SHARE EUROPE

Rapport technique

Texte et codes - L'antre du dragon

Février 1988

Les objets codifiés ne sont pas propres au domaine informatique. La vie de tous les jours est également parsemée de messages codés. Il est toutefois évident que l'on ne peut tirer aucun enseignement intéressant de cette expérience quotidienne, car toute cette information n'est utilisable que si son code est connu. On peut ainsi considérer le dilemme de la codification comme un second "problème informatique du siècle" (à côté de celui de la normalisation des bases de données).

Rapport technique

Un "rapport technique" de SHARE Europe est émis par un organisme membre de l'Association.

Il traite de sujets d'intérêt pertinent pour d'autres de ses membres. Ainsi, par exemple :

- Information de base pour le travail d'un STWG (groupe de travail technique spécial);
- Information liée à une présentation donnée lors d'une conférence (mais trop volumineuse pour être incluse dans les Actes).

Un tel "rapport technique" est imprimé aux frais de l'organisation membre. Son contenu doit être conforme aux statuts et au règlement de SHARE Europe. Le membre émetteur y est présenté selon les mêmes règles que si le document devait paraître dans les Actes.

Ce rapport "Text et code - L'antre du dragon" est traduit de l'allemand par le Ministére des Comminications du Québec.

Liability

The ideas and concepts set forth in this publication are solely those of the respective authors and not of SHARE Europe (SEAS). SHARE Europe does not endorse, guarantee or otherwise certify any such ideas or concepts in any appliaction or usage.

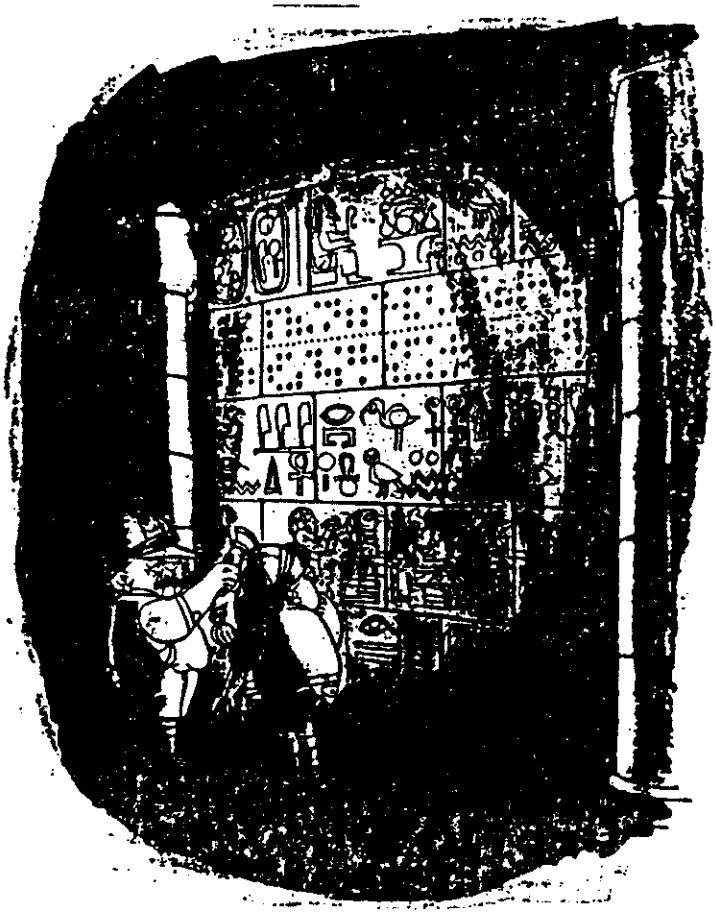
OBRZ

Manuel 410

Traitement de texte sur MVS

Version 1

410.10.25 – Texte et code



K. Daube

Oerlikon-Bührle
Rechenzentrum AG
Jungholzstr. 43

CH-8050 Zurich

Tél. 01/301 24 66

410.10 GÉNÉRALITÉS SUR LE TRAITEMENT DE TEXTE

410.10.25 Texte et code

GÉNÉRALITÉS SUR LE TRAITEMENT DE TEXTE

Texte et code

1. Introduction

1.1 Antécédents

En 1986, la Société a élaboré des directives qui fixaient des règles pour l'échange d'information (notamment les textes et les documents). Elles précisaient également que, dans l'environnement IBM, il fallait insérer la *page code 500*. Les considérations suivantes ont pour but de montrer que :

- les directives établissent un objectif sans que l'on sache combien de temps il faudra pour le réaliser ;
- qu'on ne peut pas dire d'avance non plus si l'objectif est proche ou non ;
- que la réalisation de cet objectif ne sera pas le fruit de nos seuls efforts car nous nous trouvons dans un environnement très dynamique ;
- que nous n'attendons pas l'*ultima ratio* et que nous ne sommes pas en mesure de produire de colossales conversions de données ;
- que nous devons donc nous accommoder de ces imperfections.

Il convient aujourd'hui de retenir que les directives, tout en reconnaissant la complexité des problèmes, ne les exposent pas.

[dessin humoristique]

Le *National Language Support* fait intervenir dans les applications les caractéristiques suivantes :

- les données en entrée et en sortie peuvent comprendre tous les signes qui sont indispensables à une langue donnée dans un pays donné. Par exemple, l'allemand en Allemagne et en Autriche requiert le signe ß, mais non en Suisse.

- Le dialogue avec l'utilisateur peut se faire dans la langue choisie par ce dernier. Cela comprend le nom des instructions, les abréviations, les textes servant d'aide, les masques d'écran, etc.

- L'application peut faire entrer en ligne de compte les usages nationaux tels que les signes de groupement des chiffres (CH : 17'300,-- USA : 17,300.--), la forme des données, les symboles monétaires, l'ordre de tri, etc.

Les considérations exposées ci-après porteront exclusivement sur une partie des problèmes liés au *National Language Support*, notamment le **codage des signes**.

1.2 Aperçu

- Types de codage
- Applications et code
- Chaînons manquants pour résoudre les problèmes
- *IBM CH National Language Support cookbook*
- *IBM National Language Support design guide*
- Marche à suivre
- *Code pages 500, 037, 850, ISO, 136 multinational* et *136 German/Austria*

2. Codages

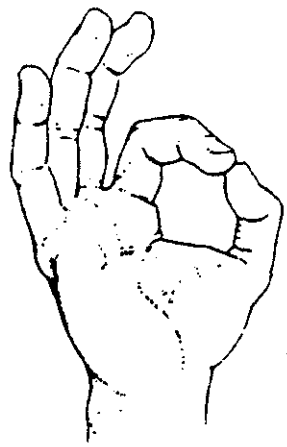
2.1 Les codes dans la vie de tous les jours

Dans la vie de tous les jours, nous sommes entourés de codes dont nous n'avons pas conscience :

- mimique et gestes
- pictogrammes
- règles de la langue
- caractères.

2.1.1 Gestes

Nous ne pouvons comprendre les mimiques et les gestes que si nous sommes connaissons la situation dont ils tirent leur origine. Le plus souvent, il est même indispensable de connaître le milieu culturel de la personne qui nous fait un geste [6] :



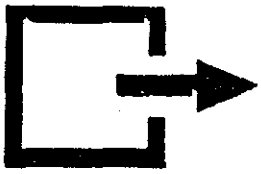
Ainsi, le geste reproduit ci-contre a des sens très différents :

- chouette, précis (le geste montre qu'on tient quelque chose de très filiforme – dans les pays de langue allemande ;
- argent – au Japon ;
- rien, zéro, sans importance – en France ;
- allusion obscène – en Sardaigne.

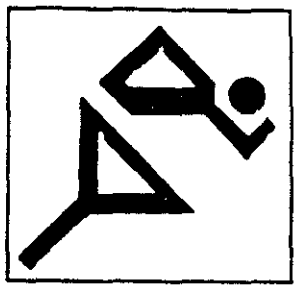
D'autres gestes ne peuvent être compris que par un groupes d'initiés. Si un sapeur-pompier se place à un carrefour, les gens qui le voient savent que ses gestes servent à réglementer le trafic. Ses camarades comprennent le geste mais au sens plus précis de « réduire la pression ».

2.1.2 Pictogrammes

Les pictogrammes sont certes des abstractions représentant des situations quotidiennes mais il n'empêche qu'on remarque ici que l'imagination de l'auteur ne correspond pas tout à fait à l'imagination du destinataire. L'imagination dépend de l'expérience acquise.



Le premier pictogramme représenté ci-contre peut toujours être compris comme « sortie ».



Le second pictogramme est construit de façon quelque peu curieuse. J'ai longtemps pensé que cette pancarte au bord d'une route qui traverse la forêt signifiait : voici des gens qui marchent avec des béquilles... Or, elle fait simplement allusion à des coureurs.

