

# Présentation du système adaptable

*Comment présenter les propriétés du système adaptable ? (ou informatique « molle », comme l'appellent familièrement mes camarades).*

*Vraisemblablement la meilleure manière consiste à écrire quelques programmes « bateaux » et le comparer à ceux de l'informatique de Von Neumann (IBM), ou informatique dure en usage partout depuis la dernière guerre (1945).*

## INFORMATIQUE DURE

## INFORMATIQUE MOLLE

### MATERIEL

Caractéristiques du matériel fixées à la construction du modèle de l'ordinateur.

Caractéristiques du matériel choisies par le constructeur, l'organisation du matériel étant déterminée par l'utilisateur à l'écriture et assemblage de son programme.

### EXECUTION

Commande de l'utilisateur à travers un langage machine qui nomme les parties de celle-ci où les fonctions s'exécutent.

La machine se configure de manière à rendre le texte de l'utilisateur exécutable. Où ? A la limite le programmeur ne le sait pas. Aucun circuit ou dispositif n'est nommé.

### COMPATIBILITE

Lorsque le modèle de l'ordinateur change, le langage machine change, et le programme de l'utilisateur n'est plus exécutable en bien des cas. La durée de vie du programme est liée souvent à celle de l'ordinateur. On appelle cela le problème de la compatibilité, insoluble en informatique dure.

Lorsque le modèle de l'ordinateur change, le nouveau modèle relit le texte du programme utilisateur, de manière à adapter ses circuits à une telle exécution. L'exécution devient alors indépendante du matériel (ou de la technologie utilisée) pour n'être spécifiée que par le texte de l'utilisateur. L'informatique « molle » résout donc le problème de la compatibilité.

### MIGRATION

N'importe quel programme écrit dans le langage extensible du système adaptable, s'exécute dans n'importe quel système adaptable, indépendamment de la technologie de ce dernier, de sa capacité de mémoire (il faut quand même des ressources suffisantes) et de son constructeur. Une politique informatique cohérente peut alors être menée en France. Autrement, c'est impossible.

La conséquence pratique du mur de la compatibilité c'est la construction des micro-ordinateurs, minis, grande informatique etc. Chacun avec du logiciel particulier où toute migration est souvent impossible, indépendamment du constructeur et souvent même des langages de programmation évolués.

### CRISES

Si les programmes sont écrits pour durer « éternellement » (sauf si son

De fait, tous les 5 ans environ nous vivons une « crise » informatique

programmeur les modifie), lorsque la technologie change on remplace l'ordinateur, sans toucher au reste. L'informatique « molle » permet d'écrire des programmes en « dur » tandis que l'informatique « dure » exige des programmes « mous » à terme.

en France, correspondant à la durée de vie d'une technologie particulière. Cela peut continuer sous les socialistes.

## LANGAGES EVOLUES

Lorsque un utilisateur désire informatiser, il le fait de préférence dans un langage évolué (FORTRAN BASIC, ALGOL, APL, PL/1, ADA, COBOL, PASCAL, LISP, SIMULA, etc.).

Il existe environ 2 500 langages évolués, pas toujours faciles à apprendre (leur manuel d'utilisation fait souvent plus de 200 pages !) pas toujours disponibles (chaque ordinateur n'en comprend que 5 environ, et pas 2 500) et pire encore, il se peut que l'application ne soit pas totalement ou très difficilement exprimable dans un langage existant dans l'ordinateur choisi, ce qui conduit à écrire des « brèves » ou « verrues » en assembleur. Tout ceci fait de l'informatique une affaire de spécialistes qui ne font que programmer provisoirement, en attendant que des progrès technologiques l'obligent à recommencer le tout presque à partir de zéro.

Tous les langages évolués sont aisément traduits dans le langage extensible, avec des généralisations éventuellement (par exemple, la longueur d'une donnée en FORTRAN peut être quelconque et non pas fixée à 32, voire 16 bits). Les « brèves » ou « verrues » ajoutées s'expriment aussi dans le langage extensible qui, étant toujours compatible, assure alors la pérennité de l'application, quelle soit écrite en langage évolué ou pas. L'apprentissage du langage extensible est équivalent en difficulté à celle du FORTRAN, mais elle ne se fait qu'une seule fois, car le langage extensible permet de réaliser tous les mécanismes existants dans les langages évolués utilisés aujourd'hui.

