible d'afficher ou d'imprimer le caractère «ü» sur une pièce d'équipement donnée, mais à la rigueur, l'usage permet de le remplacer par «ue». Il est cependant toujours essentiel de conserver la «signification» intégrale dans les fichiers.

Dans le même ordre d'idée, dans certaines langues, des graphèmes simples en minuscules doivent être représentés comme lettres doubles en majuscules. Par exemple, en allemand, le «s» dur, qui ressemble à la lettre grecque bêta en minuscules, est représenté en majuscules invariablement comme «SS», la minuscule n'ayant pas d'équivalent majuscule simple. Ceci ne signifie toutefois pas que l'on puisse systématiquement considérer deux «S» qui se suivent en majuscules comme un «s» dur en minuscules.

L'opération de conversion majuscules-minuscules étant essentielle dans beaucoup de fonctions informatisées, il est nécessaire d'y porter une attention toute spéciale.

On ne répétera jamais assez dans le cas de conversions de minuscules à majuscules, qu'il ne faut sous aucun prétexte éliminer les accents pendant le traitement, même si en sortie, comme nous venons de le dire, il est préférable de les éliminer par une conversion simple plutôt que d'imprimer du «garbage». Ceci constituerait une perte d'information qu'il serait très difficile de reconstituer dans le cas d'une reconversion en minuscules subséquente du même texte. Ce genre de conversion et de reconversion est fréquent, même dans les systèmes les plus classiques. Voilà une raison de plus qui milite en faveur des majuscules accentuées si controversées dans certains milieux!

La modularité est encore ici de rigueur.

2.3.4 Les tris.

Les tris constituent une autre manoeuvre délicate, et même beaucoup plus délicate qu'on le croit. En effet, les logiciels parmi les mieux conçus pour un usage international considèrent généralement comme égaux les caractères accentués et les caractères non accentués correspondant pour les besoins du tri. C'est déjà beaucoup mieux que les logiciels purement anglophones, qui considèrent les caractères accentués comme non-alphabétiques et qui les trient donc de façon erronnée.

Cette façon «améliorée», ignore cependant les particularités de classement propres à chaque langue.

En français, par exemple, servons-nous d'un dictionnaire pour faire quelques observations: dans le cas où des lettres accentuées doivent être classées, on ne tient pas compte des accents, sauf dans le cas des homographes absolus, c'est-à-dire des mots qui s'écrivent exactement de la même façon sans accents. Dans ces cas, c'est la fin du mot qui détermine le classement, selon un ordre de priorité des accents qui obéit à la symétrie; le caractère non accentué précède le caractère affecté d'un accent aigu, qui vient, dans l'ordre, avant l'accent grave, l'accent circonflexe et les trémas. Cependant il est à noter soigneusement que, pour suivre la même logique de classement que les mots de longueurs différentes, dont la distinction s'effectue à la fin des mots, le classement nécessaire de ces homographes, accentués différemment, est déterminé en fonction de la dernière différence dans le mot, et non en fonction de la première [17]. Ainsi, pour illustrer complètement ces règles, «du» vient avant «dû», «pêche» vient avant «péché», «pécher» vient avant «pêcher», «pêcher» vient avant «pechère»[6]. On voit toute la subtilité nécessaire à un tri strictement correct des mots français.

Dans d'autres langues, les séquences de tri sont tout à fait différentes: en danois, par exemple, si le «A» vient bien sûr avant le «Z», le «Å» vient après, sans compter les autres lettres danoises affectées d'un signe diacritique, dont l'ordre n'est pas du tout évident pour un non-Danois [3]. D'autre part, les lettres doubles (par exemple les ligatures comme «æ», le «å» danois, etc.) doivent être classées avec les doubles lettres correspondantes (respectivement «ae» et «aa» dans les exemples précédents).

Il n'existe donc pas de tri universel et une révision des modules de tri est à prévoir pour chaque pays où un logiciel prévoyant cette fonction doit être publié. L'usage bilingue canadien, si l'on restreint la discussion au français et à l'anglais, doit utiliser un tri français au lieu d'un tri basé sur l'anglais, ce qui sera alors correct dans les deux langues.