2.1.3 **Abréviations et règles de la langue**

Dans le traitement de l'information, nous avons souvent affaire à des abréviations : elle prêtent toujours à équivoque. Qui plus est, certains mots reçoivent une nouvelle signification avec le temps. Parfois, des faits sont voilés consciemment (cf. Newspeak de George Orwell).

AM en radiophonie, signifie : amplitude de modulation ; sur une montre toutefois : *Ante Meridiam* (avant-midi).

ACF *Access Control Facility* ou *Advanced Communication Facility* ?

incident signifie le plus souvent un accident dans une installation technique (généralement de grande taille)

ministère de la Défense est employé également par les États qui ont une politique très belliqueuse...

Dans ces domaines aussi, notre expérience, notre connaissance du milieu, etc. sont essentielles pour comprendre le « message ».

2.1.4 Caractères

Chacun reconnaît qu'il s'agit ici du codage de sons ou de significations. Celui qui croit que l'univocité est de mise n'a qu'à jeter un coup d'œil sur les caractères chinois : les trois zin = signifient homme [9] ou servent à représenter la prononciation des mots américains ou anglais « lite », « light » et « enough » ou encore « after ». Souvent, les sons et l'écrit ne peuvent être interprétés ou compris qu'en contexte. Comment un procédé automatique peut-il venir à bout de ces difficultés ?

2.2 Code et traitement de l'information

Les considérations relatives aux « codes dans la vie de tous les jours » montrent que la situation dans le traitement de l'information ne peut certes pas être plus simple, car

La connaissance de conventions est le passage obligé de la compréhension d'un code.

Et les machines ou les programmes sont incapables aujourd'hui encore de comprendre les contextes. Pour la définition du mot « code », on se reportera à la citation [7] exposée ci-après. Ce qu'il y a d'intéressant pour nous ici, c'est que le sens qui nous est familier n'apparaît qu'à la toute fin de l'article...

> **Code** 1) Recueil de lois. **Cinq Codes** fr. Ensemble des lois sous Napoléon Ier ; qui comprend le **C. civil (C. Napoléon)**, fr. Code civil français publié le 21 mars 1804 et qui fut également introduit en Belgique et au Luxembourg ; en Bade, le Code civil est resté en vigueur jusqu'en 1899 en tant que **Droit national badois ; Code de commerce**, fr. code de 1807 ; **C. de procédure civile**, fr. code de 1806 ; **C. d'instruction criminelle**, fr., code de 1808. Le C. de commerce a été pour l'essentiel abrogé par de nombreuses lois complémentaires et remplacé par des lois individuelles. Le C. d'instruction criminelle a été remplacé par le **C. de procédure pénale** de 1957/58.
>
> **Uniform Commercial C.**, code de commerce uniformisé des États-Unis d'Amérique, moins la Louisiane.
>
> **C. law system**, désignation anglaise du droit civil continental européen codifié. Droit civil, par opposition aux lois anglo-américaines qui en grande partie ne sont pas écrites.– cf. Common Law

2) Génétique moléculaire : cf. code génétique
3) Règles de déchiffrage, de conversion, clés pour déchiffrer ; cf. codage.

Pour la représentation interne des signes dans le traitement de l'information, on utilise divers codes – cf. code binaire (bit, - octet), par exemple EBCDIC (BCD), ASCII, Aiken, 3-Exzeß-C. Le code BCD est un **code à base fixe**. Des **codes de contrôle** sont souvent construits par l'addition d'un bit de parité, où une erreur de transmission de 1 bit peut être remarquée, puisque le chiffre du >1< dans chaque signe est pair ou impair. Dans le cas des **codes réfléchis (cycliques)** seul un bit est modifié lors du passage d'un signe au signe voisin (cf. convertisseur analogique-numérique). En programmation, des **codes d'instructions** servent à donner des instructions à l'ordinateur (coder).

Dans le traitement de l'information, les codages sont à l'ordre du jour et très formalisés. Nous entendons généralement par code la reproduction d'un signe sur les 8 bits d'un octet. Mais des valeurs numériques sont également codées. Un nombre entier sera construit sur 16 ou 32 bits.

Dans le présent texte, on utilise aussi des codes spéciaux, pour rendre visible ce qui n'est pas représentable :

<...> représente un caractère de commande. Une application l'exécute, il ne sera pas représenté dans la sortie. Par exemple <esc> constitue le caractère de commande de « escape ».

3. Description d'un état final

Par état final j'entends ici le produit d'une action. Par exemple un signe écrit, qui est la conséquence visible de traits couchés sur du papier. Ou un signe affiché à l'écran, qui a été mis en relief ou agrandi. Un menu constitue un exemple d'« état final » que l'on trouve dans la vie de tous les jours.

Un état final peut être décrit de deux façons au moins :

- directement par un nom ;
- indirectement par un chemin.

Dans les deux cas, des codes entrent en jeu. Je dois savoir sur quel territoire j'avance ! Les désignations, comme les descriptions de chemins, sont nécessairement assorties de conditions et exigent connaissances de base et imagination.

De quoi s'agit-il dans ce qui suit ?

- Châteauneuf-du-Pape
- Châteaubriand
- Château Chillon

L'amateur de bons vins croira qu'il s'agit dans tous les cas de marques de vins. L'amateur de bonne chère dira peut-être : Au Château Chillon, dernièrement, j'ai mangé un Châteaubriand arrosé d'un Châteauneuf-du-Pape.

La description du chemin qui mène au Châteaubriand (la recette) utilise des codes tels que « faire mariner », « saignant » ou « à point ».

Châteaubriand

FAIRE CUIRE
à la minute

600-700 g de filet de bœuf, par exemple la pointe de filet
marinade de viande de bœuf

Si possible, laisser mariner la viande pendant la nuit.
Sécher avec du papier avant la cuisson.

l cuillerée à thé de sel
l cuillerée à soupe d'huile Sais

Bien frotter la viande avec du sel, badigeonner d'huile, faire chauffer la poêle, faire cuire la viande sur tous ses côtés, diminuer la chaleur tout en laissant la viande cuire. De temps en temps, retourner et enduire d'un peu de marinade qui reste.

Temps de cuisson pour un Châteaubriand d'environ 5 cm d'épaisseur :

– bleu : environ 7-8 minutes
– saignant : environ 12-14 minutes
– à point : environ 17-18 minutes
– bien cuit : environ 22-25 minutes.

Conseil
A la fin du temps de cuisson, redresser la viande pour en cuire légèrement les bouts.

3.1 Désignation d'un état final

- L'état final reçoit un nom (par exemple « petit signe italique PI » ou Châteaubriand)

IBM, entre autres, a choisi cette méthode, car, pour la plupart des applications, il est très efficace de pouvoir avoir recours directement à un signe, pour le traitement. Cela ne convient toutefois que lorsque les conditions suivantes sont réunies :

- Le codage du fichier est bien établi.
- L'application connaît le codage. (Qu'arrive-t-il lorsque l'application peut certes faire le traitement de texte, mais assimile tout simplement π à p ?)

Il ne faut pas pour autant **supposer** que toutes les données sont placées dans un même code servant à une application. Le code EBCDIC que plusieurs disent « standard » n'est que la partie invariante de plus de 800 différentes pages code !

On pensait également que cette méthode permettait d'accélérer le traitement de texte. Ainsi peut-on (à ce qu'on dit) déterminer la longueur d'une chaîne de caractères en comptant les octets. Cela est peut-être vrai pour le traitement en mémoire, mais non forcément aussi pour l'entrée ou la représentation à l'écran et pour l'imprimante (fonte proportionnelle, changement de fonte, modifications de la taille, etc.).

On peut peut-être mentionner ici que l'écriture proportionnelle est « normale » (écriture manuelle, typographie), que les jeux de caractères à espacement fixe sont issus de contraintes propres aux moyens mécaniques et sont donc « anormaux ».

Dans le contexte qui nous intéresse, l'état final est appelé en général **page code.** Des codes sont attribués à une série de 192 signes différents [1] . Ces pages codes sont enregistrées par IBM et portent donc un numéro d'identification (CPGID).

---

[1] Un ensemble de code comporte 64 caractères de commande et 190 signes graphiques. À cela s'ajoutent le caractère espace et un caractère dont le code comprend tous les bits. Ce dernier caractère n'est représenté d'aucune façon.

Un ensemble partiel de signes rassemblés sur une page code est appelé jeu de caractères graphiques codé (identifié par un numéro - CGCSGID). Un ensemble partiel peut s'étendre pour constituer un ensemble complet :

- Le jeu de caractères de la page code 500 est identique au jeu de caractères de la page code 037 et porte le numéro CGCSGID 00697. Ce jeu est depuis le milieu de 1987 en tous points identique au jeu de caractères ISO 8859/1 (alphabet latin numéro 1).