### COMPILATION

La compilation de tous ces langages pose des problèmes aux petits constructeurs (français par exemple), car réaliser 5 langages par machine, multiplié par disons 10 machines, représente 50 compilateurs à 100 années-hommes pour chacun. Cela fait un total de 5 000 années-hommes à recommencer chaque 5 ans... Il y a du travail pour toute la France.

La compilation des langages évolués est beaucoup plus simple, car la partie « génération » peut s'adapter à chaque langage, rendant inutile l'optimisation du code. D'autre part, la génération est le plus souvent immédiate. De plus, une fois le compilateur réalisé, pour tous les systèmes, il est « éternel », la compatibilité étant assurée par le système adaptable.

### MONITEURS

Et les « moniteurs » ou système d'exploitation ? La situation est analogue à celle des compilateurs de langages évolués. Il y a du travail indéfiniment... Car, ils sont le plus souvent écrits en assembleur.

Pour nous, un « moniteur » n'est qu'une application particulière exprimée dans un langage particulier. Elle s'exprime sans difficulté dans le langage extensible et « éternel ».

### CONVERGENCE

Et la « convergence » des langages et des moniteurs ? Rechercher cette convergence revient à chercher un « Langage de tous les langages » tout comme on a cherché « l'ensemble de tous les ensembles ». Or il est vraisemblable que ce Super langage n'existe pas ; il serait contradictoire et, en conséquence inutilisable. En réalité, ce vocable cherche à « standardiser » ou « stabiliser » l'informatique pour écarter des réalisations non pré-

L'informatique « molle » est, dans ce point, exactement le contraire. Au lieu de chercher la convergence dans les moyens d'expression informatiques, nous admettons que chaque utilisateur puisse s'exprimer comme il l'entend. Nous garantissons l'exécution de ses programmes et leur pérennité. De ce fait, un utilisateur peut, à la limite, créer (assez facilement d'ailleurs) son propre langage, voire son propre moniteur. Cela

vues par les constructeurs. Je ne pense pas que l'on puisse enfermer l'informatique dans un bocal, sans enfermer aussi la pensée des informaticiens.

n'exclut pas qu'il utilise des programmes écrits par d'autres, dans la mesure où ils adoptent une convention de communication commune.

## CENTRALISATION

Par ailleurs, c'est cette convergence (au moins partielle) qui permet l'inter-connexion de presque tous les fichiers et pose les problèmes de « privacy » car l'informatique résultante, obéissant à un langage unique (à la limite), devient alors « centralisée ». L'Etat peut alors « voir » tous les citoyens s'il le désire, l'inverse devenant souvent impossible, car seuls les spécialistes attirés ont accès au Super langage.

Si à la limite, chacun peut créer sa propre informatique, si complexe soit elle, et l'isoler des autres, même si elle s'exécute dans un ordinateur commun, l'informatique devient « privée » et « secrète ». (Si l'utilisateur le désire). L'accès est « naturellement » interdit aux autres. Si tout le monde fait cela, il en résultera une informatique totalement décentralisée. On peut cependant admettre que, au niveau de l'Etat, un langage commun adopté par tous puisse être défini. Toute l'astuce du langage extensible consiste à fixer une syntaxe (très simple) et permettre à l'utilisateur de construire naturellement sa sémantique.

## PERFORMANCES ECONOMIQUES

L'ordinateur Von Neumann est plus simple (20 %) à construire, cher à programmer. En outre, le prix de la programmation (au niveau de la France) ne cesse pas d'augmenter. Le logiciel devient alors une véritable boule de neige sans arrêt prévisible !

Le système adaptable utilise d'avantage des circuits et légèrement moins de mémoires que l'informatique dure. Le temps d'exécution est légèrement supérieur (10 % en moyenne ?). Cependant, cela dépend de l'application. Celles qui sont très simples s'exécutent mieux dans l'informatique dure, les autres sont à l'avantage du système adaptable. Par contre, le prix de la programmation ne cesse de diminuer avec le temps, car les programmes durent.

