



J.-P. Vickoff :
*« je me suis rendu compte
 que les programmeurs
 perdaient leur temps dans des tâches
 répétitives et sans intérêt »*

“En programmation, les problèmes de gestion sont souvent des problèmes prosaïques, mais leur spécification est très complexe. »

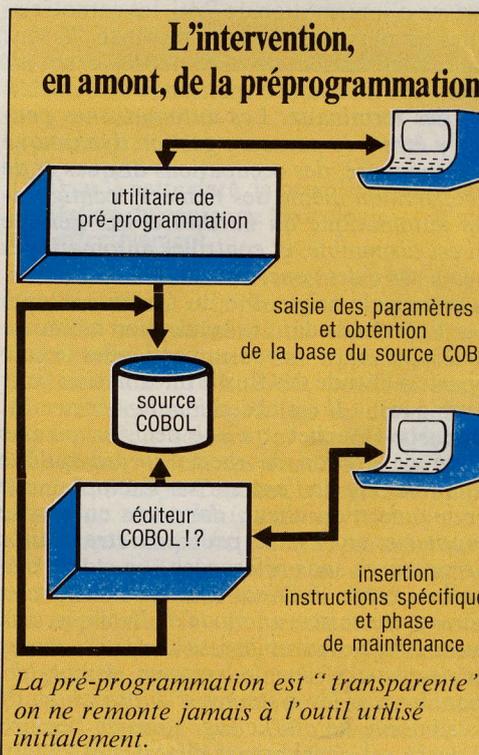
Sans vouloir cultiver le paradoxe, Jean-Pierre Vickoff constate que dans la majorité des cas, ce sont les descriptions de données et leur manipulation qui, en Cobol, accaparent 75 % de l'activité d'un service informatique. Dans le même temps, les phases de l'analyse, pourtant déterminantes, ne représentent qu'une part réduite de la programmation. « Avant d'arriver à la procédure proprement dite, le programmeur doit faire de multiples descriptions. Ainsi, pour parvenir à la mise en œuvre d'une opération — additionner un nombre dans un compteur par exemple — il faut au minimum plusieurs dizaines de lignes ! Quand il s'agit d'une description de fichier, c'est évidemment encore plus lourd. »

Autodidacte devenu « chercheur indépendant en méthodologie », à l'âge de trente-deux ans, J.-P. Vickoff a lui-même acquis l'expérience des avatars du Cobol, langage lourd, comportant de graves lacunes et pourtant indispensable, dans la mesure où il est « universel » : plus de 90 % des centres de développement pratiquent le Cobol, créé il y a bientôt vingt-deux ans. « Au départ raconte J.-P. Vickoff, qui a une formation en informatique à l'IBM et Arts et Métiers (CNAM), depuis 1966, je me suis rendu compte que les programmeurs perdaient leur temps dans des tâches répétitives et sans intérêt ni pour eux-mêmes ni pour le problème à résoudre. L'écriture des trois premières divisions de Cobol.

La “préprogrammation” un outil pour optimiser la programmation en Cobol

Comment remédier au profond malaise qui affecte les « OS de l'informatique », ceux qui, sur le terrain, accomplissent des prouesses en écrivant des programmes en Cobol et en assurant leur maintenance. Pourtant longue est la chaîne de génie logiciel qui comprend de nombreux outils généraux — générateurs de programme, précompilateurs, préprogrammateurs — capables de réduire la dépendance des utilisateurs à l'égard des contextes machine et langage de programmation.

Jean-Pierre Vickoff propose ici sa démarche, dont l'avant-gardisme ne sera pas toujours apprécié. Pour beaucoup de professionnels, ce « méthodogicien » tente seulement de maintenir l'usage, envers et contre tous les langages évolués, d'un Cobol aux abois...





« Débarrassé de la programmation l'agent peut se consacrer au travail intelligent »

Situation intolérable

De même, des traitements simples, comme les cumuls, l'édition, les ruptures, la cinématique des fichiers, demandent à chaque fois une résolution nouvelle. »

De surcroît, la rédaction des ordres d'accès, déjà passablement ennuyeuse, est devenue réellement fastidieuse dans des applications nécessitant outre des descriptions d'« interfaces », l'utilisation d'instructions non normalisées, composées de cinq, six, dix paramètres, voire plus. C'est le cas dans la gestion des bases de données, des fichiers spéciaux, des écrans et du télétraitement.