- La page code 850 (page code multilingue de PS/2) comprend tous les signes du CGCSGID 00697 comme sous-ensemble. La page code 850 complète porte le numéro CGCSGID 00980.

La multitude de jeux de caractères et de pages code tient au fait qu'à l'origine ils s'inscrivaient dans le cadre d'un code à 7 bits (128 codes, dont 32 pour les caractères de commande). Il y eu aussi pendant de longues années des limitations dues au matériel, qui ont ancré cet état de choses.

Dans la plupart des pays, on a donc cédé à la tentation de remplacer les signes « inutilisés » d'un code par d'autres :

USA	USA	¢	¦	!	$	`	#	@	{	}	\
France	Frankreich	°	!	$	$	µ	£	à	é	è	ç
Brésil	Brasilien	È	!	$	Ç	ã	Õ	Ã	õ	é	\

3.2 Description du chemin qui mène à l'état final

- L'état de départ doit être connu (par exemple ASCII ou table de cuisine dégarnie)

- La description du chemin doit prendre la forme d'un code (séquences de commande ou menu).

La meilleure description du chemin ne mène pas au but lorsque le mauvais point de départ est choisi.

Dans le monde de l'ANSI, il est supposé que chaque fichier commence par ASCII. Les différents jeux de caractères, les mises en relief, les changements de taille, etc. sont représentés par des codes (séquences de commandes). Ils ont tous été normalisés. Le contenu des jeux de caractères et leur identification sont bien établis. Par exemple :

<esc>) E	Désormais le (premier) jeu de caractères de substitution est NATS (NATS = Newspaper text transmission in Denmark and Norway). Le code hexadécimal 5C y représente par exemple un Ø. En ASCII on trouve un \ à cette position.
<so>	Activation du jeu de caractères de substitution (NATS devient donc actif).
<si>	Réactivation du premier jeu de caractères.
<esc> [1 ; 3 m	Les signes graphiques suivants sont représentés en caractères gras et en italiques.

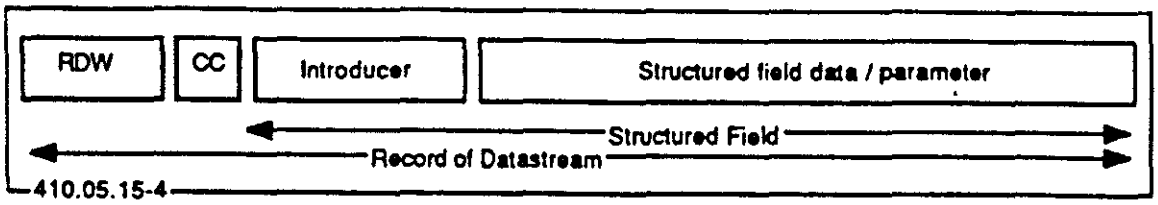
Pratiquement toutes les applications des mini-ordinateurs et des micro-ordinateurs utilisent cette méthode. Au centre OBRZ, VARIO s'en sert.

Si des fichiers utilisent cette méthode, les applications doivent pouvoir accepter toutes les séquences normalisées (en tout état de cause signaler une erreur d'entrée). Elles ne sont toutefois pas tenues de les comprendre toutes et de pouvoir les traiter.

Une application ne peut pas ici « plonger » directement dans un fichier et, par exemple, traiter l'octet 337. Ses attributs (comparables aux coordonnées sur le terrain, altitude, etc.) sont inconnus, puisque le chemin qui mène à cet état est inconnu. Ces données doivent donc toujours être traitées en tant que flot.

Pendant le traitement de ce flot d'information, il convient de rétablir les états pertinents. Un programme d'édition doit par exemple être informé de la largeur et de la hauteur du signe. Il doit également savoir, en ce qui concerne le jeu de caractères, si des signes servant à tracer des dessins (par exemple ⊢┼⊣) peuvent être traitées par des commandes d'édition spéciales (par exemple « dessiner » avec le curseur).

IBM a repris cette méthode dernièrement avec les champs structurés (**structured fields**) qui sont utilisés par exemple en SNA, DCA, IPDS, etc.



Ces constructions peuvent s'enchevêtrer. Contrairement aux séquences de commandes ANSI, toutes les **configurations** binaires peuvent apparaître dans les données, puisqu'un *symbole d'arrêt* n'est pas recherché pendant le traitement. Cela est très important pour les applications à images tramées.

Il est intéressant de noter à cet égard que les caractères peuvent très facilement être reconnus automatiquement, lorsque leur composition est susceptible d'être analysée, par exemple lorsque des caractères chinois sont saisis pendant qu'ils sont couchés sur du papier (étant donné que l'ordre des traits est très rigoureux). Il est de toute évidence plus facile de saisir une configuration pendant sa composition que de comprendre son « état final » déjà arrêté.

4. Applications

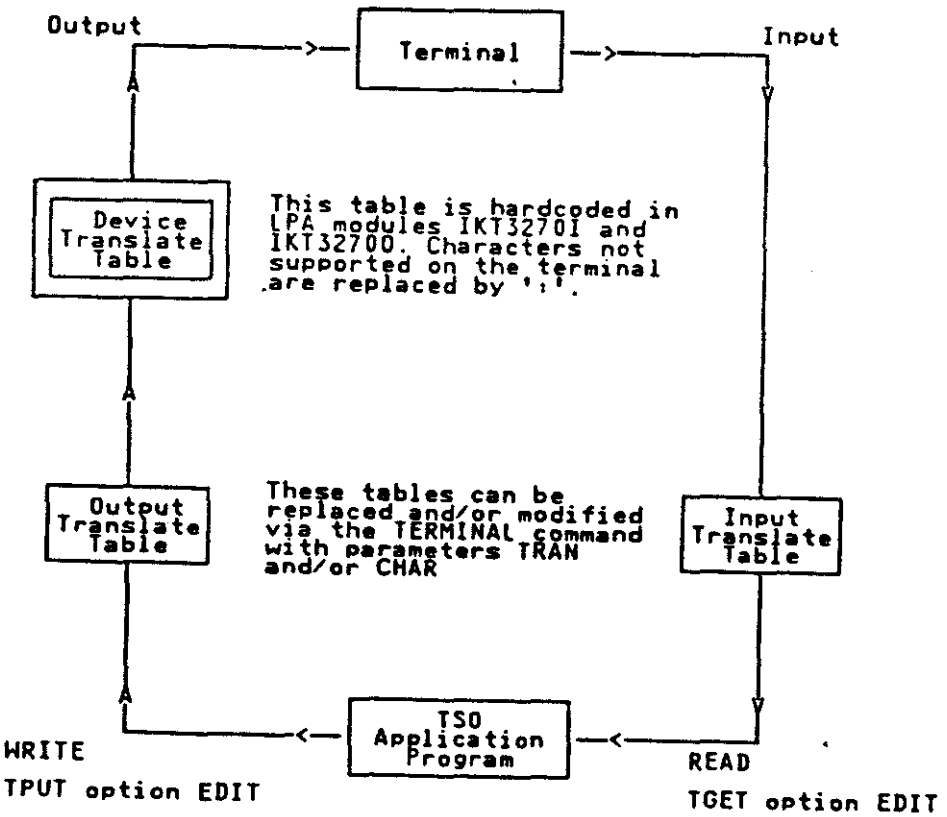
- Entrée hors système
- Enregistrement dans le système
- Sortie hors système

Toute application fait généralement intervenir ces trois tâches. Étant donné que les bits ont une « valeur neutre », leur signification doit être connue. La série de 8 bits B'110000011000010' peut signifier le chiffre 171, les lettres majuscules AB ou une configuration de points ou encore autre chose.

Dans le cadre des applications pour les banques de données, on finit par se rendre compte que les attributs des données ne doivent pas être établis dans le programme mais dans une description des données (dictionnaire). Cette règle doit être utilisée dans les profondeurs mêmes du codage.

Les données doivent être gravées dans le code utilisé, et une application doit pourvoir les reconnaître.

Avec l'apparition des systèmes ouverts ou faisant partie d'un réseau, les sources de données sont multiples aussi. Elles ne se conforment nullement à la partie invariante de l'EBCDIC. Les gestionnaires des données peuvent très bien déterminer quels codes seront admis sur leurs supports de mémoire. Il n'empêche qu'on ne peut pas prescrire quel code par exemple arrivera par communicateur de Sydney. Je ne saurais présumer tout simplement non plus que tout ce qui vient de Sydney est en anglais et que l'« EBCDIC standard » (le jeu partiel de caractères US de la page code 037) sera utilisé. Les données peuvent très bien émaner des îles Fidji (où le français est la langue officielle), Sydney ne servant que de plaque tournante.



