Informatique et langue japonaise

par Dominique GIROUX

Les premiers ordinateurs ont été conçus essentiellement pour effectuer rapidement des calculs scientifiques, impossibles à réaliser jusque-là. Très vite s'est fait sentir le besoin de pouvoir traiter aussi du texte, afin de répondre aux exigences de la gestion des grandes entreprises et des administrations. A cette époque, dans les années 50, seul l'usage de l'anglais, c'est à dire les vingt-six lettres de l'alphabet sans aucun signe diacritique, était possible, et souvent uniquement en majuscules ! On reconnaît bien là l'origine américaine de ces machines. Cependant, ces vingt-six signes avaient, et ont toujours, l'avantage de représenter un tronc commun pour un bon nombre de langues occidentales. Par contre, pour les langues orientales et plus particulièrement celles qui font appel à plusieurs milliers de caractères, le problème restait entier. Au Japon commencèrent alors de longues recherches qui, vers la fin des années 60, débouchent sur un procédé de conversion automatique des *kana* en caractères chinois et, dans les années 70, à l'établissement de normes pour le traitement de l'information.

Les normes de codage

Les constructeurs américains qui avaient au début presque tous mis au point leur propre système de représentation numérique des caractères utilisés en anglais, mirent finalement au point[1] un code commun, le code ASCII (American Standard Code for Information

[1]Certains auteurs en attribuent la paternité à Robert W. Bemer (1965). *Cf.* Collins (Rip), *Byte*, janv. 1990.

Interchange). Ce code utilise les 7 bits (le bit, contraction de *binary digit*, étant l'unité minimale d'information) de poids faible d'un octet (ensemble de 8 bits, *byte* en anglais), et peut donc prendre $2^7 = 128$ valeurs différentes (entre 0 et 127), ce qui permettrait de représenter un maximum théorique de 128 caractères. Les 32 premières valeurs (codes 0 à 31) étant réservées à des codes dits de contrôle (contrôle d'équipements périphériques : imprimante, modem, etc.), il ne reste plus que 94 caractères imprimables en pratique, auxquels il faut ajouter l'espace (code 32) et le caractère « del » (code 127). Ce nombre est suffisant pour l'anglais : minuscules, majuscules, chiffres de 0 à 9, signes de ponctuation plus quelques symboles du type &, $, *, @, #, y trouvent place, mais toujours pas nos chers accents ni notre cédille. Comment donc étendre les possibilités de codage ? En se servant du huitième bit, inutilisé par le code ASCII original : mathématiquement cet artifice double la capacité de codage ($2^8 = 256$) mais, pour les mêmes raisons que ci-dessus, toutes les valeurs ne sont pas utilisables.

En revanche, on voit immédiatement qu'il est toujours impossible de traiter les quelques milliers de caractères nécessaires pour écrire en japonais. Le problème allait être résolu par l'utilisation, cette fois, non pas d'un bit supplémentaire mais d'un deuxième octet. Après une période anarchique chez eux aussi[2], les japonais se mirent enfin d'accord sur le nombre de caractères à retenir et sur les codes correspondants. En 1978 l'Association de normalisation industrielle du Japon publie la norme JIS (Japan Industrial Standard) C6226 en s'appuyant sur la norme JIS C6228 de 1975, qui définit les techniques d'extension du code ASCII à 7 bits et permet ainsi d'utiliser 94 fois 94, soit 8 836 valeurs numériques différentes pour le codage[3]. On n'utilise en effet, comme pour le code ASCII, que les 7 bits de poids faible de

[2]C'est encore le cas en Corée du Sud où l'on compte au moins quatre systèmes différents de codage pour les micro-ordinateurs de type compatible IBM.

[3]Ce qui représente tout de même un sérieux gâchis : seulement 8 836 signes différents au lieu des $2^{16} = 65\,536$ théoriques !

chaque octet, tout en évitant les 33 premières valeurs ainsi que la dernière, ceci afin que chaque octet pris indépendamment corresponde à un code ASCII imprimable et visualisable. Deux « séquences d'échappement » permettent de faire la différence avec le code ASCII normal sur un octet : la première (Kanji In ou KI) indique que tous les octets suivants devront être traités 2 par 2, la seconde (Kanji Out ou KO), que l'on revient au codage sur un octet, donc à de l'ASCII.

