SYNTHESE SUR LES LANGAGES ET LES SYSTEMES D'AIDE A LA PROGRAMMATION

Cette synthèse a été rédigée par **A. Guillon** de la GIXI

dans le cadre du contrat SESORI n° 77.142

Edité par l'Institut de Recherche d'Informatique et d'Automatique
Dépôt légal 150279/500
I.S.B.N. 2-7261 - 0188 7

SOMMAIRE

DEUXIEME PARTIE : VERS LES REPRESENTATIONS MATHEMATIQUES

INTRODUCTION

S'il est une constatation éculée, c'est bien l'observation des évolutions contraires qu'ont démontré en coût logiciel et coût matériel les projets informatiques, évolutions qui ne cessent de s'accentuer et se traduisent par la part prépondérante qui revient désormais au premier dans la plupart des réalisations récentes. Il en résulte bien évidemment une actualité toujours plus pressante du problème d'aide à la programmation. Dans le contexte actuel, où la plus grande part de la facture finale émane de la construction du logiciel,tout aménagement ou amélioration de cette dernière voit un impact immédiatement sensible sur le prix de revient de l'ensemble.

En outre, la charge résultant de la maintenance peut, elle-même dans certains cas, encore outrepasser le coût de la réalisation proprement dite. Ceci alourdit à ce point les opérations que d'aucun considère cette maintenance comme le boulet majeur qui pèse sur tout projet et qui déboute bien des rêves informatiques, qui ne demandaient qu'à germer.

C'est dire que le double objectif, réaliser un logiciel de façon efficace d'une part, et le construire robuste d'autre part, constitue un des centres de préoccupation essentielle des responsables techniques dans toute activité où sévit la programmation. Telle est également, pour notre part, l'interrogation originelle qui nous a conduit à aborder le présent travail.
Néanmoins quelques précisions s'imposent quant aux perspectives qui ont présidé à son élaboration.

Selon le titre même, la principale caractéristique de ce travail est un effort de synthèse. Il faut entendre par là que l'on s'est abstenu d'approcher d'une quelconque façon l'esprit d'une "somme", d'une encyclopédie relatant aussi exhaustivement que possible les diverses expériences et techniques ayant trait à la question.

Bien au contraire, la perspective s'est attachée à dégager quelques grandes lignes et s'est traduite par une tentative d'organisation dans une compréhension synthétique globale, qui puisse au moins s'avérer de quelque secours pour le non-spécialiste.

Ceci a plusieurs conséquences immédiates. En ce qui concerne le contenu technique précis des divers points abordés, il n'a jamais paru nécessaire de relater aussi précisément que possible les expériences et recherches correspondantes. Seuls les caractères essentiels, communs à plusieurs, ou bien au contraire divergents, ont été pris en compte. A titre d'exemple, l'aspiration à un logiciel robuste, évoqué quelques lignes plus haut, voit plusieurs tendances aborder différemment la question. Certains, plus portés il est vrai à des considérations sur des logiciels très généraux, tels les systèmes d'exploitation, cherchent à développer des techniques où l'erreur est un événement quotidien, dont on cherche a priori à prendre la mesure et éviter ses trop grands dégats (confinements d'erreur, récupération, reconfiguration automatique, etc...). D'autres entrevoient la possibilité de certains logiciels certifiés, c'est à dire ayant fait l'objet de vérifications démontrant rigoureusement qu'ils ne comportent aucune faute : la preuve de compilateurs, ou tout au moins de parties de compilateurs, peut par exemple n'apparaître pas tout à fait utopique pour un avenir pas trop lointain.

Or dans aucun des deux cas de figure , la présentation détaillée des techniques correspondantes ne sera abordée. Tout au plus, une ébauche des idées ou thèmes qui y président sera évoquée.

Une autre conséquence de cette approche réside en ce que les recherches, expériences, réalisations, ou systèmes méthodologiques rencontrés sont éclatés et se voient répartis dans les différents points généraux constituant l'ossature de notre tentative de vision synthétique, lorsqu'ils en constituent des images fort représentatives. En outre, la citation dont ils font l'objet est, naturellement, à lire comme une illustration d'un thème ou d'une idée et certes pas comme une référence prétendant qualifier la valeur intrinsèque de la réalisation correspondante.

Ainsi donc, l'ensemble du sujet nous est apparu se stratifier en deux catégories plus homogènes. La première qui se développera dans la première partie contient ce que l'on peut observer de l'état de l'art actuel, ainsi que toutes extensions et recherches qui s'y greffent suffisamment naturellement pour laisser espérer une possibilité de réalisation industrielle dans un avenir pas trop lointain. Le fil de lecture qui nous est apparu le plus simple et concret consiste à rappeler dans un premier chapitre les caractéristiques et difficultés propres aux métiers de la programmation. Cette vision permet plus naturellement d'exposer, en un second chapitre les principes d'aménagements qu'il semble que l'on ait tenté d'apporter à cette réalité. Dès lors, en un troisième chapitre, on pourra envisager ce que l'automatisation informatique elle-même peut ajouter à la mise en oeuvre des moyens résultants.

Le malheur c'est qu'existe dans le monde réel un certain nombre d'anomalies aux recommandations et développements émanant de cette première partie, dont le cas d'APL constitue une illustration assez spectaculaire. Or il se trouve que ces constats d'anomalies s'estompent considérablement lorsqu'ils se trouvent relus au travers des principales idées vers lesquelles semblent curieusement converger les recherches dites de "langage de très haut niveau" et assimilés, langages de spécifications, etc...

C'est pourquoi, même si les résultats que l'on peut attendre de ce mouvement semblent nettement futuristes, néanmoins les perspectives qui les gouvernent nous sont apparues suffisamment fondamentales pour être évoquées ici, et constituer la trame de la seconde partie de ce document.