Le diagramme ci-dessus montre bien les transformations de code qui se produisent lors de l'entrée et de la sortie à un terminal dans le cadre de l'application TSO. Il convient également de remarquer que l'éventuelle adaptation des tableaux correspondants par les instructions TERMINAL et CHAR données par TSO s'égarent lorsque le système de dialogue ISPF est appelé !

Pour l'application « courrier électronique » qui n'a pas encore été intégrée complètement dans l'OBK, ces rapports seront montrés dans le détail.

4.1 Entrée de données

- Le clavier détermine le jeu de caractères.
- Il représente la section d'une page code.
- En Suisse, il y a eu d'abord 107 caractères ; il y en a maintenant 131.
- L'écran peut souvent afficher plus de caractères (de 107 à 116).

4.1.1 Jeu de caractères

Par jeu de caractères, on entend une série de caractères visualisables (à l'écran, sur imprimante). Le nombre de ces caractères est en général inférieur au nombre total des caractères qui sont affectés avec un code donné.

En EBDCIC, une page code peut contenir 256 caractères différents (1 octet permet 256 différentes combinaisons de bits). Sur ce nombre, 64 constituent des caractères de commande et sont réservés, et un (qui rassemble tous les bits) n'est pas affecté à un caractère graphique. L'espace (caractère blanc) est également invisible. Il reste donc 190 caractères visibles (graphiques) dans une page code EBCDIC.

Différents jeux de caractères peuvent se servir du même codage. Le cas échéant, ces jeux de caractères s'inspirent de la même page code.

Étant donné que le jeu de caractères susceptible de traitement par un appareil ne doit pas nécessairement contenir 190 (192) caractères, ces jeux sont numérotés (CGCSGID) par

IBM. C'est en prévision du jour où les appareils pourront s'identifier. On pourra alors vérifier si l'appareil peut ou non représenter les caractères en cause...

4.1.2 Clavier

Généralement, le clavier qui accompagne un appareil (écran) ne comprend pas suffisamment de touches pour produire 190 signes graphiques différents. Un clavier donné représente un jeu de caractères donné.

Le clavier peut toutefois être assorti de touches mortes à l'aide desquelles on peut combiner des signes. Si par exemple les accents aigu, circonflexe et grave sont définis comme touches mortes, on peut dès lors produire toutes les voyelles accentuées. Ainsi, on peut avec 3 accents + 5 voyelles = 8 touches produire 3 x 5 = 15 caractères ou codes.

Dans la famille des terminaux 3270, les claviers et les contrôleurs d'écran produisent des codes qui sont envoyés à l'application. La famille des terminaux 3270 est limitée à moins de 192 signes, ce qui est de toute évidence une limitation d'ordre matérielle due au contrôleur plus ancien. Il faut préciser que, en réalité, l'application ne peut s'adresser directement au terminal. Il existe entre les deux un moyen d'expression limité au niveau du code : le code d'affichage.

En général, la définition du clavier (code produit) peut être différente sur chaque appareil qui « dépend » d'un contrôleur 327x. L'attribution des codes et des adresses de périphériques est définie de manière permanente dans les tableaux VTAM. Il est donc impossible d'enficher les claviers selon les besoins.

4.1.3 Entrée de données

Le code qui est reçu par l'application n'est pas identifié quant à sa nature. C'est pourquoi, dans le cadre d'applications à utilisateurs multiples comme CICS on ne peut raccorder que des claviers identiques. Ou alors l'application peut, en raison d'un indicatif d'utilisateur, déterminer le genre de code qui en proviendra.

En ISPF par exemple, il faut inscrire dans un tableau le genre de terminal qui est utilisé (menu 0.1). Certaines transformations peuvent alors être faites automatiquement. Celles-ci n'ont trait cependant qu'à l'entrée de données au moyen du clavier et à la sortie sur écran. Ce qui est mis en mémoire n'est identifié d'aucune façon, comme auparavant.

Si toutefois l'utilisateur change de terminal, il n'oublie que trop facilement de changer cette définition.

De nombreuses applications ne sont pas en mesure de soumettre les entrées à des transformations de code. Citons à titre d'exemple :

AS	! est utilisé comme séparateur de commandes.
Pascal	[et] peuvent heureusement être représentés plus traditionnellement par(. et .). Pour { et } on peut également utiliser en remplacement les représentations suivantes : (* et *).
PL/1	¬ et | sont des opérateurs.
QMF	¬ est utilisé dans les opérations.
REXX	| et ¬ sont des opérateurs.

Ces caractères ont une code différent selon le clavier utilisé et donc la page code qui intervient.

Dans le cadre d'une application, on peut procéder dès aujourd'hui à l'identification du code, comme c'est le cas avec l'application *SUSI* :

- Le clavier est déterminé via ISPF. Cela est établi par le code d'entrée.
- Si des données sont mises en forme, elles peuvent être converties de la page code 500 à la page code 037 ou inversement (macro-instruction de mise en forme ADAPT).
- Les données sont identifiées par une instruction *SUSI* appelée .codepage xxx. Cette instruction est, au besoin, placée au début des données (macro-instruction de mise en forme ADAPT).

4.1.4 Systèmes /36 et /38

À cette époque de la génération par système, on peut déterminer quel codage sera utilisé :

- jeu de caractères national ;
- jeu de caractères multinational.

Par jeu de caractères cependant, on entend aussi la « page code ». Les terminaux peuvent passer d'un code à l'autre. N'empêche que les données ne sont identifiées d'aucune façon ici aussi. Les données ne comportent aucun attribut « page code ».

La spécification qui est faite globalement pour le système d'exploitation sur le code utilisé peut en outre être définie autrement sur le système /38 pour chaque terminal ou imprimante. Il est alors procédé, dans le système, aux transformations de code correspondantes pendant le trajet entre mémoire et périphérie. Les données ne sont rangées que dans le codage établi pour le système.

4.2 Enregistrement de données

- Une page code ne suffit pas (192 signes différents).
- L'identification des données est indispensable.
- Selon l'application, l'identification peut aller jusqu'au niveau *champ*.

Celui qui, par exemple, rédige un rapport technique, regrettera l'absence de maints signes dans les pages code 037 ou 500 ou même 850 qui sont offertes. Dans le cadre d'applications, on peut avoir recours à des métasymboles, comme c'est le cas en *SUSI*. Le signe ∫ par exemple ne se trouve dans aucun des jeux de caractères offerts par IBM. En *SUSI*, on écrit donc 23'-23 par exemple (le signe en question est défini dans le troisième jeu de caractères « de référence » de *SUSI*).

Il n'est pas souhaitable que chaque application doive insérer ses propres mécanismes. La commutation entre des codes ou des jeux de caractères différents n'est définie que dans le cadre du DCA. Les applications qui admettent ce format (DW/x) sont encore peu nombreuses toutefois. En outre, dans ces applications, seuls les caractères qui figurent dans les jeux de caractères du PC sont définis (jeu de caractères 980).

Étant donné que, pour le temps d'entrée comme le temps de sortie, différents codages interviennent ou sont nécessaires, le code des fichiers doit pouvoir être reconnu. Les applications ne peuvent s'en occuper elles-mêmes qu'au sein du fichier (cf. *SUSI*). Des mécanismes qui entrent en jeu au niveau du système pourraient le faire dans les attributs de fichier (comme les attributs DCB).

4.2.1 Situation actuelle

Étant donné que les données (sur les systèmes /3x aussi) ne comportent aucun attribut « page code », partant, ne peuvent pas être identifiées, leur codage ne peut être modifié que par des « actions en catimini ». Il faut songer cependant qu'un fichier se compose rarement de texte seulement. De nombreux champs doivent plutôt rester affectés à des valeurs binaires (numériques) dans leur configuration.

Les programmes utilitaires voués à la transformation ne peuvent donc être employés que si la composition des fichiers est connue et fixe. Le texte « à code fixe » dans de nombreuses « applications standard » se soustrait à la transformation, lorsque aucun code source n'est présent.

4.3 Sortie des données

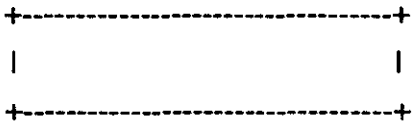
- L'appareil peut-il exploiter le code ?
- Le code est-il représenté correctement ?