La norme C6226 (1978) définit un jeu de caractères répartis en 2 niveaux, appelés JIS 第一水準 et JIS 第二水準 (JIS niveau 1 et JIS niveau 2), que l'on peut représenter sous forme d'un tableau de 94 lignes (*ku*) et 94 colonnes (*ten*). Cette norme a été modifiée en 1983 (JIS X0201), notamment au niveau des séquences d'échappement KI et KO.

Le niveau 1 comprend les signes de ponctuations et divers symboles (lignes 1 et 2 du tableau *kuten*), les chiffres arabes et l'alphabet anglais majuscule et minuscule (ligne 3), les *hiragana* (ligne 4), les *katakana* (ligne 5), les alphabets grec (ligne 6) et cyrillique (ligne 7) et 2 965 *kanji* classés dans l'ordre phonétique *gojûon* (lignes 16 à 47).

Le niveau 2 contient 3 384 caractères chinois d'usage moins fréquent ou des versions non simplifiées de certains caractères du premier niveau, classés selon l'ordre des 214 clefs défini par le dictionnaire chinois *Kangxi zidian* 康熙字典, (*kôki jiten* en japonais), et ensuite par nombre de traits (lignes 48 à 83).

On constate tout d'abord qu'il reste plusieurs cases vides dans ce tableau : les lignes 8 à 15 (752 cases) et 84 à 94 (1 034 cases). Un certain nombre (de l'ordre de la centaine, variable selon le constructeur) est laissé à la disposition de l'usager qui peut ainsi ajouter, grâce à un programme annexe, des caractères ne figurant pas parmi les 6 349 *kanji* retenus par la norme ; ce sont souvent des caractères utilisés pour des noms de personnes ou de lieux, mais on

peut aussi créer des symboles ou des logos particuliers[4]. Ensuite, c'est l'absence de tout signe diacritique (accents, cédille etc.) qui frappera, entre autres, un français alors que l'on se posera des questions sur la présence de l'alphabet cyrillique, celle du grec pouvant se justifier plus facilement[5].

Une variante de cette norme, appelée Shift-JIS, a été mise au point pour les micro-ordinateurs japonais fonctionnant sous le système d'exploitation MS-DOS. On utilise ici le bit de poids fort d'un octet pour indiquer qu'il s'agit du premier octet d'un caractère japonais, donc qu'il faut aussi prendre en compte l'octet suivant pour déterminer le caractère, ce qui permet de se passer des séquences d'échappement. C'est aussi le système retenu par Apple pour le Macintosh, dont le système d'exploitation permet d'utiliser des langues à grands nombres de caractères (chinois, coréen etc.).

Il faut toutefois remarquer qu'aucun de ces systèmes de codage n'est compatible avec le jeu de caractères français défini par le code ASCII « étendu ». L'utilisation du 8^{e} bit de l'ASCII normal permet en effet d'ajouter 94 caractères supplémentaires : voyelles accentuées, c cédille etc. Il y a alors confusion : l'octet dont le bit de poids fort est ainsi mis à 1 est considéré par un système japonais comme le premier octet d'un code Shift-JIS, ce qui aura pour effet de faire apparaître,

[4]Nous avons dû ajouter trois caractères et en modifier un pour ce premier numéro de *Cipango*.

[5]On se plaît à rêver à la définition d'un tronc commun de caractères entre la Chine, la Corée et le Japon : le même caractère aurait le même numéro de code à Pékin, Taipeh, Séoul ou Tokyo, les variantes régionales ou nationales, comme nos accents, pouvant être regroupées dans des jeux séparés… En 1983, un projet de norme internationale, ISO 10646, allait en ce sens et proposait un code sur deux octets. En raison de l'opposition du Japon et de la Corée à un sous ensemble commun de caractères *Han*, ce projet prévoit donc maintenant un code sur quatre octets. En 1987, quelques personnalités du monde informatique, réagirent et formèrent un groupe de travail en vue d'aboutir à la définition d'un code sur deux octets, plus simple et plus efficace, appelé *Unicode* qui reprend à son compte l'idée d'un ensemble unifié de caractères chinois. Mais les obstacles politiques demeurent les mêmes… *Cf*. Sheldon (Kenneth M.), *Byte*, juillet 1991.

dans le mot « système » par exemple, un *kanji* à la place des deux caractères « èm » : « syst塾e ». Ce caractère incongru est appelé *moji bake* 文字化け. Dans le cas du Macintosh, il suffit d'utiliser une police de caractères correspondant à la langue ou, plus exactement, au « *script* » utilisé si l'on ne veut pas voir une voyelle accentuée et le caractère qui la suit transformés en caractère chinois au milieu d'un mot français[6].

