



CLUB INFORMATIQUE DES GRANDES ENTREPRISES FRANÇAISE - 21,
avenue de Messine, 75008 Paris, tél. 256.94.00 poste 24.94

Le 06.06.77

PORTABILITÉ DES APPLICATIONS INFORMATIQUES - Rapport nO 2 - INTRODUCTION

-

Le CIGREF avait eu l'occasion dans un document de réflexion publié en 1976 sur la portabilité des applications informatiques, d'évoquer les difficultés principales que ce problème posait, et de suggérer un certain nombre d'actions prioritaires à ses yeux dans ce domaine.

Depuis cette date, la poursuite de la réflexion, comme les faits eux-mêmes, ont montré l'importance considérable et croissante du problème. Ce dernier qui a d'ailleurs toujours existé dans le passé, deviendra de plus en plus contraignant dans l'avenir, car l'absence de portabilité des applications informatiques obérera de plus en plus, du fait des volumes en cause, la liberté de choix tant de l'utilisateur pour ses moyens, que du constructeur pour sa politique produits.

L'investissement consacré à la préparation de tels outils d'aide à la portabilité des applications paraît d'autant plus utile qu'il peut réduire dans des proportions notables le coût d'une transposition et, compte-tenu du nombre d'opérations de ce type à réaliser, entraîner une économie globale importante au plan national.

D'ailleurs les utilisateurs importants seront conduits à disposer d'un parc hétérogène nécessitant la coexistence de matériels de constructeurs différents, mettant en œuvre parfois les mêmes applications et devant donc utiliser pour cela des programmes compatibles entre eux. Le besoin de portabilité des programmes est donc permanent chez les grands utilisateurs et ne se limite pas aux problèmes ponctuels de la transposition nécessitée par la transposition de matériels.

Certains des moyens proposés d'ailleurs par le CIGREF sont souvent déjà envisagés, voir définis ou même déjà existants dans certains cas.

Les grandes entreprises, -cependant toutes capables, comme Les - Administrations, de régler individuellement leurs problèmes, attendent des Pouvoirs Publics la définition d'une politique claire qui leur apporte une aide véritable dans ce domaine. Cette aide ne doit pas être ponctuelle à chaque utilisateur, mais au contraire s'appliquer à la réalisation et à la promotion d'outils tels que ceux évoqués ci-après, dans un deuxième rapport du groupe de travail "Portabilité" du CIGREF qui fait suite à celui déjà publié dans ces colonnes en septembre 1976.

Réf. 77 224

le 23.03.77

- PORTABILITE DES PROGRAMMES -

Réflexion sur trois sujets prioritaires
édités dans le rapport "Portabilité" de février 1976

Ce rapport fait suite au rapport de synthèse du groupe "Portabilité des programmes" de février 1976.

Devant l'absence de réaction efficace aux propositions de ce rapport de la part tant des organismes de recherches que des constructeurs et des Pouvoirs Publics, et tenant compte de l'impact favorable rencontré dans l'ensemble de la profession, le CIGREF a décidé de préciser les objectifs des utilisateurs sur les points prioritaires de son précédent rapport.

Le seul but de ces propositions est de rendre moins dépendants les uns des autres les utilisateurs et les constructeurs, qui doivent en permanence essayer d'accorder leurs évolutions alors que les critères de base de ces évolutions sont totalement différents. De plus, l'existence d'une réelle portabilité des programmes d'application mettrait les Pouvoirs Publics en position plus confortable pour définir les meilleures orientations à proposer à la profession, les contingences instantanées pouvant alors être négligées.

Pour définir la portabilité, nous prendrons une définition du Swedish Board of Technical Development :

"Portability is taken to mean the possibility of running programs and data unchanged on different computers"

que l'on peut traduire dans les termes suivants

"La portabilité est la possibilité de faire exécuter des programmes sur des ordinateurs différents sans modification des programmes ni des données".