Il n'y a guère d'appareil IBM aujourd'hui qui connaisse la fonction interrogation (*query*). Chez DEC, les terminaux VT réagissent depuis plus de 10 ans de manière analogue. C'est pourquoi on a pu y écrire des programmes d'édition qui sur VT-100 ne représentent que 128 signes différents et qui sur VT-220 toutefois manipulent des jeux de caractères chargeables. Et chaque signe peut afficher les attributs (par exemple clignotant ou rouge) et les tailles (par exemple deux fois la hauteur) les plus différents.

Une application ne peut donc pas déterminer si le code en présence peut ou non être exploité sur l'appareil. À la faveur d'une action de l'utilisateur, on peut en tout état de cause activer un tableau servant à la filtration. Mais on peut dire ici la même chose que pour ISPF : l'information risque d'être fausse (l'utilisateur ne sachant pas par exemple sur quel genre d'appareil il travaille).

Il est donc très important de connaître les capacités de l'appareil de sortie, lorsque des caractères non admis peuvent être représentés par des substituts. Il est préférable d'écrire ae plutôt que rien du tout, lorsqu'un ä n'est pas disponible. Ou de représenter

par



sur des machines non intelligentes, plutôt que de laisser un vide. Il me semble essentiel de perfectionner les possibilités de sortie d'abord, avant de créer de nouvelles sources de données.

Pour la sortie des données, il faudrait également que les codes non définis (c'est-à-dire les codes pour lesquels aucun caractère n'est défini sur un appareil donné) soient en principe susceptibles d'être reconnus. Pour de tels codes, il ne faudrait donc pas produire un espace (caractère blanc). Par exemple, dans le cas des lignes suivantes, l'utilisateur s'aperçoit tout de suite que des signes non définis ont été utilisés :

```
IF (A ■ B) CALL (xyz, abc, 'why ■', on)
```

On ne sait absolument pas en revanche ce que les lignes suivantes veulent dire :

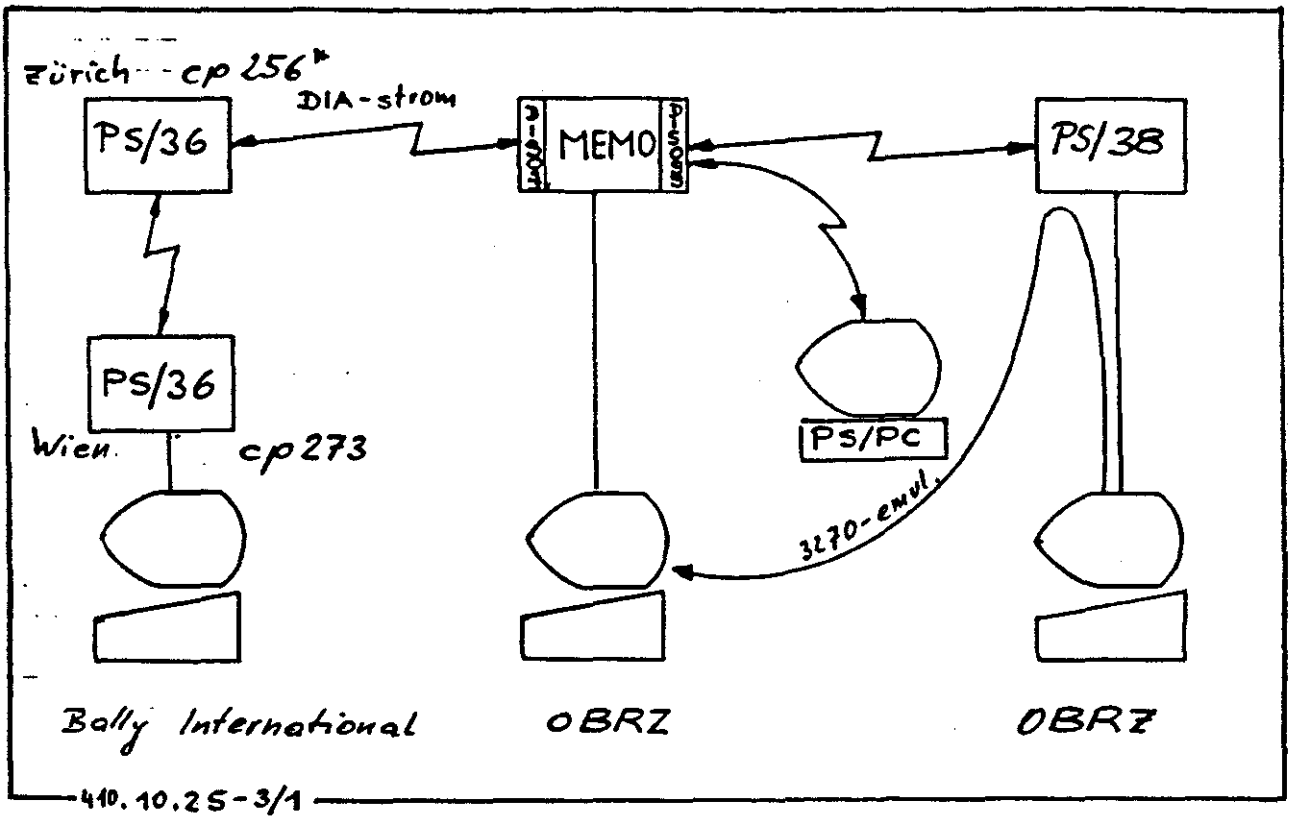
```
IF (A  B) CALL (xyz, abc, 'why  ', on)
```

Ou (pis) ce qui se passe avec les options par défaut d'IBM :

IF (A - B) CALL (xyz, abc, 'why - ', on)

Ainsi, les imprimantes au laser centrales au Centre OBRZ ne sont pas encore en mesure de représenter le jeu complet de caractères « Swiss ». Pour des raisons de performance ou d'autres raisons, il a été nécessaire de limiter à moins de 128 le nombre des caractères. Ce qui explique par exemple que certaines caractères accentués (á, ç, í, ñ, ó, ú, ȳ, É, Ñ) sont absents. Le jeu de caractères VSM comprend 131 signes.

4.4 L'application « Courrier électronique » avec MEMO



Le codage dans ce réseau est utilisé dans les conditions suivantes :

- Bally Vienne utilise le jeu de caractères autrichien/allemand (page code 273) sur S/36.
- Bally International à Zurich utilise le jeu de caractères multinational sur S/36. Celui-ci est à peu près identique (pour nous) à la page code 256.
- DISOSS utilise le code par défaut de la page code 256. À l'heure actuelle, tous les tableaux de conversion connus (environ 40) sont installés.
- MEMO travaille de façon complètement transparente, c'est-à-dire qu'aucun code n'est modifié d'aucune façon.
- Au centre OBRZ on emploie le « jeu de caractères multinational » sur S/38. Pour la transparence du terminal (émulation 3270), on se sert donc de « faux » codes. Le bureau /38 n'utilise pas le jeu de caractères multinational.

Les problèmes ne peuvent pas être exposés en détail ici (encore). Trop de choses n'ont pu être vérifiées. Voici toutefois ce qui est apparaît à la surface : Bally International envoie un message contenant le texte « Grüße an André » [Mes salutations à André] à divers destinataires. Ces derniers reçoivent l'image suivante sur leur écran :

Grüße an André	sur toutes les stations branchées sur /36 chez Bally International à Zurich
Grüße an André	sur toutes les stations branchées sur /36 à Vienne
Gr}~e an Andr?	sur tous les écrans branchés sur OBRZ/MVS. L'endroit occupé par ? est représenté par é sur les écrans ITT au centre OBRZ, par un point vraisemblablement sur les écrans IBM, et d'une autre façon encore sur d'autres écrans. Cela dépend de la configuration du contrôleur d'écran.
Gr}~e an Andr?	sur tous les écrans S/38 du centre OBRZ qui se comportent comme des terminaux en MVS.

Il convient de rappeler ici que les problèmes liés semble-t-il à MEMO ne sont pas attribuables à ce produit. Même lorsque des terminaux différents (avec des claviers différents et donc des codes différents) sont raccordés à DISOSS sur PS/370, le même problème survient.

5. Chaînons manquants

Les sections précédentes montrent clairement qu'une solution globale doit faire intervenir au moins deux éléments :

- identification du code utilisé ;
- possibilité d'interrogation des caractéristiques des périphériques.

Jusqu'à ce que la société IBM puisse offrir des solutions « propres », les applications ne fourniront que des « béquilles ». Ces béquilles doivent toutefois tenir compte des éventuels développements futurs. Il convient donc de suivre ces développements à la lumière de publications et de la mise en œuvre :

- de la stratégie SAA d'IBM ;
- des services X.400 du secteur public ;
- etc.

5.1 Identification du code

Le codage utilisé peut être « gravé » dans les fichiers selon les méthodes suivantes :

- les *structured fields* (champs structurés) qui constituent la méthode de choix (par exemple DCA, IPDS) ;
- les travaux relatifs à l'application (par exemple *SUSI*).