3. Particularités linguistiques et culturelles.

3.1 Les coupures syllabiques.

Si l'on doit effectuer des coupures syllabiques dans un texte, un message à l'écran ou ailleurs, il ne faut pas oublier que les règles sont différentes d'une langue à l'autre, en particulier entre le français et l'anglais: il y a plus de règles dans ce dernier cas. Il s'agit d'une des rares fonctions où l'anglais est grammaticalement plus complexe que le français. C'est pourquoi les logiciels anglophones ont tendance à recourir à un dictionnaire pour cette fonction, au lieu de prévoir un algorithme simple, comme on pourrait le faire en français [7]. Dans ce dernier cas, on peut hautement améliorer la rapidité de traitement et l'espace requis en prévoyant ces quelques règles dans un court algorithme. Encore une fois ici, la modularité est toujours de rigueur.

3.2 Les règles grammaticales dans les messages variables.

Il arrive que des éléments de messages fixes soient concaténés pour former un message complet dans les systèmes informatiques. Dans ces cas, excepté en anglais où pratiquement les seuls accords grammaticaux se limitent en général aux pluriels et aux adjectifs possessifs, il faut être à l'affût que les règles grammaticales doivent être respectées dans les algorithmes de composition des messages. Ces règles varient beaucoup d'une langue à l'autre et comprennent des notions variées qui peuvent être inexistantes dans d'autres langues: ainsi les «genres» sont pratiquement très rares en anglais alors qu'ils sont très importants en français de même que le «vous» et le «nous» de politesse, alors que les déclinaisons, inexistantes en français, sont fréquentes dans certaines autres langues européennes (toujours à l'exception de l'anglais).

Certaines règles sont passablement subtiles. Comme il existe de «faux amis» dans le vocabulaire de deux langues d'origine commune (comme le mot «actual» de l'anglais, qui se traduit par «réel» en français, est le «faux ami» du mot français «actuel», qui se traduit par «present» en anglais; on peut aussi citer le mot «billion» en anglais américain, qui se traduit par «milliard» alors que le mot «billion» du français indique un million de millions, soit 1000 fois plus), il existe des «quasi-ennemis» dans les grammaires étrangères qui ont un impact sur le développement des logiciels. Ainsi, par exemple, est considéré comme pluriel en anglais tout ce qui excède l'unité, alors qu'en français le pluriel commence à partir de deux unités. On sait que l'on doit traduire «1,5 pomme» par «1.5 apples» en anglais. Il y a fort à parier que de telles différences, si subtiles soient-elles, sont à prévoir pour chaque langue différente de celle qu'utilise un concepteur.

Les concepteurs de logiciels prévoyant produire des messages comportant plusieurs éléments variables dont le contenu final ne sera connu qu'en cours d'exécution des programmes ont tout intérêt à isoler dans un module spécial facilement repérable et bien documenté l'algorithme de composition de ces messages.

3.3 Les types de données spéciales.

Certains types de données universelles et à valeur quantifiable sont représentés différemment, selon le pays ou même la région [8]. Les types les plus communs entrant dans cette catégorie sont:

- les dates (présentation numérique)
- les noms des jours
- les noms des mois
- les heures
- les séparateurs décimaux
- les devises monétaires
- les couleurs
- les unités de mesures
- les nombres (forme textuelle) [voir 3.2 «Les règles grammaticales...»]

et finalement, un type bien particulier, propre à l'informatique:

- les tables de bits et les algorithmes nécessaires à la représentation graphique du répertoire de caractères et des types de données ci-dessus énumérés dans les logiciels produisant des sorties sur traceurs, écrans ou imprimantes en mode graphique.

Par ailleurs, il existe aussi des données non quantitatives à portée internationale, pouvant être intégrées dans les logiciels, et dont le format, s'il est figé, risque de nuire à la commercialisation d'un produit exportable. Il s'agit des numéros de téléphone, des codes postaux et autres données d'adresse, dont le positionnement, la taille, les délimiteurs et même l'identification (songer au «zip code» américain) varient de pays en pays.

Ces éléments doivent eux aussi être isolables pour une adaptation acceptable dans le pays d'adoption d'un logiciel.

3.4 Une fonction globale très importante.

Il est important autant pour le développement de nouveaux logiciels que pour l'implantation de logiciels développés pour un usage international que le système d'exploitation choisi soit normalisé pour permettre de savoir dans quelle langue, dans quel pays et dans quelle région le logiciel est utilisé, ce qui peut permettre la conception de logiciels comprenant déjà des éléments multilingues et culturels dynamiquement configurables au niveau de l'utilisateur final. La façon de reconnaître cet environnement doit être fournie sous forme d'une fonction documentée du système d'exploitation. De même, il doit aussi exister une fonction de système qui permette de modifier ce type d'environnement en cours d'exécution d'un logiciel. Dans le cas des systèmes multi-utilisateurs, l'environnement linguistique et culturel doit pouvoir être différent d'un utilisateur à l'autre.