Nous considérons qu'il faut remplacer le mot "ordinateur" par "système" au sens large, et le mot "programmes" par "applications". Cela conduit à la définition suivante

"La portabilité est la possibilité de faire exécuter **des** programmes sur des systèmes différents sans modification des applications ni des données".

Nous dirons également que l'absence de portabilité est cause de nombreux inconvénients pour les utilisateurs. Nous ajouterons aussi qu'elle impose aux constructeurs des dépenses considérables en traducteurs et émulateurs de toutes sortes pour permettre aux utilisateurs des changements de matériel rendus obligatoires soit par l'obsolescence technique, soit par un développement important de la charge à écouler.

Enfin, le CIGREF émet les recommandations indiquées ci-après, qui seront expertisées par le CXP* puis proposé aux diverses associations d'utilisateurs. C'est une action commune de ces associations d'utilisateurs qui permettra de créer, de promouvoir et de maintenir les outils de portabilité proposés.

NORMALISATION DU COBOL

1 - Objectif

1.1 - Portabilité

La normalisation recherchée par les utilisateurs a pour but de permettre le passage(des programmes sources d'un compilateur à un autre (quel qu'ils soient), sans modification ou presque, et de faciliter les maintenances et la formation. La normalisation proposée vise, comme toute normalisation de langages, à définir celui-ci sans ambiguïté. Il y a actuellement malgré la norme 74, carence en la matière. En effet, la norme (ISO et AFNOR) actuelle de Cobol permet de proposer à l'utilisateur sous le même nom, des langages plus ou moins différents les ~s des autres puisque, sans parler des ajouts toujours autorisés, les diverses combinaisons possibles, de niveaux de chacune des huit parties du Cobol permettent de l'ordre de 100 000 réalisations différentes.

Dans ces conditions, l'utilisateur se trouve confronté à des problèmes redoutables dès qu'il change de matériel (quelquefois même quand il change de système d'exploitation sur un matériel donné).

C'est pourquoi, sans remettre en cause profondément la norme ISO, il apparaît nécessaire de définir enfin un (ou quelques) langage(s) Cobol, c'est-à-dire une (ou quelques) association(s) bien définie(s) de niveaux des huit parties de la norme Cobol, c'est-à-dire. un Cobol enfin portable. Par ailleurs, cette normalisation ne doit pas entraîner des baisses de performance sans rapport avec les avantages qu'elle apporte.

1.2 - Méthodologie

Afin de faciliter les travaux de maintenance et l'efficacité de la programmation, elle vise à permettre des apports méthodologiques au niveau de l'entreprise: par exemple introduction de contraintes particulières, ne mettant pas en cause la portabilité d'un compilateur à un autre, relatives à l'emploi d'un dictionnaire de données, à l'utilisation privilégiée de telle instruction plutôt que de telle autre (amélioration globale des performances dans une utilisation de caractéristiques données propres à l'entreprise) ou à l'utilisation de telle ou telle méthode de conception de projet.

1.3 - Formation

La formation des programmeurs à un cobol normalisé ne doit pas ~tre le seul but de la formation, bien qu'une formation identique des programmeurs de gestion soit intéressante.

C'est en effet l'esprit "normalisation" qui doit être enseigné, en particulier par les écoles et l'université, la normalisation devant aborder aussi bien **les** données d'analyse, les dictionnaires de données, la programmation, les méthodes de diagnostic et d'essai, ...

2 - Etat de la situation

2.1 - De nombreuses entreprises ont cherché à atteindre tout ou partie de ces objectifs.

- Objectif 1.1-Portabilité entre deux compilateurs (deux langages) connus d'avance.
- Objectif 1.2 - Méthodologie avec un compilateur et donc un langage Cobol donnés.
- Objectif 1.3 - Formation en éditant une brochure particulière de description du langage pour leurs programmeurs.

o o
0

Un projet intéressant pour l'aspect portabilité a été réalisé par la Caisse Nationale de Crédit Agricole. Toutefois, la norme maison n'est utilisée que pour les programmes à caractère général.