Il n'existe de solution « propre » que pour le domaine DCA, donc pour les applications DW/x. L'offre de jeux de caractères est toutefois plus maigre encore, et le nombre des appareils de sortie admis est minime. Des liens avec le « traitement de l'information classique » sont notoirement absents.

Pour ce qui concerne les sorties, IPDS fait usage d'identifications. J'ai le ferme espoir que dans le cadre de SAA ces principes trouveront faveur dans l'ensemble des logiciels IBM. C'est là, bien entendu, un travail de très longue haleine. Le solutions doivent toujours tenir compte de la situation existante – en particulier lorsqu'elles émanent d'une autorité « centrale ».

Pour *SUSI*, les fichiers sont identifiés en leur sein. Et avec l'application ISPF/PDF, l'ensemble fonctionne fort bien. Cette méthode repose toutefois sur l'identification des caractéristiques de l'appareil servant à l'entrée des données.

On pourrait adopter pour tactique générale d'interdire les signes qui ne s'inscrivent pas dans ce qu'il est convenu d'appeler le *syntactic set* (jeu syntaxique). Ce jeu de caractères comprend :

Type de caractère	caractère
Alphabet	ABCDEFGHIJKLMNOPQRSTUVWXYZ
	abcdefghijklmnopqrstuvwxyz
Chiffres	1 2 3 4 5 6 7 8 9 0
Caractères spéciaux	. , : ; ? ( ) ' " - _ & + % * = < >

On ne saurait conseiller cette méthode en général. Dans certains cas (par exemple courrier électronique universel), nous devrons certainement nous en accommoder.

Pour mettre en relief les problèmes posés par l'*identification de code*, on a posé les conditions suivantes du moins dans le cadre de la SEAS [2], (conditions qui ont toutes été élaborées par le centre OBRZ mais qui ont trouvé un grand écho) :

- identification des fichiers avec la page code en DOS ou OS/2 (inscription au répertoire)
- identification des fichiers avec la page code en MVS (par exemple attribut DCB)
- le compilateur VS Fortran doit pouvoir exploiter tous les signes d'une page code dans le texte source
- le compilateur VS Pascal doit pouvoir exploiter tous les signes d'une page code dans le texte source.

Chez IBM, c'est surtout le *Toronto National Language Technical Centre* qui a la tâche considérable de mettre les 25 000 concepteurs de logiciels au fait de ces problèmes [1, 2].

---

2 SEAS = SHARE European Association, association d'utilisateurs de systèmes MVS et VM.

5.2 Interrogation des caractéristiques des périphériques

Étant donné qu'une véritable interrogation auprès des périphériques est impossible à l'heure actuelle, les applications existantes se servent de diverses « béquilles » :

- ISPF peut être « réglé » par l'utilisateur.
- GDDM accepte une interrogation rudimentaire.
- *SUSI* assume les tâches d'ISPF.

Il est impossible à l'heure actuelle de savoir d'un appareil d'entrée de données de la famille 3270 quel code il utilise. Cette indication est certes présente dans le contrôleur, mais celui-ci ne peut pas être interrogé.

C'est pourquoi cette information est inscrite dans des tableaux propres aux applications ou en tout état de cause en VTAM, mais elle ne correspond pas nécessairement à la réalité. Car la situation des terminaux est devenue tout à fait dynamique.

En ce qui a trait à la sortie aussi (sur imprimante ou à l'écran), la situation est la même. Dans le cas de définitions complexes qui se prêtent fort mal à la maintenance, une application doit être informée des possibilités de mise en forme pour la sortie qui lui sont offertes.

Pour les imprimantes, IBM a adopté IPDS dans le cadre du SAA. SAA ne peut bâtir sur des appareils non intelligents comme par exemple les imprimantes de lignes. Pour les écrans, l'évolution du contrôleur, et surtout le passage aux postes de travail (les PC) en tant que terminaux, laisse entrevoir une tendance à une intelligence accrue. C'est par là justement qu'une multitude de possibilités s'offrent. Tant que la « puissance » sera virtuelle, il faudra partir du dénominateur commun le plus petit !

6. Recommandations de la société IBM

6.1 IBM Cookbook CH

If national keyboards are to be introduced on an existing network, careful analysis and planning is required in order to avoid severe end user and data integrity problems.

[Si des claviers nationaux sont branchés sur un réseau existant, il faut bien analyser la situation et s'organiser pour éviter de graves problèmes pour les utilisateurs finals et l'intégrité des données.]

Les ouvrages cités en [3] et [4] recommandent d'être prudent. En [3], on explique comment, dans le cas d'une installation pour diverses applications, passer au « jeu de caractères suisse » 00908 (131 caractères) de la page code 500 :

DFSORT	Proposition d'une méthode de tri pour la page code 500.
GDDM	avec les applications ICU et QMF ainsi que PS/370 et APL2 : modifications aux tableaux de conversion et au jeu de symboles vecteurs.
Adaptation à ISPF et PDF de divers tableaux ou introduction de tableaux supplémentaires	Ce travail a d'ores et déjà été entrepris par DTA (en toute ignorance des documents IBM).
JES328X	L'installation d'impression JES3/327x n'est pas utilisée au centre OBRZ.
PL/1	L'utilisateur est invité à utiliser d'autres caractères.
QMF	L'utilisateur est invité à utiliser d'autres caractères. Il faut réadapter ISPF et GDDM au jeu de caractères suisse.
SDSF	*Spool Display and Search Facility* : que je sache, il n'est pas utilisé au centre OBRZ.
TSO/E	Il faut réadapter le *session manager* de TSO/VTAM et TSO.

Les considérations exposées dans le *FSC cookbook* montrent clairement que, hormis ISPF/PDF aucun produit n'est en mesure de manipuler plus d'un codage (page code) ! Pour le tri, on peut être assisté de l'extérieur par une procédure. Le passage au jeu de caractères Swiss ou à la page code 500 doit donc être envisagé et planifié avec grand soin.

Aujourd'hui comme hier, il me semble qu'il faut passer par l'application, que la seule possibilité consiste à inscrire en divers endroits dans des bibliothèques correspondantes les modules de programme qui dépendent d'une page code. Mais le problème de savoir comment les programmes ou les procédures seront informés du code qu'ils doivent traiter – et cela, dans la mesure du possible, sans l'intervention de l'utilisateur – reste entier !

6.2 National Language Support Design Guide

Ce document, qui était « pour diffusion interne chez IBM seulement » jusqu'au milieu de 1987, est maintenant en circulation libre [1, 2]. Il renferme des directives à l'intention de l'équipe de développement IBM concernant la conception de produits qui se prêtent à diverses langues et à divers pays. Les exigences sont énoncées sous forme de règles et visent, entre autres, les éléments suivants :

- *Character sets and code pages* [Jeux de caractères et pages code]
- *General design considerations* [Considérations générales relatives à la conception]
- *Electronic character generation for displays and printers* [Production électronique de caractères pour les consoles de visualisation et les imprimantes]
- *Keyboard and keystroke processing* [Traitement du clavier et de la frappe]
- *Machine readable information* [Information exploitable par une machine]
- *Panels and messages* [Tableaux et messages]
- *National usage considerations* [Considérations sur les usages nationaux]
- *Terminology - grammatical and style sensitivity* [Terminologie – sensibilité grammaticale et stylistique]

Ces travaux sous la direction de Denis Garneau rattaché au Toronto Lab de la société IBM ont été considérablement influencés par le *SEAS White Paper* du *National Character Task Force*, dans lequel le centre OBRZ a joué un rôle important tant sur le plan du contenu qu'au point de vue de l'élaboration.

7. Marche à suivre

Il convient à l'heure actuelle d'adopter une stratégie qui vise à résoudre les problèmes évoqués, et ce, dans les domaines les plus divers :

- informer les utilisateurs (comités d'informatique, etc.) de la portée des problèmes liés au code ;
- organiser la sortie avec le jeu de caractères complet 00697 et éventuellement le jeu de caractères 00980 ;
- ne créer désormais que des contrôleurs 3270 « multilingues » (ou leurs équivalents) ;
- rendre les outils (compilateurs, etc.) davantage perméables aux codes ;

- mettre en œuvre de nouvelles applications ou des modifications importantes conformément à [2]
- exercer des pressions sur IBM (par le biais de SEAS, de GUIDE ou dans le cadre de négociations de vente)
- considérer comme irréaliste la volonté de créer une « île suisse », la conversion complète ;
- axer les mesures sur X.400.