4. La modularité des éléments textuels.

Il a été amplement question au cours des pages précédentes de la modularité des éléments fonctionnels des logiciels. Il existe une autre dimension à la modularité qui consiste à rendre indépendants tous les éléments textuels des programmes. On peut considérer comme éléments textuels non seulement les messages et les réponses anticipées, mais aussi les noms de variables que l'utilisateur a la liberté de choisir, de même que les mots réservés ou noms de commandes des langages informatiques, en plus des sous-programmes dépendants de la représentation interne des caractères. Nous discuterons à ce chapitre des considérations ayant trait à chacun de ces éléments.

4.1 Les messages.

Il va sans dire qu'aucun message ne devrait faire partie des programmes eux-mêmes, mais bien être inclus dans un fichier séparé qui pourra facilement être traduit directement en une autre langue ou même selon les coutumes du pays ou de la région visée (par exemple, en Angleterre, on se plaint souvent que l'anglais américain utilise des expressions difficilement compréhensible par la majorité des Anglais). Une véritable architecture multilingue impose que la structure de tels fichiers soit documentée et que leur modification soit permise au niveau de l'utilisateur final. De même, elle doit aussi permettre un mécanisme de migration sans changement des messages déjà modifiés par l'utilisateur, dans les cas où une nouvelle version du logiciel est offerte sur le marché, et ce, même si de nouveaux messages sont ajoutés.

Il est important de mentionner ici qu'il est fréquent que des programmes soient conçus pour lire les messages générés par d'autres logiciels et prendre action en conséquence. Pour préserver l'indépendance de tels programmes par rapport à la langue, il est hautement recommandable que tout message généré par un logiciel comporte un préfixe unique, non modifiable dans le processus de traduction ou d'adaptation par l'utilisateur. Ce préfixe sera généralement un numéro de message qui pourra être retrouvé dans la documentation des messages, et ce, quelle que soit la langue de cette dernière documentation. Ceci aura de plus comme effet secondaire de permettre la compréhension éventuelle d'un message par un utilisateur qui ne connaîtrait pas la langue d'affichage du logiciel, mais qui disposerait d'une documentation dans sa propre langue pour le même produit.

4.2 Les réponses anticipées.

D'autre part, il y a aussi lieu de séparer, dans un autre fichier, les réponses anticipées (de type «oui», «non», etc.) aux questions posées par un lociciel à son utilisateur. Ce fichier devrait, lui aussi, pouvoir être modifié pour s'adapter aux coutumes locales, ou tout simplement pour permettre l'adaptation à une organisation spécifique et à la terminologie qui lui est propre.

4.3 Les noms de variables.

Il doit être possible, dans le cas de logiciels nécessitant que l'on choisisse des noms pour identifier des objets, des concepts ou toute variable, que l'utilisateur fasse le choix des dénominations à l'aide du répertoire complet de graphèmes de sa langue, incluant les caractères accentués ou dotés d'un signe diacritique quelconque.

4.4 Les langages informatiques et les noms de commandes.

La conception de certains logiciels engendre parfois la nécessité de créer un langage informatique précis (par exemple, les types d'énoncés des compilateurs et interpréteurs destinés à la programmation classique, les commandes dans les tableurs et les logiciels de traitement de texte). Ces langages, qu'ils utilisent des mots réservés ou non, ne devraient pas être implantés dans une autre langue avant que toutes les commandes aient été traduites par le concepteur ou ses représentants directs. Dans ce cas précis, et malgré le fait qu'on peut toujours laisser à l'utilisateur le soin de modifier lui-même les noms de commandes, il y a nécessité que tous les usagers d'une même langue puissent échanger entre eux leurs applications de façon normalisée. Ceci facilite de plus l'élaboration de programmes de formation cohérents.

De plus, le partage d'applications entre utilisateurs de langues différentes nécessite que le codage interne des commandes dans les fichiers soient commun, quelles que soient les versions du même logiciel. Si ceci vaut pour assurer la compatibilité des versions entre elles, même pour un logiciel monolingue, cela vaut à plus forte raison pour des logiciels à portée multilingue. Il va sans dire que l'affichage des commandes comme telles devrait toujours être fait en utilisant le répertoire des commandes propre à la langue de l'utilisateur, quelle que soit la langue d'origine d'une application.

4.5 Les traitements dépendants de la représentation interne.

Il est essentiel d'assurer une totale indépendance du traitement par rapport au jeu de caractère utilisé. Ceci implique que tout traitement dont la logique est dépendante de constantes textuelles soit isolé du corps des programmes, si on prévoit transporter un logiciel entre diverses architectures de machines, ce qui entraînera alors plus qu'une simple conversion des caractères utilisés dans les logiciels.