Les expériences de portabilité réalisées après coup entre deux compilateurs ont utilisé des langages de traduction par table. Celles-ci sont réalisées pour permettre le passage du compilateur A vers le compilateur B. Dans tous les cas, un très important travail manuel reste à faire.

2.2 - A l'heure actuelle, un effort de normalisation qui a été entrepris pour les Administrations par le BIADI* a conduit à un produit appelé COBOL C qui semble répondre de manière relativement satisfaisante à l'ensemble des trois aspects portabilité, méthodologie et formation.

Les réalisateurs de ce produit ont en effet clairement défini trois niveaux Cobol (donc 3 langages. Les niveaux 1 et 3, c'est-à-dire les deux extrêmes, sont définis dans des brochures du BIADI, le niveau moyen faisant l'objet d'une rédaction du Centre Electronique de l'Armement.

2.21 - Aspect Portabilité

On peut noter au sujet du Cobol C que presque toutes les options retenues au niveau le plus élevé (niveau 3) sont compatibles avec l'ensemble des compilateurs ANS actuels (sauf ceux des mini-ordinateurs qui semblent relever du niveau 1 ?) les quelques différences existantes sont répertoriées et donc faciles à corriger manuellement.

De plus, un examen rapide indique que Cobol C niveau 3 est conforme à la norme AFNOR 76 et que son emploi n'a pas altéré d'une manière notable les performances des programmes obtenus après compilation, dans les installations qui l'utilisent. Une expertise du CXP pourrait confirmer ou infirmer ces impressions. La conformité du programme écrit au standard Cobol C est vérifiée par un pré-compilateur (FILTRE). Ce point est particulièrement important, la tendance naturelle en programmation n'étant heureusement pas de relire constamment les brochures, ce qui entraînerait une baisse considérable de rendement. Par contre, les mauvaises habitudes se prennent vite. Le passage Filtre avant compilation permet d'éviter un glissement de la rédaction des programmes.

2.22 - Aspect Méthodologique

L'existence du pré compilateur (filtre) écrit lui-même en Cobol C, permet de créer facilement un "super filtre" maison introduisant des contraintes spécifiques. Une utilisation de filtre dans cette voie a été faite par certains membres du CIGREF qui comptent y ajouter la consultation d'un 'dictionnaire' de données.

* Bureau Inter-administration de Documentation et d'Information

De plus, aucune particularité de Cobol C n'est de nature à empêcher l'emploi des méthodes d'analyse (l'article "Une tentative Pragmatique d'Approche Globale des Futurs Systèmes Informatiques" de Jean-Claude GUEZ, Ingénieur en Chef au Cabinet Arthur Andersen, paru dans Informatique et Gestion~ montre que Cobol et méthodes d'analyse ne sont pas antinomiques) ou d'autres précompilateurs comme la DATA AUTOMATIQUE d'EDF/GDF...).

Il serait d'ailleurs souhaitable tant pour les réalisateurs que pour leurs clients, que les générateurs de Cobol des méthodes d'analyses produisent du Cobol C.

2.23 - Aspect Formation

Un manuel de formation "support de cours Cobol C" est disponible au BIADI pour assurer le premier niveau de formation.

Les notes Cobol C (plusieurs par niveau) sont à disposition pour préciser cette première formation.

Le programmeur dispose ensuite des "guides du programmeur Cobol" du fournisseur de matériel.

Il semble donc que l'emploi de Cobol C puisse répondre à l'ensemble des problèmes que se posent les utilisateurs. De plus, un traducteur par table appelé TRADUC, permet de faciliter le passage des Cobols constructeurs au Cobol C.

3- Que reste-t-il à faire?

Bien que Cobol C réponde dans son ensemble au problème des utilisateurs, certains problèmes inhérents à Cobol ne sont pas résolus par ce standard.

On peut noter par exemple le problème des sorties sur imprimantes (saut de page, ...)
-il semble que la voie à suivre est que le seul saut canal autorisé soit le saut de page et que le reste se fasse par avancement programmé, toutefois, ce point demande une étude précise-. Retenons également l'anarchie complète des "noms systèmes" ou le manque de rigueur de la normalisation en ce qui concerne le mode séquentiel indexé.