Cette situation « intolérable » justifie l'engouement actuel des spécialistes pour le développement de langages comme le Pascal ou l'APL, plus souples ou plus puissants. Ce que ne renie pas J.-P. Vickoff, qui a été un homme de terrain pendant sept ans — d'abord comme informaticien, ensuite comme utilisateur dans un établissement financier, avant de redevenir « technicien », chargé des questions de méthode : « *Je ne suis pas formellement opposé à cette évolution. Mais il faut être réaliste et tenir compte du poids « historique » du Cobol et des réticences des intéressés... Le nombre potentiel de programmes à convertir dissuaderait les meilleures volontés. En outre, si les nouveaux langages sont incontestables, avantageux au niveau de la procédure, ils ne modifient pas, fondamentalement, notre façon de programmer. Sans compter que leur emploi ne sera généralisé que dans une perspective lointaine. Or c'est immédiatement et de manière pratique qu'il faut résoudre les problèmes de productivité qui se posent aux équipes de développement d'applications.* »

Les générateurs de programmes sont dans une impasse...

Il y a bien des « générateurs de programmes » mais J.-P. Vickoff, qui en a étudié une douzaine, considère qu'ils sont dans une impasse : « *La structure des programmes résultants comporte de nombreuses failles : manque de commentaires, présence de nombreux alters, les identifiants de données et de paragraphes ne sont pas parlants c'est-à-dire « mnémoniques » et, de toute façon, excessivement longs.* » Générant des produits « complets », ils prennent en compte jusque et y compris la partie spécifique d'un programme, donc du problème (et celui-ci est souvent complexe). « *Je leur reproche également, argumente notre interlocuteur, d'appauvrir le travail du programmeur, en le dépouillant de sa liberté.* »

Autre solution, couramment avancée par les « avant-gardistes », surtout des universitaires : le recours à des méthodes plus ou moins contraignantes, comme la programmation structurée. Mais force est de consta-

ter qu'elles n'ont pas, à l'heure actuelle, « révolutionné » les centres informatiques, comme le prétendaient leurs auteurs, même si elles apparaissent indispensables pour « programmer convenablement ». A ce propos, J.-P. Vickoff est formel : « *Nulle part les techniques de structuration, associées au Cobol, ne se sont révélées, à elles seules, suffisantes en la matière. Aucune d'entre elles ne dispensera le programmeur de décrire ses manipulations : une ouverture ou une fermeture de fichier, des ordres sur ce fichier, etc. Au contraire, elles n'ont fait que souligner le vide dans le domaine des outils d'aide à la programmation.* »

Eliminer les tâches répétitives

Cette réflexion est à l'origine de la prise de conscience, chez J.-P. Vickoff, de l'importance de la préprogrammation. Elle serait prise en charge par des utilitaires, — objet de ses travaux depuis trois ans — qui fourniraient automatiquement les parties répétitives, et uniquement elles, d'une application. Les outils proposés interviennent après l'analyse du problème et en amont de la programmation. « *Ce n'est pas tant une démarche nouvelle qu'une synthèse de bonnes recettes de cuisine et d'« astuces » traditionnellement utilisées dans les services informatiques.* »



Quand ils ont à traiter des problèmes très proches, la plupart des programmeurs n'hésitent pas, en effet, à « piquer » dans des procédures déjà écrites, par le biais de l'éditeur. « *Cet usage rapporté à une méthode comme la programmation structurée ou l'arbre programmation qui est son support donne un outil permettant d'écrire les trois divisions ainsi que les paragraphes de procédure. Ces résultats, correspondant aux « feuilles » du programmeur, sont obtenus séparément — l'aspect « lien logique » demeurant du ressort exclusif de l'agent, quelle que soit sa manière d'opérer. Le fait de le soulager des ordres de manipulation sur les données d'entrée ne porte en rien atteinte à ses prérogatives. En revanche, débarrassé de la programmation, il peut se consacrer entiè-*

rement au travail, plus intelligent, de construction d'algorithmes et de leur traduction en Cobol. »