Pour mieux comprendre de quoi il s'agit ici, voici un exemple très simple: si on compare une variable à la constante «0» pour déterminer laquelle est la plus petite, le résultat sera différent selon que l'on utilise une table de représentation ASCII ou EBCDIC.

5. Claviers, écrans, imprimantes et leurs interfaces.

On trouvera à la section 7 un répertoire des normes nationales et internationales concernant en particulier la disposition des touches sur les claviers, de même que les tables de représentation binaire normalisées pour les jeux de caractères.

Généralement, les types d'écran sur le marché, ainsi que les imprimantes, sont conformes aux normes élaborées il y a quelques années. Les normes plus récentes supportant des jeux de caractères étendus, les seules à se prêter adéquatement aux applications multilingues, sont encore peu implantées. On a recours plutôt à des normes «de facto», qui abondent sur le marché: on ne peut donc dire aujourd'hui encore que l'industrie se conforme aux normes internationales.

Entretemps, il faut se référer à la documentation des divers constructeurs pour le soutien multilingue des écrans et imprimantes de conception récente ou à d'autres sources pour une comparaison des divers jeux de caractères [8].

Il est acceptable et parfois nécessaire (en particulier pour le traitement de texte) que les logiciels d'application contiennent des interfaces particulières pour chaque type d'imprimante ou d'écran, pour le contrôle des effets spéciaux (fontes ou couleurs spéciales, soulignés, espacement proportionnel, etc.). Il est cependant préférable de laisser au système d'exploitation le soin d'effectuer les conversions de caractères nécessaires pour assurer la compatibilité de l'ordinateur avec le périphérique visé. Le système d'exploitation peut en effet déjà inclure un tel outil de conversion, et il faut éviter les traductions à répétition du même caractère, ce qui peut donner des résultats erronnés en sortie.

Pour ce qui est des claviers, qui constitueront encore longtemps d'être les unités d'entrée de données les plus utilisées, il faut se garder de la tentation de les soutenir directement dans les logiciels d'application, même si les technologies nouvelles permettent facilement leur reconfiguration dynamique. Le soutien des claviers est une fonction essentielle des systèmes d'exploitation, et c'est à ce niveau, et à ce niveau seulement, qu'il doit être fait. Ceci garantit que l'utilisateur n'aura pas à s'adapter continuellement à d'autres dispositions de clavier pour entrer ses textes, surtout s'il utilise physiquement toujours le même poste de travail.

Comme le soutien des langues autres que l'anglais est souvent déficient à l'heure actuelle, on doit parfois recourir à des interfaces de claviers reconfigurant ce dernier selon différents usages. Il vaut mieux alors avoir recours à ces solutions uniques (pour un poste de travail donné) au niveau du système d'exploitation qu'à des solutions multiples intégrées aux différents logiciels d'application. En parallèle, il convient que les logiciels acceptent tels quels (sans les filtrer!) les caractères générés par ces interfaces (voir à la section 2.2 la discussion concernant les utilitaires de UNIX) ou par le système d'exploitation lui-même, si le soutien linguistique propre au pays y est déjà intégré. De nombreux programmes américains de conception archaïque sur PC effectuent de malencontreux filtrages qui nuisent à leur portée internationale; d'autre part, il n'est pas rare de rencontrer des systèmes classiques sur ordinateurs de grande puissance qui effectuent de telles opérations à proscrire, ne serait-ce que pour empêcher les programmes COBOL d'invalider les données (voir section 2.3.1).

6. La documentation et la terminologie utilisée.

Le logiciel constituant désormais un facteur culturel important qui peut favoriser les échanges entre peuples de langues différentes et, à plus forte raison, entre peuples partageant une même langue mais géographiquement éloignés, comme c'est le cas des francophones d'Amérique par rapport au reste de la francophonie, il est de la plus haute importance que les concepteurs de logiciels soignent la terminologie utilisée dans la documentation qu'ils élaborent.

Il va de soi que la documentation, tout comme les autres éléments textuels composant un logiciel complet, doit être écrite dans la langue de l'utilisateur. La terminologie utilisée, tant dans la conception d'un logiciel que dans sa traduction ou son adaptation pour un pays, doit refléter une langue internationale. Les traductions de documentations américaines en français effectuées parallèlement ici et en Europe ont déjà fait naître des néologismes aux deux endroits qui risquent de créer des divergences d'habitudes, difficiles à corriger, qui nuisent à la communication internationale. Il faut éviter cette situation au Canada, même si nous n'avons que peu d'emprise sur les pratiques du reste de la francophonie.