D'autres problèmes existent qui méritent une action résolue des utilisateurs. Leur inventaire, l'étude de solutions possibles et de l'action politique à mener pourraient être réalisés par un groupe commun CIGREF-CXP

Enfin, dans certains cas d'incompatibilité simple (nominal key au lieu de symbolic key) peut-être serait-il possible que le programme filtre correspondant au compilateur en cause prenne la transformation à sa charge. Là encore, un groupe commun des utilisateurs et du producteur pourrait apporter une réponse.

Conclusion

Il apparaît que le problème de la normalisation du Cobol peut être en grande partie résolu par l'adoption de Cobol C. Toutefois, il reste un certain nombre de problèmes pour lesquels une action énergique des groupements d'utilisateurs et des pouvoirs publics est nécessaire. La définition précise de ces problèmes ainsi que de l'action souhaitable pourrait être étudiée dans le cadre du groupe commun.

Par ailleurs, -une étude CIGREF-CXP pourrait permettre la définition d'une action méthodologique annexe déjà entreprise d'ailleurs par des membres du CIGREF utilisateurs ou non de Cobol C.

Notons par ailleurs, que l'action méthodologique ne doit pas se cantonner au niveau de la programmation et qu'une normalisation Cobol est d'autant mieux acceptée par les programmeurs d'une entreprise qu'une normalisation existe déjà pour la présentation des dossiers d'analyse, l'emploi des mots dans l'entreprise (dictionnaire de données), etc ...

* cf. Informatique et Gestion n0 85 - mars 1977.

PORTABILITE DES FICHIERS

1 - Fichiers bandes magnétiques

L'écriture sur bande magnétique est assez bien normalisée au niveau des supports physiques en particulier.

Toutefois, le code d'enregistrement varie d'un fournisseur à l'autre ou, chez certains fournisseurs qui offrent une possibilité de choix, d'une application à l'autre. On trouve en effet simultanément les codes EBCDIC, ASCII,

Il Y a également des divergences au niveau du contenu, de la forme et du nombre des labels.

Le but que se proposent les utilisateurs est d'obtenir

- en première étape, une normalisation du code, de la forme et du contenu des labels, ceux-ci fournissant automatiquement au système le code dans lequel sont inscrites les données. Les systèmes opératoires devront être capables d'acquérir directement cette information.

Le code est à définir ainsi que le nombre et le contenu en conformité maximum avec la norme AFNOR pour le nombre et le contenu des labels, normes AFNOR 762 ; pour le code la norme ISO 646 définit un code 7 bits. Peut-être une norme 8 bits peut-elle apparaître devant la généralisation de l'EBCDIC).

- en deuxième étape, l'extension du code retenu après l'étude à l'ensemble des données.

Par ailleurs, les utilisateurs estiment que la possibilité d'écrire des enregistrements "spanés" sur bande est à interdire de façon absolue pour les fichiers utilisateurs.

2 - Fichiers sur disques magnétiques

Les organisations sur disques sont les suivantes -
séquentielles

- directe par adressage absolu
- par adressage relatif (par piste, par bloc) -
- séquentielle indexée.

Etat du problème et propositions générales nécessaires à la portabilité des fichiers

- a) En IBM, l'organisation séquentielle indexée est passée en classe C et donc risque d'être abandonnée au profit d'une organisation particulière sur VSAM. Il faut toutefois noter qu'alors que le séquentiel indexé ne faisait pas partie de la norme COBOL 1968, il en fait partie en COBOL ANSI 74, fait confirmé par la norme ISO 76. Nous sommes donc amenés à conserver cette organisation.
- b) L'organisation directe par adressage relatif par bloc qui impose des fichiers de format fixe, est la seule qui ne soit pas liée au type de support externe et n'entraîne donc pas de recomposition en cas de changement de configuration. Elle doit être préconisée dans tous les cas de fichier accès direct.
- c) L'adressage direct absolu est à proscrire en règle générale, les programmes étant alors beaucoup trop liés au matériel périphérique utilisé .