La saisie du texte

Les claviers

Il fallait aussi résoudre le problème de la saisie de plusieurs milliers de caractères. Plusieurs systèmes ont été mis au point que l'on peut actuellement répartir en 2 catégories : les claviers « complets » à accès direct, présentant environ 3 000 caractères classés par ordre phonétique, et les claviers à accès indirect.

Pour les premiers, il faut encore distinguer deux types : ceux où l'on sélectionne des caractères d'une seule main, par pression avec un stylet ou un crayon électronique sur le caractère choisi, et ceux qui nécessitent l'usage des deux mains, un doigt de la main droite appuyant sur une touche parmi plus de 200, comportant chacune 12 caractères, tandis que la main gauche choisit sur un clavier numérique de 12 touches celle correspondant à la position du caractère voulu parmi les 12 figurant sur la première touche[7].

Les claviers à accès indirect sont des claviers semblables à ceux des ordinateurs américains, les touches disposées selon l'ordre QWERTY permettant d'obtenir soit le caractère latin, soit un *katakana*, le choix étant conditionné par la position d'une touche analogue à celle du blocage en majuscule d'une machine à écrire (la disposition

[6]Cette revue, par exemple, est composé en caractères « Palatino » pour le français, écriture romane, et en caractères « Kyoto » pour le japonais.

[7]Système utilisé au siège du journal *Asahi* pour la saisie des articles.

des *katakana* varie suivant le type de claviers : clavier normalisé JIS, nouveau clavier normalisé JIS, ou ordre phonétique du syllabaire). Le texte japonais peut alors être obtenu de plusieurs façons. Le procédé le plus simple à mettre en œuvre est la saisie du code du caractère voulu. Ce code peut être exprimé soit par les coordonnées du caractère dans le tableau défini par la norme JIS, 2 chiffres pour le numéro de ligne puis 2 chiffres pour le numéro de colonne (mode *kuten*), soit par le code JIS lui-même ou sa variante Shift-JIS. Dans ces trois cas, il suffit de taper les 4 chiffres correspondants. Ce procédé implique de la part de l'opérateur une recherche dans le tableau des codes des caractères, ce qui n'est intéressant que lorsqu'on ne connaît pas la lecture du caractère cherché. On lui préfère donc le procédé de conversion *kana-kanji*.

Conversion *kana-kanji* かな漢字変換

La saisie s'effectue en transcription phonétique, soit directement en *kana*, soit en caractères latins automatiquement convertis en *kana*. La conversion en *kanji* sera ensuite effectuée à la demande, caractère par caractère, segment de phrase par segment de phrase ou même phrase par phrase selon le degré de sophistication du logiciel. Dans le premier cas, en raison du grand nombre des homophones, l'utilisateur devra intervenir souvent pour choisir parmi plusieurs caractères celui qui convient. La conversion d'un segment de phrase et d'une phrase entière sera plus aisée, certaines ambiguïtés étant levées par une analyse syntaxique automatique du contexte.

Sur le plan technique, cette conversion prend place au niveau du système d'exploitation, en amont des logiciels d'application. Elle est effectuée par ce que l'on appelle un *Front End Processor* (FEP) ou « processeur de saisie », logiciel chargé de gérer les entrées-sorties clavier-écran, qui intervient quelle que soit l'application utilisée. Le FEP fait appel à un dictionnaire contenant de 30 000 à plus de 50 000 « mots ». Il existe plusieurs de ces FEP, aussi bien sur les machines

MS-DOS[8] où les plus utilisés sont ATOK et VJE, que sur Macintosh, où le FEP fournit par Apple Japan est concurrencé par plusieurs autres logiciels.

L'affichage

Les caractères chinois qui sont particulièrement riches en informations demandent une certaine précision pour être représentés graphiquement. Lorsqu'IBM lança son premier modèle d'ordinateur personnel, celui-ci n'avait qu'une résolution d'affichage limitée, alors que NEC avait déjà pourvu ses machines (séries PC-8801 puis 9801) d'une résolution verticale double, afin de pouvoir afficher sur l'écran 25 lignes de 80 caractères codés sur un octet (en « demi-largeur », *hankaku*) ou de 40 caractères codés sur 2 octets (« pleine largeur », *zenkaku*).