En microinformatique, par exemple, la naissance du concept des «electronic spreadsheets» a fait apparaître un néologisme en Europe, le mot «tableur», qui représente très bien cette nouveauté, alors que le Ministère de l'Éducation du Québec a répandu le terme «chiffrier électronique», traduction libre rendant de façon incorrecte l'idée du tableur (un chiffrier ne servant qu'à un usage de contrôle de la concordance entre un journal comptable et un grand livre [6]). Dans d'autres cas, les néologismes québécois sont à préférer aux traductions libres européennes (ex.: une «bogue», transcription normalisée en France du «bug» américain [13], terme auquel on devrait préférer «erreur» [9]).

Heureusement, dans l'ensemble, le Québec est généralement au diapason de la francophonie sur la plupart des aspects terminologiques des technologies de l'information (incluant la bureautique et l'informatique), grâce à de bons échanges internationaux et à l'excellent degré d'expertise développé par l'Office de la langue française (OLF) du Québec, une autorité francophone mondiale en matière de terminologie (voir en particulier [10] et [11]). Il est à noter que l'OLF offre un accès public à sa banque de terminologie informatisée.

Le secrétariat d'état canadien, responsable de la terminologie au gouvernement fédéral, a lui aussi publié plusieurs outils terminologiques ([14], [15]), en plus d'offrir les services de sa banque de terminologie aux traducteurs canadiens. Ce même organisme contribue également, par ses nombreux contacts, à l'avancement de la terminologie au niveau international et assume actuellement la présidence du comité de terminologie de l'ISO concernant le traitement de l'information. Ce comité a déjà publié, jusqu'à présent, plus de 20 lexiques en matière d'informatique [16].

Il est essentiel que les Canadiens et les Québécois utilisent les publications et services de tels organismes oeuvrant dans le secteur de la terminologie. Ce sont des moyens majeurs dans un environnement anglophone comme l'Amérique du Nord, d'autant plus que nous sommes éloignés des autres peuples francophones. L'aspect des échanges internationaux soutenus est très important à cet égard.

Dans le même ordre d'idées, il importe, lors de la traduction d'un logiciel en d'autres langues, de s'adjoindre des traducteurs qui ont un lien direct avec la culture actuelle du pays auquel est destiné le logiciel, et de préférence avec une communauté nationale familière avec la sphère d'activité dans laquelle s'inscrit le logiciel, pour éviter le même type de problème qui a été mentionné plus haut. En effet, si un logiciel est traduit ici dans une langue étrangère, il importe que les contacts du traducteur avec le pays où se parle cette langue aient été soutenus jusqu'au moment de la traduction. La culture évolue de façon beaucoup plus rapide en ce siècle qu'au cours des siècles passés, et la terminologie de chaque sphère d'activité évolue presqu'au même rythme, et ce, parallèlement, dans chaque langue.

La terminologie n'est pas simplement une question de vocabulaire, mais d'expressions préfabriquées (exemples de telles expressions dans le français usuel international: «contre toute attente», «en désespoir de cause», «comme dans du beurre» [12]) et même de sigles et d'acronymes de toutes sortes véhiculés dans le quotidien et largement répandus par les médias. Ainsi, le néologisme «SIDA», inconnu il y a dix ans, nécessite un contact régulier avec la culture anglo-saxonne pour qu'on puisse le traduire de façon significative en anglais par «AIDS». Ce terme, connu de tous, n'a plus besoin d'explication aujourd'hui dans les médias, mais sa traduction n'est pas évidente pour quelqu'un qui a appris l'anglais (ou une autre langue) il y a dix ans et plus. L'informatique et la bureautique étant aujourd'hui présentes dans toutes les sphères de l'activité humaine, elles ne sont pas exemptes de ces obligations culturelles.

7. Les normes nationales et internationales.

Comme il a été dit précédemment, les pratiques actuellement implantées dans l'industrie informatique tiennent le plus souvent compte de normes en usage depuis longtemps. Ces normes n'ont pas toutes été révisées pour tenir compte des facteurs linguistiques, et plusieurs nouvelles normes en voie d'acceptation n'en sont encore qu'au stade d'ébauche ou de norme préliminaire.