7.1 Généralités

Le texte qui précède avait pour but de montrer que les problèmes liés aux jeux de caractères et aux codes sont très profonds. Des solutions sans bavures ne sauraient être trouvées sans la bonne volonté du fabricant (dans le cas qui nous intéresse, en première ligne IBM).

Pendant longtemps, aucune solution véritable ne sera possible. On se contentera de trouver le moyen d'éviter les désagréments les plus fâcheux. Ces mesures doivent cependant être choisies en fonction du but, de manière à bien préparer l'avenir. Pour ce faire, il faut suivre continuellement les développements nouveaux dans ce domaine. Cela veut dire aussi que les mesures prises devront éventuellement être modifiées à nouveau.

Pour ces mesures, on ne pourra que tracer de grandes orientations sans fournir d'indications détaillées. Les solutions extrêmes consistant par exemple à remplacer tous les terminaux « non intelligents » par des ordinateurs personnels « intelligents » (pour pouvoir procéder à des transformations de codes de manière transparente) sont irréalistes. Il faut également rejeter la solution consistant à convertir toutes les données existante., car le temps exigé, à lui seul, empêcherait un fonctionnement ordonné pendant un telle action.

IBM est une société « animée par la volonté de vendre » comme toute autre firme d'ailleurs. Il faut donc insister par tous les canaux possibles sur la nécessité du *National Language Support*. L'attitude courante selon laquelle « on » s'était quand même arrangé jusqu'ici est cynique en cette époque où l'utilisateur final est roi. Les exigences seront entendues ! L'élaboration et la présentation de ces exigences entraînent naturellement des dépenses, mais cela permet aussi de jeter la lumière sur les demandes qui existent véritablement.

Le centre OBRZ est à la remorque de l'évolution « globale ». Il ne faut donc pas engager des dépenses pour des solutions transitoires qui sont de toute évidence à court terme. Au lieu d'adapter les tableaux de conversion au goût du jour, il faudrait les rendre entièrement transparents (comme en MEMO). C'est seulement ainsi que l'on peut éviter la multiplicité des « filtres » qui existent à l'heure actuelle et plus tard introduire un procédé automatique à l'endroit qui convient.

7.2 Recommandations

Il faut d'abord créer les possibilités de sortie avant de développer de nouvelles sources de données. Sinon le procédé en amont ne pourra jamais être contrôlé.

Les écrans avec leurs contrôleurs devraient pouvoir être interrogés – s'il le faut, par une instruction émanant de l'application ! Cela s'applique aussi aux appareils de sortie, qui ne sont pas dotés d'une capacité de fonction définie (toutes les imprimantes Post Script ont une capacité de fonction complète, tandis que les imprimantes IPDS affichent des capacités diverses).

En créant de nouveaux appareils (notamment des contrôleurs d'écran), il faut s'assurer que chaque terminal est capable de travailler avec un autre code (et un autre clavier). Car, devant un même contrôleur, les exigences d'un programmeur et d'un opérateur de traitement de texte ne coïncident pas.

Pour pouvoir convertir des applications, il faut d'abord pouvoir convertir les outils. Les compilateurs doivent au besoin être munis de pré-compilateurs, pour permettre l'exploitation de fichiers dont le codage est différent. Pour DELTA, cela devrait à vrai dire se faire sans peine !

Les nouvelles applications doivent absolument se conformer aux règles [2] établies par IBM, pour permettre l'exploitation de codages et de langues de tous genres. C'est là la meilleure façon de s'accorder avec la voie suivie par IBM. Il faut donc éviter de faire des entorses importantes à ces règles.

J'estime qu'il est inopportun de convertir toutes les activités de traitement de l'information en fonction de la page code 500. Le fonctionnement du système dans son ensemble doit être laissé en l'état. D'où mon opinion que le processus doit s'inscrire dans l'application. Cela signifie toutefois qu'au moins deux codages doivent coexister – et pas seulement de manière passagère.

7.3 Façon de procéder

Lorsqu'on emploie des produits, il ne faut pas perdre de vue leur raison d'être.

MEMO a été créé aux seules fins du « courrier électronique ». Lorsque des problèmes liés au transfert de fichiers surviennent, il est donc absurde de vouloir les lever (en engageant des dépenses exorbitantes). Pour les problèmes propres au « courrier électronique », qui en général sont abordés de façon trop large, il convient de classer les exigences :

- Les exigences minimales s'appliquent à la qualité de la transmission pour l'échange de messages sur divers nœuds de réseau. Dans ce cas, il est jugé acceptable de se limiter au « jeu de caractères syntaxique ».

- Les exigences peuvent être augmentées pour les connexions « locales » – lorsque l'émetteur et le destinataire travaille sur le même nœud de réseau avec le même matériel (écran).

- Ces exigences peuvent également être augmentées pour des connexions sur divers nœuds de réseau, lorsque les mêmes « systèmes terminaux » sont utilisés (par exemple connexion entre Bally Autriche et Bally International à Zurich).

8. Appendice

8.1 Page code 500/1

Pendant l'été de 1987, le jeu de caractères de la page code 500 a été adapté, pour ce qui est de quatre caractères, au jeu de caractères ISO 8859/1. Depuis lors, cette page code est donc appelée 500/1.

Le jeu de caractères de la page code 500/1 complète porte le numéro CGCSGID 697.

	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0	space	&	-	ø	Ø	°	µ	¢	{	}	\	0
x1	req. sp	é	/	É	a	j	~	£	A	J	÷	1
x2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
x3	ä	ë	Ã	Ë	c	l	t	•	C	L	T	3
x4	à	è	À	È	d	m	u	º	D	M	U	4
x5	á	í	Á	Í	e	n	v	§	E	N	V	5
x6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
x7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
x8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
x9	ñ	ß	Ñ	`	i	r	z	¾	I	R	Z	9
xA	[	]	¦	:	«	ª	¡	¬	syl. hy	¹	²	³
xB	.	$	,	#	»	º	¿	|	ô	û	Ô	Û
xC	<	*	%	@	ð	æ	Ð	¯	õ	ü	Ö	Ü
xD	(	)	_	'	ý	¸	Ý	¨	ò	ù	Ò	Ù
xE	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
xF	!	^	?	"	±	¤	®	×	õ	ÿ	Õ	all bits

space espace
req. space (*required space*) espace requis
syl. hy (*syllable hyphen*) césure possible, représentée par -
all bits (*x'ff'*) tous les bits

8.2 Page code 037

La page code 037 est l'équivalent US de la page code 500. Les caractères que l'on trouve habituellement dans le traitement de l'information, c'est-à-dire les majuscules et les minuscules, les chiffres ainsi que les caractères spéciaux { } [ ] / \ ^ ~ ! ` ¢ : . $, # < * % @ ( ) _ ' + ; > = | ¬ ? " sont codés de la même façon que dans le *« Reference Summary »* du système 3270, que les programmeurs connaissent bien. Tous les compilateurs sont par exemple interprétés en fonction de ce code.

Le jeu de caractères de la page code 037 complète porte le numéro CGCSGID 697.

	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0	space	&	-	ø	Ø	°	µ	^	{	}	\	0
x1	req. sp	é	/	É	a	j	~	£	A	J	÷	1
x2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
x3	ã	ë	Ã	Ë	c	l	t	•	C	L	T	3
x4	à	è	À	È	d	m	u	©	D	M	U	4
x5	á	í	Á	Í	e	n	v	§	E	N	V	5
x6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
x7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
x8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
x9	ñ	ß	Ñ	`	i	r	z	¾	I	R	Z	9
xA	¢	!	¦	:	«	ª	¡	[	syl. hy	¹	²	³
xB	.	$	,	#	»	º	¿	]	ô	û	Ô	Û
xC	<	*	%	@	ð	æ	Ð	¯	ö	ü	Ö	Ü
xD	(	)	_	'	ý	,	Ý	¨	ò	ù	Ò	Ù
xE	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
xF	|	¬	?	"	±	¤	®	×	õ	ÿ	Õ	all bits

space espace
req. space (*required space*) espace requis
syl. hy (*syllable hyphen*) césure possible, représentée par -
all bits (*x'ff'*) tous les bits

8.3 Page code 850

La page code 850 est l'une des quatre pages code définies avec le PS/2. Par rapport aux anciennes « pages code PC » 860 (Portugal), 863 (Canada français), 865 (Norvège) et 437 (USA et le reste du monde), tous les signes pour les formulaires, qui passent d'une ligne double à une ligne simple, sont absents.

Le jeu de caractères de la page code 850 complète porte le numéro CGCSGID 980. Étant donné que le jeu de caractères 697 est entièrement contenu dans le jeu de caractères 980, il est recommandé d'utiliser cette page code sur PS/2.