Le dessin des caractères est conservé en général dans une mémoire morte (ROM) de grande capacité. Chaque caractère est représenté par une matrice de 16 x 16 points le plus souvent, et parfois de 24 x 24 pour certains terminaux ou pour les stations de travail.

L'impression

Elle peut se faire soit sur imprimante matricielle ou à jet d'encre, soit sur imprimante laser. On utilise des fontes de 24 x 24 points en général, mais certaines machines spécialisées pour le traitement de texte vont jusqu'à 32 x 32, ce qui donne une très bonne qualité d'impression. Les imprimantes laser utilisant le langage *Postcript* font encore mieux (stations de travail sous Unix, Macintosh...) sans parler des photo-composeuses utilisées dans l'imprimerie.

[8]Pour environ les trois-quarts, ce sont des ordinateurs NEC de la série PC-9801, incompatibles avec les machines de type IBM-PC. *Cf.* bibliographie, Mariani (M.) 19 88 .

Le classement des données

On ne peut utiliser un classement par prononciation en raison des lectures multiples des caractères. Le classement par ordre de clefs puis de traits utilisé par les dictionnaires de caractères ne peut convenir car l'ordre est indéfini lorsque le nombre de traits est identique. D'autre part, le nombre de caractères n'est pas figé une fois pour toutes comme les lettres de notre alphabet.

Pour trier une liste de mots japonais, il faut faire appel à une rubrique supplémentaire dans laquelle figure la transcription selon un des deux syllabaires *kana*. Ces syllabaires comprennent en effet un nombre fini d'éléments pour lesquels existe un ordre reconnu. Le tri se fera donc sur cette rubrique.

Conclusion

Cette écriture représente un défi que la technologie japonaise n'a cessé de relever avec succès : dès les années 60, l'optique japonaise s'est imposée sur le marché mondial ; en ce qui concerne la télévision, les tubes cathodiques japonais sont parmi les plus réputés pour leur finesse de reproduction. Le procédé de télévision haute définition de NHK est déjà une réalité. Il en va de même pour les imprimantes et, d'une façon beaucoup plus générale, pour tout ce qui, dans la vie quotidienne, sert soit à écrire, soit à reproduire l'écriture. Ces performances sont, à notre avis, directement liées à la complexité graphique de l'écriture japonaise.

Dominique Giroux

Bibliographie

JIS Handobukku Jôhôshori **JISハンドブック情報処理**. Tôkyô, Nihon kikaku kyôkai (Japanese Standards Association), 1983, 1 583 p.

NEC PC-9801F USER'S MANUAL. Tôkyô, Nihon denki kabushiki kaisha, Nihon denki hômu erekutoronikusu kabushiki kaisha, s.l., s.d.

NEWTON Bessatsu 1 Wâdopurosessa no subete **ニュウトン別冊1ワードプロセッサのすべて** [NEWTON, Hors série n°1 Tout sur le traitement de textes], Kyoikusha, Tôkyô, mai 1982, 300 p.

Script Manager Developer's Package 1.0 Release Note, Kanji Talk 1.1 Usage Note, AIS 1.1 Usage Note, Apple Computer Inc., Cupertino, 1987, 58 p.

Collins (Rip), « Time to replace ASCII ? », *in Byte*, janv. 1990, Mac-Graw Hill.

Griolet (Pascal) *La modernisation du Japon et la réforme de son écriture*, Publications Orientalistes de France, Paris, 1985, 124 p.

Lucas (Nadine), « Le traitement de texte en langue japonaise », texte dactylographié, s.l., s.d., 16 p.

Mariani (Michel), *Le logiciel au Japon (annexe Traitement informatique de la langue japonaise)*, Ministère des Affaires Etrangères, Direction de la coopération scientifique-technique et du développement, Paris, 1988, 35 p.

Sheldon (Kenneth M.), « ASCII Goes Global », *Byte*, juillet 1991, Mac-Graw Hill.

Thévenet (Michaël), « Mac au Japon », *Mac Informatique*, juil.-août 1989, n° 14, Exa-Informatique, p. 10 à 13.