Les organismes normatifs oeuvrant au niveau international et se préoccupant des technologies de l'information sont l'ISO et le CCITT. Dans chaque pays important, il existe un organisme de normalisation qui révise les normes internationales (tout en participant à l'élaboration de ces dernières) pour y apporter ou non des modifications «nationales». L'organisme officiel attitré pour accomplir cette fonction en notre pays est le Conseil canadien des normes. L'Association canadienne de normalisation (ACNOR - en anglais CSA) a quant à elle pour tâche de publier ces normes (et parfois de valider leur application).

Cette section pourra sembler un peu plus rébarbative que les autres, mais il est important que les concepteurs de logiciels aient une idée des difficultés de la normalisation en informatique, processus volontaire en constante évolution, avant d'effectuer des recherches plus approfondies. Pour les besoins du présent guide, nous avons souligné les normes qui nous semblent pertinentes dans l'élaboration d'une architecture linguistique des technologies de l'information. Sauf indication contraire, ces normes proviennent de l'ISO. Il est à noter que les normes marquées «DIS (Draft International Standard)» ont un statut de normes préliminaires, sujettes à modifications mineures dans un proche avenir. Les normes marquées «DP (Draft Proposal)», quant à elles, ne sont pas encore reconnues comme véritables normes parce qu'elles n'en sont qu'à la première étape des discussions internationales.

7.1 Normes touchant les jeux de caractères:

Norme n°	Titre	Méthode utilisée
646-1983	Traitement de l'information - jeu ISO de caractères codés à 7 éléments pour l'échange d'information.	Cette norme prévoit dix positions pour des caractères nationaux. La norme canadienne Z243.4-1973 utilise ces positions pour les caractères suivants: àâéèêôùûç et î ou É.
963-1973	Traitement de l'information - Guide pour la définition de jeux de caractères à 4 éléments dérivés du jeu de caractères à 7 éléments pour l'échange d'information entre matériels de traitement de l'information.	Cette norme ne prévoit aucune minuscule ni aucun caractère accentué
4873-1979 (en révision)	Traitement de l'information - Jeu de caractères codés à 8 éléments pour l'échange d'information.	Cette norme est une réplique de la norme 646 mais pour un caractère codé à 8 éléments. Les autres codes disponibles ne sont pas définis.

Norme n°	Titre	Méthode utilisée
5426-1980	Extension du jeu de caractères latins codés employé pour l'échange d'information bibliographique	Cette norme sépare les accents des lettres et donne accès à pratiquement toutes les langues utilisant l'alphabet latin tout en permettant la translitération des langues non latines.
DIS 6937/1-1983 DIS 6937/2-1983 DIS 6937/3-1986	Information processing - Coded Character sets for Text Communication Part 1: General Introduction. Part 2: Latin Alphabetic and Non-Alphabetic Graphic Characters Part 3: Control Functions for Page-Image Format.	Cette ébauche définit dans dans la partie 2 le jeu de caractères requis pour l'utilisation de codes distincts pour les accents et autres signes diacritiques.
DIS 8859/1-1985	Information Processing - 8 bit Single Byte Coded Graphic Character Sets. Part 1: Latin Alphabet No. 1	Cette ébauche définit un jeu de caractères utilisant des codes uniques pour les caractères accentués.

Au Canada, certaines normes ont été adoptées en ce qui concerne les jeux de caractères:

Z243.4-1985	CSA Preliminary Standard - 7-bit and 8-bit Coded Character Sets for Information Processing and Interchange.	Cette norme utilise des codes uniques pour les caractères accentués. Deux tables de caractères accentués y sont définies. Seule la table définissant l'Alphabet latin No 1 a une portée internationale.
T500-1983	Syntaxe du protocole de la couche présentation du vidéotex/télétex NAPLPS.	Cette norme utilise des codes séparés pour les accents et les lettres.

7.2 Normes touchant les techniques d'extension de codes.

En plus des normes sur les jeux de caractères, l'ISO a défini des normes régissant l'utilisation des tables elles-mêmes. Notons que la technique d'extension de codes à 7 bits n'est pas recommandée par le présent guide à l'intérieur des logiciels d'application (voir 2.2), en raison de sa complexité, génératrice d'erreurs multiples. Pour des applications précises nécessitant des jeux contenant plus de 256 caractères, il peut cependant être nécessaire d'utiliser des techniques d'extension de codes à 8 bits. Les normes traitant de ces aspects sont les suivantes:

Norme n°	Titre	Méthode utilisée
2022-1982	Traitement de l'information - Jeux ISO de caractères codés à 7 et 8 éléments - Techniques d'extension des codes.	Norme permettant le passage d'une table de codes à une autre.
2375-1980	Traitement de l'information - Procédure pour l'enregistrement des séquences d'échappement.	Norme définissant la procédure d'appel des différentes tables de codes.
7350-1984	Transmission de texte - Procédure d'enregistrement des sous-répertoires de caractères graphiques.	Registre des différents répertoires de caractères enregistrés.