- d) Certains constructeurs mettent par ailleurs à disposition des utilisateurs une organisation utilisée en général pour les seules bibliothèque les fichiers partitionnés. Etant donné 'lue l'avantage de cette ~organisation n'est pas déterminant, cette organisation sera refusée-pour les fichiers d'application

Solution proposée pour assurer la portabilité physique des fichiers disques

Compte-tenu des paragraphes précédents, la meilleure solution pour assurer la portabilité est de le faire par le passage à un fichier séquentiel sur bande magnétique créé par exemple à chaque archivage de sécurité ou à chaque vidage de disques sur bandes magnétiques.

Le mode d'établissement sera celui de la réorganisation des fichiers - aucun problème pour les fichiers séquentiels,
lecture en "mode séquentiel" des fichiers séquentiels indexés par recopie sur bande,
- lecture séquentielle des données des fichiers d'organisation directe.

Afin de permettre ensuite la réorganisation du fichier sur disques, il convient de créer sur support bande magnétique normalisé (voir §1). Deux fichiers :
- le premier contenant toutes les informations nécessaires à la reconstitution du fichier sur disque (type d'organisation, longueur des enregistrements...),
- le second contenant les données enregistrées séquentiellement.

Avantages et limites de la solution

La solution préconisée n'étudie pas le cas des fichiers "base de données". Le principe de la réorganisation doit permettre de le résoudre de la même manière 'lue les autres.

Etant donné 'lue les supports à accès directs ne sont physiquement pas normalisés (sauf cas particulier de supports compatibles), seule la recopie bande semble apte à apporter une solution générale bien qu'elle oblige à faire deux copies (disque à bande puis bande à disque). Dans le cas où le support disque peut être porté d'un système à l'autre, le problème du code d'enregistrement se pose dans les mêmes termes 'lue pour les bandes.

La solution à la portabilité physique de fichiers disques suppose 'lue la normalisation des fichiers sur bande ait été obtenu.

Comme pour le problème bande, la réalisation de ce système de recopie sur bande normalisé doit être imposé aux constructeurs. Il serait même intéressant que ce système soit le seul système de sauvegarde pour le fichier clients admis au niveau des systèmes opératoires •

- Enfin, si cette solution permet de résoudre 70 à 80 % des problèmes de portabilité, un certain nombre de cas particuliers relèveront encore longtemps de solutions spécifiques.

DEFINITION D'UN LANGAGE EXTERNE DE DESCRIPTION DE TRAVAUX

La description des travaux s'effectue au moyen d'un "Langage de commande" ("Job Control Language", "cartes de contr8le") très lié au système opératoire des machines. Ce langage LdC a pour but:

- D'alléger la tâche des exploitants en leur permettant de préparer, en quelque sorte à l'avance, le dialogue opérateur-machine (ce qui va même jusqu'à le .supprimer).
- De simplifier dans une certaine mesure la gestion de la bandothèque afin d'éviter des erreurs génératrices de baisse du rendement global des machines (catalogue des programmes et fichiers).

Il est par contre évident que ce LdC étant très lié au système opératoire et s'adressant à des spécialistes,

- d'une part, est essentiellement non portable,
- d'autre part, n'est absolument pas tourné vers le problème du client occasionnel du centre de calcul.

Par ailleurs, il évolue aussi vite, ou presque, que les systèmes opératoires.

C'est pourquoi, sans toucher au JCL lui-même, il nous semble indispensable de définir un Langage Externe de Description des travaux (L.E.D.), indépendant des matériels et logiciels utilisés, à l'usage des clients occasionnels des centres de calcul. Ce L.E.D. doit être vu comme un langage "source" normalisé.