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	ð	Ó	
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▒	┴	Ð	ß	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▓	┬	Ê	Ô	‗
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	Ë	Ò	¾
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	È	õ	¶
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	Á	┼	ı	Õ	§
-6	♠	▬	&	6	F	V	f	v		å	û	ª	Â	ã	Í	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	À	Ã	Î	þ	¸
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	©	╚	Ï	Þ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	®	╣	╔	┘	Ú	¨
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Û	·
-B	♂	←	+	;	K	[	k	{		ï	ø	½	╗	╦	█	Ù	¹
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	ý	³
-D	♪	↔	-	=	M	]	m	}		ì	Ø	¡	¢	═	¦	Ý	²
-E	♫	▲	.	>	N	^	n	~		Ä	×	«	¥	╬	Ì	¯	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	»	┐	¤	▀	´	

8.4 ISO 8859/1

Cette page code contient le même jeu de caractères que la page code 500/1. La désignation complète est la suivante :

ISO 8859/1.2 : Traitement de l'information - Jeux de caractères graphiques codés sur un seul octet. Partie 1 : Alphabet latin numéro 1.

	2x	3x	4x	5x	6x	7x	Ax	Bx	Cx	Dx	Ex	Fx
x0	space	0	@	P	`	p	req. sp	°	À	Đ	à	ð
x1	!	1	A	Q	a	q	¡	±	Á	Ñ	á	ñ
x2	"	2	B	R	b	r	¢	²	Â	Ò	â	ò
x3	#	3	C	S	c	s	£	³	Ã	Ó	ã	ó
x4	$	4	D	T	d	t	¤	´	Ä	Ô	ä	ô
x5	%	5	E	U	e	u	¥	µ	Å	Õ	å	õ
x6	&	6	F	V	f	v	¦	¶	Æ	Ö	æ	õ
x7	'	7	G	W	g	w	§	·	Ç	×	ç	÷
x8	(	8	H	X	h	x	¨	¸	È	Ø	è	ø
x9	)	9	I	Y	i	y	©	¹	É	Ù	é	ù
xA	*	:	J	Z	j	z	ª	º	Ê	Ú	ê	ú
xB	+	;	K	[	k	{	«	»	Ë	Û	ë	û
xC	,	<	L	\	l	|	¬	¼	Ì	Ü	ì	ü
xD	-	=	M	]	m	}	syl. hy	½	Í	Ý	í	ý
xE	.	>	N	^	n	~	®	¾	Î	Þ	î	þ
xF	/	?	O	_	o	delete	¯	¿	Ï	ß	ï	ÿ

space espace
req. space (*required space*) espace requis
syl. hy (*syllable hyphen*) césure possible, représentée par -
all bits (*x'ff'*) tous les bits

8.5 Jeu de caractères multinational /36

Cette page code coïncide grosso modo avec le prédécesseur de la code page 500, la page code 256.

	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0	space	&	-	ø	Ø	°	µ	¢	{	}	\	0
x1	req. sp	é	/	É	a	j	~	£	A	J	÷	1
x2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
x3	ä	ë	Ä	Ë	c	l	t	Pt	C	L	T	3
x4	à	è	À	È	d	m	u	©	D	M	U	4
x5	á	í	Á	Í	e	n	v	§	E	N	V	5
x6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
x7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
x8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
x9	ñ	ß	Ñ	`	i	r	z	¾	I	R	Z	9
xA	[	]	¦	:	«	ª	¡	¬	syl. hy	1	2	3
xB	.	$	,	#	»	º	¿	|	ô	û	Ô	Û
xC	<	*	%	@	ð	æ	Ð	¯	ö	ü	Ö	Ü
xD	(	)	_	'	≤	¸	Ý	¨	ò	ù	Ò	Ù
xE	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
xF	!	^	?	"	±	¤	®	×	õ	ÿ	Õ	all bits

space espace
req. space (*required space*) espace requis
syl. hy (*syllable hyphen*) césure possible, représentée par -
all bits (*x'ff'*) tous les bits

8.6 Jeu de caractères autrichien/allemand /36

Par rapport au « jeu de caractères multinational », les caractères suivants présentent un codage divergent :
{ } [ ] ß ä Ä ö Ö ü Ü ~ @ § ¦ \

	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0	space	&	-	ø	Ø	°	µ	¢	ä	ü	Ö	0
x1	req. sp	é	/	É	a	j	ß	£	A	J	÷	1
x2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
x3	{	ë	[	Ë	c	l	t	Pt	C	L	T	3
x4	à	è	À	È	d	m	u	©	D	M	U	4
x5	á	í	Á	Í	e	n	v	@	E	N	V	5
x6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
x7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
x8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
x9	ñ	¨	Ñ	`	i	r	z	¾	I	R	Z	9
xA	Ã	Ü	õ	:	«	ª	¡	¬	syl. hy	1	2	3
xB	.	$	,	#	»	º	¿	|	ô	û	Ô	Û
xC	<	*	%	§	ð	æ	Ð	¯	¦	}	\	]
xD	(	)	_	'	≤	¸	Ý	¨	ò	ù	Ò	Ù
xE	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
xF	!	^	?	"	±	¤	®	×	õ	ÿ	Õ	all bits

space espace
req. space (*required space*) espace requis
syl. hy (*syllable hyphen*) césure possible, représentée par -
all bits (*x'ff'*) tous les bits

8.7 Glossaire

Derived from [1,3].

ANSI American National Standards Institute.

ASCII American National Standard Code for Information Interchange. A coded character set consisting of 128, 7-bit characters. There are 32 control characters, 94 graphic characters, the space character and the delete character.

CECP Country Extended Code Page. The extended code page of countries with code pages based on a western latin alphabet.

CGCSGID Coded Graphic Character Set Global IDentifier. A number given to a set of graphic characters like 00697 for the set of characters contained in code page 500.

coded character set A specific set of bit patterns to which specific graphic meanings and control meanings have been assigned. This is synonymous with *code*.

code page A specification of code points for each graphic character in a set or in a collection of graphic character sets. Within a code page, a code point can have only one specific meaning.

control character A specific bit pattern with an assigned control meaning. Contrast with *graphic character*.

CPGID Code Page Global IDentifier. The number of the code page given by IBM. → code page 500 or code page 037.

EBCDIC Extended Binary Coded Decimal Interchange Code. A coded character set consisting of 8-bit coded characters.

GCGID Graphic Character Global IDentifier. A name for a graphic like LA010000 for the letter a, or ND021000 for ^{2} (superscript 2).

graphic character ISO: A character, other than a control character, that is normally represented by a graphic.

invariant character set (1) A character set, such as the Syntactic Character Set, that does not change from code page to code page. (2) A minimum set of characters that is available as part of all character sets.

IPDS Intelligent Printer Data Stream. Datastream consisting of *structured fields* to control a printer. Functions are grouped into classes. A specific printer normally can perform only a set of function classes, not the total set of functions specified in IPDS.

language ISO: A set of characters, conventions, and rules, that is used for conveying information. The three aspects of language are pragmatics, semantics, and syntax.

NLS National Language Support. The ability for a user to communicate with applications in a language other than US English.

8.8 Bibliographie

[1] National Language Information and Design Guide Volume 1: Designing enabled Products, Rules and Guidelines; IBM publication SE09-8001

[2] National Language Information and Design Guide Volume 2: Left-to-Right Languages and double byte Character Set Languages; IBM publication SE09-8002

[3] IBM Switzerland FSC Cookbook: Swiss Terminal / Swiss Character Set Support; Ulrich Abderhalden. Available on request from SE at Zürich.

[4] Actes du congrès du 28.10.87: le *National Language Support* dans le matériel et les logiciels IBM ; W. Dormann / U. Abderhalden.

[5] ISO 8859/1 : ISO 8859/1.2 : Traitement de l'information - Jeux de caractères graphiques codés sur un seul octet. Partie 1 : Alphabet latin numéro 1.

[6] Desmond Morris : *Der Mensch, mit dem wir leben* [L'Homme avec lequel nous vivons], Droemer & Knaur, 1978.

[7] *Der Grosse Brockhaus*, édition commémorative, 1978.

[8] Carl Faulmann : *Das Buch der Schrift* [Le livre de l'écriture], Kaiserlich-Königliche Hof-und Staatdruckerei, Vienne, 1880, réimpression en 1985 par Greno, Nördlingen.

SHARE Europe (SEAS)

Incorporated in The Netherlands

Founded in 1961

Published by

SHARE Europe Headquarters
17, Rue Pierres-du-Niton
CH-1207 Geneva
Switzerland

Phone +41 (22) 735 40 66
FAX +41 (22) 735 47 51

This document was printed by and at the expense of
OBRZ, Jungholzstrasse 43, CH-8050 Zürich obrz.