7.3 Normes concernant les langages de programmation.

Bien que les normes qui suivent soient restrictives, puisqu'elles imposent des limites sur les répertoires de caractères permis, il importe que les concepteurs soient avisés de leur existence, qui peut affecter d'une certaine façon les aspects linguistiques, jusqu'au moment d'une éventuelle révision. Il faut souligner que certains constructeurs ne respectent pas intégralement ces normes, ce qui peut avoir un effet positif ou neutre en ce qui concerne les considérations linguistiques.Ces normes concernent différents langages de programmation (voir 2.3.1):

Norme n°	Langage
1538-1984	ALGOL 80
1539-1980	FORTRAN
1672-1977	ALGOL
1989-1985	COBOL
6160-1979	PL/1
6522-1985	(sous-ensemble de 6160)

7.4 Normes relatives à la reconnaissance formelle des caractères.

Les normes qui suivent définissent les formes et cotes de caractères utilisés spécifiquement pour la reconnaissance optique:

Norme n°	Titre
1073/1-1976	Jeux alphanumériques de caractères pour la reconnaissance optique. Partie 1: Jeu de caractères ROC-A - Formes et cotes de l'image imprimée.
1073/2-1976	Partie 2: Jeu de caractères ROC-B - Formes et cotes de l'image imprimée.

7.5 Normes relatives aux claviers.

Toutes les normes qui sont répertoriées ici concernent la disposition des touches. Au Canada, une seule norme définissant un clavier bilingue est officielle. Elle ne s'applique qu'aux machines à écrire et elle n'a pas encore été implantée à la connaissance des auteurs du présent guide: il s'agit de la norme Z243.19-1985, intitulée «Correspondence Typewriter Keyboard Layout for the English and French Languages; A National Standard of Canada».

Il est à souligner qu'un comité officiel travaille actuellement à l'élaboration d'une norme pan-canadienne de clavier bilingue pour la bureautique et l'informatique. Nous recommandons fortement aux lecteurs de se tenir au courant de la publication de cette norme préliminaire, prévue pour 1987. On pourra contacter l'un ou l'autre des auteurs du présent guide pour être informé de l'état d'avancement de ces travaux.

Voici maintenant une liste des normes internationales concernant les claviers. Ces normes nécessitent toutefois des adaptations nationales:

Norme n°	Titre
1091-1977	Machines à écrire - Disposition des touches d'impression et des touches de fonction.
2126-1975	Machines de bureau - Disposition de base pour la section alphanumérique des claviers manoeuvrés avec les deux mains.
2530-1975	Clavier pour échanges internationaux en traitement de l'information employant le jeu ISO de caractères codés à 7 éléments - Zone alphanumérique.
3243-1975	Claviers pour les pays dont la langue utilise des caractères alphabétiques supplémentaires - Guide pour l'harmonisation.
3244-1984	Machines de bureau et machines employées en traitement de l'information - Principes directeurs pour l'affectation des touches de commande sur les claviers.
DP 8884-1985	Information Processing - Text Preparation and Interchange - Keyboard Layout for Multiple Latin - Alphabet Languages.

7.6 Normes à incidences culturelles.

Certaines normes informatiques touchent plus ou moins quelques aspects présentés dans le présent guide, en particulier la structure des documents et la représentation des dates, de l'heure et des devises monétaires. On les désigne dans le jargon de la normalisation sous la rubrique générale de «Information Processing - Text and Office Systems». En voici quelques unes, représentatives à cet égard:

Norme n°	Titre
DP 8613-1985	Text and Office Systems - Document Structures - Part 1: General Introduction Part 2: Office Document Architecture Part 3: Document Layout and Imaging Processes Part 4: Document Profile Part 5: Office Document Interchange Format Part 6: Character Content Architecture.
3307	Information Interchange - Representations of Time of the Day.
4031	Information Interchange - Representation of Local Time Differentials.

Finalement en matière de terminologie, l'ISO a aussi normalisé un certain nombre de lexiques, comme il a été mentionné à la section 6 [16].

8. Conclusion.

On aura constaté tout au long de ce guide, que les idées soulignées dans le texte sont les points auxquels les concepteurs pourront se référer rapidement comme liste de contrôle pour la réalisation de logiciels facilement adaptables dans la majorité des langues européennes.