Un tel langage apporterait plusieurs avantages à la profession:

- 1 - Ce langage étant universel et stable, éviterait les passages multiples pour erreurs de cartes paramètres et les ingénieurs sortant d'école, qui sont actuellement aptes à écrire un programme, seraient également capables de le faire exécuter.
- 2 - L'existence de ce langage universel ferait sauter l'un des plus gros verrous qui s'opposent au changement éventuel de fournisseur ou de système opératoire la nécessité de recommencer la formation au JCL des ingénieurs des services clients.
- 3 - Ce langage pourrait également faciliter les essais de programmes des grandes chaînes d'applications.

Le cahier des charges du L.E.D. serait le suivant : à partir des informations fournies par le client (indiquées ci-après) et de l'ensemble des renseignements disponibles dans les divers catalogues et bibliothèques de l'ordinateur relatifs tant aux programmes qu'aux fichiers, un programme interface, créerait le JCL constructeur nécessaire à l'exploitation de l'ordinateur, à moins que le réalisateur du JCL introduise directement dans le système les procédures nécessaires.

Les informations qui ne pourraient provenir ni des données utilisateurs ni des catalogues seraient fournies par des options de générations de ce programme permettant de l'adapter aux conditions d'exploitation de l'entreprise.

Au cas où le travail sortirait du cadre normal défini par l'entreprise, il devrait être impérativement rejeté.

Par ailleurs, il importe que la rédaction des commandes du L.E.D. soit stable dans le temps et non liée à une version ou à une autre des systèmes opératoires. Au maximum des cartes supplémentaires pourraient apparaître dans le temps, mais il doit toujours être possible d'exécuter des travaux ne présentant pas ces informations complémentaires qui devront être remplacées par des options implicites. Nous avons appris qu'une étude sur le sujet est menée par le Lund University Center; il serait intéressant de connaître d'autres organismes intéressés par des études dans ce domaine.

De plus, le calcul des longueurs de fichiers en pistes et cylindres doit être exécuté par le programme interface grâce à des informations fournies par les équipes système ou l'exploitation, l'utilisateur final ne devant en aucun cas être concerné par le matériel périphérique installé. Seuls le nombre d'articles logiques et leurs longueurs le concernent.

Enfin, il est nécessaire que lors du retour du travail, le programme interface reprenne la main afin de fournir à l'utilisateur et en clair:

- la définition du travail qu'il a formulée ainsi que les options implicites retenues,
- toutes les données comptables,
- les messages d'erreurs. .

Liste des informations fournies par le client : -

Son identité comptable, " u-'budgétaire .

- Le lieu de restitution de son travail (remote).

Les noms des programmes et sous-programmes à traiter et les éléments nécessaires au calcul de la taille mémoire nécessaire, associés aux noms des programmes qui en autorisent l'exploitation, et un code relatif à leur exécution (code retour indiqué par lui en cas de fin normale, par l'OS en cas de fin anormale).

Les noms des fichiers d'entrée associés aux programmes (noms internes aux programmes et noms-système). Les noms sont complétés par une indication de génération (dernière, dernière - 1, dernière - n, ...) si nécessaire avec le mode d'utilisation s'il est libre.

- Les noms (programmes et systèmes) de fichiers de sortie associés aux programmes avec l'indication du (ou des) mode(s) d'utilisation autorisé(s) (séquentiel ou non séquentiel, ...) la longueur en nombre d'articles (avec taille moyenne, minimale et maximale des articles).

Pour les fichiers à génération à la première création, toutes ces données sont à fournir ainsi que l'indication "fichier à génération". Ensuite, seuls le nom et optionnellement la longueur en nombre d'articles sont à fournir même en création.

- Les noms de programmes et fichiers dont il désire le catalogage en cas de fin normale du travail et la durée de conservation désirée. Les noms de programmes et fichiers dont il désire pouvoir demander une liste ou une analyse en cas de fin anormale ; Ces derniers seraient conservés pendant une durée fixée au sein de l'entreprise.

- L'accessibilité aux fichiers créés (libre ou interdite).

- L'urgence demandée au cas où l'exploitation fournirait plusieurs options possibles.

Toutes les options d'environnement nécessaires pour éviter des problèmes de blocage ou autre devront être indiquées au programme interface par le centre de calcul.