C'est dans cet esprit que J.-P. Vickoff présente le recours à un outil modulaire, sorte de « super éditeur Cobol », s'adaptant à chacune des phases de la programmation. « *Dans la pratique, cela se traduirait par la mise à disposition de l'analyste d'un éditeur Cobol qui constituerait le noyau du système et qui agirait en mode conversationnel, une phase diagnostic. Il renverrait, au fur et à mesure du développement, à différents modules logiciels. Ceux-ci seraient investis de fonctions similaires particulières fournissant l'identification de l'environnement des divisions, spécifications des fichiers (select. FD), interfaces (base de données, écran, fichiers spéciaux) ; la description des structures de données en data division et l'insertion d'écritures standard (zones, dates, indices, bles, ligne d'édition rapide...).* »

L'utilitaire doit, en plus, avoir pour mission de préciser les zones de travail et de rédiger la procédure nécessaire à la résolution des problèmes cinématiques, à la gestion des ruptures de cumuls... tout en respectant la méthode de construction des programmes.

Reste la partie « spécifique ». Pour en faciliter l'écriture, J.-P. Vickoff prévoit un autre module admettant non seulement des instructions classiques Cobol mais d'autres plus core, simplifiées, qui seraient restituées sous leur forme définitive. « *Toutefois, souligne-t-il, il n'est pas recommandé d'aller trop loin dans cette voie : une dizaine de lignes d'instructions se révélant suffisantes pour assurer l'efficacité du programmeur, tout devenant des automatismes.* »

Stade ultime de la sophistication de l'outil, la mise en œuvre d'une fonction permettant de décrire un état à partir d'un écran et de récupérer immédiatement la procédure Cobol correspondante. Ce qui garantit l'hétérogénéité d'un programme et sa cohérence avec l'ensemble de la production d'un service. Il serait, d'autre part, parfaitement commenté et paginé, et aisé à maintenir, du fait de sa standardisation et de sa portabilité. Mais les gains de temps les plus appréciables se situeraient dans les « manipulations ». « *Tous les paragraphes les concernant seraient appelés automatiquement par l'instruction PERFORM.* »

Démission des constructeurs

Les limites de la préprogrammation sont évidentes. « *Il ne faut pas chercher à « débarrasser » Cobol, en poussant très loin vers le « codage parallèle » qui n'aurait en aucune façon l'universalité d'un langage évolué.* » Un responsable de centre désireux de doter son équipe de moyens d'aide à la programmation pour une meilleure productivité, voudra légitimement conserver les avantages offerts par

Tandy

Le progrès en informatique!



Ordinateur TRS-80 Modèle III

Langage BASIC Modèle III • Mémoire RAM de 32 K avec 2 minidisques
Possibilité d'extension de la mémoire jusqu'à 48 K avec 2 minidisques supplémentaires
Compatibilité avec la plupart des logiciels du Modèle I

19.950

TTC (17,60%)

Pour un prix inattendu dans cette gamme d'ordinateurs, Tandy vous propose son nouvel ordinateur monobloc: clavier à 53 touches (plus un clavier numérique à 12 touches), écran vidéo à haute définition (avec majuscules et minuscules), interface à 2 vitesses pour cassettes (500 et 1.500 Bauds) et interface pour imprimante. Le Modèle III utilise le langage BASIC Modèle III avec une mémoire RAM de 32 K, mais il est possible de porter sa capacité de mémorisation à 48 K.