Cette même liste de contrôle pourrait aussi être utilisée pour l'évaluation des aspects linguistiques par les acheteurs avertis de logiciels qui ont à prendre en considération ces caractéristiques, pour une raison ou une autre.

Heureusement pour les chercheurs et les perfectionnistes, la porte reste ouverte à ceux qui voudront accomplir la noble tâche d'enrichir ce travail pour s'attaquer aux problèmes et aux solutions propres aux langues utilisant un alphabet non latin (arabe, hébreu, katakana) qui s'écrivent de droite à gauche ou de bas en haut, de même qu'aux problèmes cruciaux de l'écriture de textes bilingues ou multilingues où la direction de l'écriture n'est pas la même dans toutes les langues d'un même document, ou encore pour revoir les particularismes graphiques de l'écriture qui dépendent du contexte dans certaines langues alphabétiques (ex.: la lettre grecque «sigma», qui ne s'écrit pas de la même façon en fin de mot que dans le reste de l'écriture; les lettres arabes qui s'écrivent de diverses façons, en fonction de leur position à l'intérieur des mots, etc.).

Le sentier est aussi ouvert dans le traitement des langues idéographiques telles que le chinois. Il y a là de nombreux problèmes intéressants et d'autant plus utiles qu'ils touchent plus du tiers de la population de cette planète.

Ces secteurs relèvent plus d'un domaine d'avant-garde connu sous le nom controversé mais généralement reconnu d'«intelligence artificielle». D'ici quelques années, il est à prévoir que les fruits de ce secteur de recherches seront disponibles sous forme de produits intégrables aux logiciels classiques actuels. Ces produits devraient permettre de faciliter la solution des problèmes reliés aux nombreuses différences structurelles séparant les diverses langues de la terre.

Il est à espérer qu'une telle ébauche d'architecture universelle des langues nationales (ou internationales) intéressera suffisamment de groupes pour qu'un besoin de standardisation se fasse sentir au niveau des organismes de normalisation tels que l'ACNOR au Canada, l'ISO ou le CCITT au plan international.

D'ici là, pour terminer sur une note réaliste et humoristique, il ne nous reste qu'à tirer le meilleur parti de l'héritage historique que les temps bibliques nous ont légué avec la légende de la Tour de Babel!

Annexe. Bibliographie.

[1] Lexique des règles typographiques en usage à l'imprimerie nationale [de France], 2e édition, 1975, p.7.

[2] Ibid., p. 86.

[3] White Paper on national character, language and keyboard problems, SEAS, Nijmegen, Pays-Bas, septembre 1985.

[4] Norme Z243.4-1985, ACNOR, 1985 (voir table «Alphabet latin No 1»).

[5] Équipements terminaux pour les services de télégraphie et de «télématique». Avis des séries S et T (Commission VIII), CCITT, Genève, 1981.

ou encore:

Teletex, A worldwide link among office systems for electronic document exchange, IBM Systems Journal, vol. 22, nos 1 et 2, 1983.

[6] Petit Larousse illustré, Larousse, 1981.

[7] Le bon usage, Grevisse (M.), Duculot, 1975, 10e édition.

[8] Les aspects techniques de l'utilisation du français en bureautique, Alain LaBonté, Ministère des Communications du Québec, 29 avril 1986.

[9] Vocabulaire de la programmatique, Stella Abensur, Services linguistiques, IBM Canada Ltée, Montréal 1985.

[10] Vocabulaire du micro-ordinateur, Office de la langue française, 1986.

[11] Terminologie de l'informatique, Office de la langue française, 1983.

[12] Lexicon - Grammar, the Representation of Compound Words, Maurice Gross (Université Paris 7), Proceedings of COLING '86, International Committee on Computational Linguistics, Bonn 1986.

[13] Les mots de la micro, Alan Freedman et Bernard Sauteur, P.S.I., 1985.

[14] Glossaire d'informatique, Gouvernement du Canada, Bulletin de la technologie de l'information, juillet 1981.

[15] Vocabulaire de la technologie du traitement de l'information, Gouvernement du Canada, Bulletin de la technologie de l'information, mars 1984.

[16] ISO 2382, Traitement de l'information - Vocabulaire.

[17] Règles du classement alphabétique en langue française et procédure informatisée pour le tri, Alain LaBonté, Ministère des Communications du Québec, 6 janvier 1987.

QUEEN QA 76.6 .L33 1987
Labonté, Alain
Guide conceptuel pour la ré

CACC / CCAC
88683

Gouvernement du Québec
Ministère des Communications

Imprimé au Québec, Canada