

La pré-programmation... ou le crayon au musée

Une tendance nouvelle se dessine : la vulgarisation d'utilitaires de type précompilateurs ; des développements intéressants dans ce sens devraient permettre de concilier l'évolution du coût d'acquisition d'un matériel avec celui du logiciel d'application.

Leur aboutissement pratique permettra d'accroître, dans de considérables proportions, la productivité de la fonction étude et concourra, à l'avenir pour une large part au choix du constructeur.

Il ne viendrait à l'esprit de personne utilisant le Cobol, de programmer le contenu des ordres de lecture ou d'écriture ; aussi, est-il difficilement compréhensible d'être astreint à coder autant de descriptions avant d'accéder à la procédure, autant d'instructions pour résoudre les éternels problèmes de cinématique de fichiers ou de gestion de ruptures (pour ne citer que les deux principaux).

Des langages paramétrés, type Gap, ont supprimé ces tâches fastidieuses, mais au prix d'une rigidité de logique convenant mal aux problèmes complexes. Les avantages offerts par des fonctions Cobol indépendantes, comme le « report writer », seraient suffisants si la difficulté de mise en œuvre n'avait ramené leur utilisation à des cas d'espèce.

D'une manière plus ponctuelle, certains générateurs ont tenté de combler les lacunes des compilateurs. La voie était bonne, mais mal balisée ; aussi, beaucoup s'y sont-ils perdus « cartes et biens ». Leur tort a été de vouloir générer un programme complet dans tous les cas de problèmes posés, alors que, seuls, certains processus généraux peuvent être systématisés. Vouloir appréhender la partie spécifique d'un traitement conduit obligatoirement à la création d'un code parallèle au langage aisément évolué, sans en avoir l'universalité.

Il en découle le plus souvent des programmes très difficilement maintenables, composés de nom-

données à rallonge, de nom-paragraphe à « numéro-tiret-indicé », remarquables par l'absence de commentaires, de pagination, par la présence d'Alter et autres astuces du



par J.P. Vickoff

genre. La plupart de ces produits, ainsi que leurs résultats, sont, en outre, d'une portabilité douteuse : par ailleurs, ils génèrent des temps de production et d'exécution à justifier les derniers progrès technologiques et des occupations mémoire à faire rêver les fanatiques de la mémoire virtuelle.

Il s'avère donc nécessaire de dégager une ligne de progiciels de pré-programmation, utilitaires simples, paliant, sans plus, les faiblesses des compilateurs actuels. Ils posséderont des fonctions diverses et suffisamment étendues pour fournir aux utilisateurs une base d'écriture, souple d'utilisation qui gèrera ce qui, dans 90% des programmes de gestion, est systématique (ruptures, cumuls, cinématique de fichiers, descriptions d'enregistrements, édition, etc.). L'outil de son

obtention devra être transparent pour le programmeur qui l'utilisera ou le maintiendra ultérieurement. Il lui laissera la faculté de réaliser lui-même, avec ou sans méthode, (avec de préférence) et avec ou sans l'aide de fonctions complémentaires la partie spécifique de chaque traitement.

Les avantages de cette relation homme-produit s'exprimeront par un regain d'intérêt du programmeur, une vitesse de réalisation accrue, tant en codage qu'en test.

L'inertie devant les principes nouveaux et l'effort nécessaire à leur assimilation n'est pas à ignorer. Il serait utile que ces logiciels permettent une auto-formation des utilisateurs, laquelle pourrait consister dans la limite du possible en une correction commentée des erreurs de spécifications après la phase de diagnostics.

Nous pourrions même imaginer en mode interactif un dialogue produit-homme de ce genre :

- Quel est votre nom ?
- celui de votre programme ?
- son objet ?
- les fichiers ?
- sur quels supports ?
- quels sont les nom-donnés sur lesquels tester ? cumuler ? effectuer les concordances ? etc.

Le logiciel assume logiquement par défaut les informations non fournies et la structure Cobol adéquate est obtenue immédiatement.

Il ne manquerait plus que l'ouverture automatique d'un dossier de programmation, cet organigramme tant controversé ou un moyen de visualisation équivalent ainsi qu'un bon éditeur de texte pour ranger définitivement son crayon au musée de l'informatique.