26-1063

Micro-ordinateur TRS-80 de poche

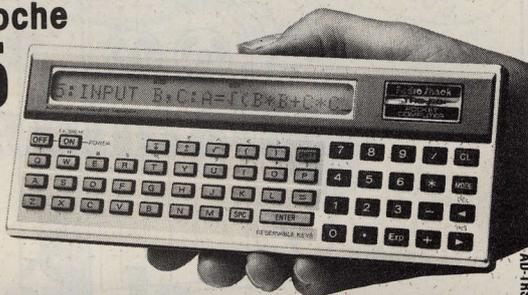
Programmes en BASIC
Mémoire RAM de 1,9 K

1.095

TTC (17,60%)

Affichage par cristaux liquides à 24 caractères. Dispositif de compacité et formulation automatique, pour utiliser au mieux la moindre partie de l'espace de la mémoire. Livré avec étui et piles «longue durée».

26-3501



AD-115-TR

Tout matériel informatique TRS-80 peut être commandé dans n'importe quel magasin Tandy

92200 NEUILLY - 23, rue du Château - tél.: 1/745.80.00 - Métro: PONT DE NEUILLY
75009 PARIS - 25, rue de la chaussée d'Antin (Galeries Lafayette) - tél.: 1/285.43.44

Métro: CHAUSSEE D'ANTIN
75015 PARIS - 26-28, avenue du Maine - tél.: 1/544.53.16
Métro: MONTPARNASSE-BIENVENUE

Cours de BASIC organisés régulièrement à:

75020 PARIS - 207, rue des Pyrénées - tél.: 1/358.27.27 - Métro: GAMBETTA
69300 CALUIRE - 12, Cours Aristide Briand (face à la Folle de Lyon - Pont de la Boucle)
tél.: 7/808.43.38

Pour connaître les adresses de nos points de vente en Belgique écrivez à:
Tandy Corporation Marketing Department - Parc industriel 5140 NANINNE

GESTION

« Le marché des logiciels n'est pas prodigue en utilitaires... »

Il faut malheureusement constater que le marché n'est pas prodigue en utilitaires. « Jusque-là, regrette J.-P. Vickoff, aucun constructeur n'a réellement voulu prendre le problème à cœur et les SSCI ne semblent pas mieux disposés ». Situation aggravée par la politique de l'« unbundling » (facturation séparée des logiciels), amorcée dans les années 70, et qui a été l'occasion, pour les fournisseurs, de baisser les bras. L'OEM a même parfois fait totalement disparaître leur responsabilité en la matière.

Aussi, ce type de logiciels continue d'être le fruit d'un travail le plus souvent artisanal et confine même au bricolage. La plupart des échecs techniques outre les erreurs commerciales en matière de prélangages, tels que les générateurs, auraient pour origine, selon J.-P. Vickoff, l'absence d'une dimension théorique à l'intérieur d'un projet. « Au départ, une société se trouve confrontée à un problème de développement d'application. Disposant d'un personnel « spécialisé », pas toujours des hommes de haut niveau d'ailleurs, le service informatique tente d'élaborer un petit outil. Il est ensuite ponctuellement sollicité par la demande des utilisateurs et l'outil acquiert, de proche en proche, d'autres fonctions jusqu'à devenir un outil « général ». En fin de parcours, et de manière tout à fait opportuniste, l'entreprise essaiera de vendre son produit, en le baptisant pompeusement « méthode » ou « progiciel », avec les lacunes que l'on sait. Un tel amateurisme, monnaie courante en programmation, ne serait toléré dans aucun autre secteur d'activité. Or seule une production « industrielle » avec les investissements correspondants peut raisonnablement répondre aux besoins très diversifiés des utilisateurs. »

En attendant l'« outil du futur » et sans aller jusqu'à souhaiter l'avènement d'informatique sans informaticiens, la préprogrammation « à la carte » reste un palliatif qui est seul en mesure de sortir les centres de développement de la crise dans laquelle se débattent des milliers d'agents. J.-P. Vickoff le croit volontiers, qui envisage avec optimisme l'avenir du Cobol. « L'association d'outils et de méthodes réfléchies permettra de revaloriser les programmeurs et de considérer leurs produits comme de vrais investissements aux yeux de l'entreprise. Je pense — et l'évolution de la norme ANS 81 semble le confirmer — que Cobol est à un tournant de sa « carrière ». Il lui faut maîtriser sa croissance quitte, pour cela, à bousculer les habitudes acquises par les utilisateurs. » M. ZAOUÏ