## DIFFICULTE DE LA PROGRAMMATION

Distinguons deux problèmes souvent mélangés. Pour informatiser, nous devons au préalable savoir résoudre le problème posé. Par exemple comment inverser une matrice. Ensuite, nous expliquons à la machine comment le faire, à travers un langage (évolué ?). Si le langage convient, cette deuxième opération est simple en général. Malheureusement, tout l'art de la programmation consiste à contourner les limites (dures) de la machine ou du langage. Par exemple, comment accéder aux bits, comme extraire des triples-mots, comment ne pas dépasser les bornes d'adressage, (capacité de mémoire maximale) comment réaliser une fonction non prévue (récur-sive, par exemple) ou comment découper le programme en moins de 256 modules etc...). Ce sont ces problèmes qui font de l'informatique une affaire de spécialistes.

En ce qui concerne la solution du problème à informatiser, la difficulté est la même. Ainsi, piloter automatiquement un Airbus est l'affaire du pilote spécialiste. Par contre, au moment de programmer le système adaptable, celui-ci apparaît à son utilisateur comme n'ayant pas de limites. Ainsi, il peut extraire les bits ou triples-mots ! Il peut adresser autant de mémoires qu'il souhaite. Ces programmes sont naturellement récur-sifs et réentrants. Il peut créer autant de nouvelles fonctions qu'il désire et découper le programme en autant de modules que nécessaire. Il peut spécifier une quantité illimitée de paramètres dans ces procédures et les appeler en cascade sans limite. Bref, sa programmation se limite à décrire son algorithme, sous forme d'un programme structuré, modulaire et aisément modifiable. La structure de données est celle de l'Algol 60.

## TRANSITION DURE - MOLLE

Lorsqu'un constructeur change de matériel, il le faut, si possible, exécuter les programmes existants de ses clients. Si ces derniers sont écrits en « langage évolué normalisé » et sans « verrous », pas de problèmes. Si non, il risque de perdre ses clients, à moins que son nouveau matériel n'émule l'ancien. On crée alors des « modes » câblés. Pour combien de temps ?

Les programmes existant en « langage évolué normalisé » sont re-compilés sans problèmes dans le langage extensible. **Nous postulons que, à long terme, le langage extensible remplacera tous les assembleurs.** Si la perte de performance est trop grande (70 %), lorsque nous interprétons les assembleurs en langage extensible, il est préférable alors de créer des « modes » câblés coexistant avec le système adaptable. La situation n'est pas pire qu'actuellement.

## QUI PEUT PROGRAMMER LES APPLICATIONS ?

Dans l'informatique dure, nous pouvons affirmer que cela ne peut être, à quelques exceptions près, que l'affaire des informaticiens, ce qui rend l'informatique ésotérique et incompréhensible par la population. Peut-on dire que dans ces conditions l'autogestion peut exister un jour ? Car, même si la population souhaite modifier une information anti-sociale qui ne lui convient pas, il faut encore que les spécialistes arrivent à le faire. C'est très souvent trop complexe ou trop cher. La population devient alors prisonnière des choix du passé, sorte de corset dur dans lequel toute évolution est interdite, conduisant la société entière à sa fîgure. L'informatique dure contribue donc à réaliser une société aussi « dure ».

Certes, les applications complexes seront toujours l'affaire des spécialistes. Mais les simples, c'est-à-dire 90 % des applications quotidiennes, nous ne voyons pas pourquoi les secrétaires, les ménagères, les étudiants, les ouvriers, les médecins... etc. ne pourraient pas programmer leur vie quotidienne, au lieu que d'autres la leur programment. Or, nous pensons que à partir de 14 ans, avec une courte formation, la population pourrait commander elle-même n'importe quel système adaptable, du mini au méga, soit en écrivant elle-même ses programmes, (avec des mots clés en français ou dans une autre langue), soit en enchaînant et combinant ceux des autres écrits avec des conventions communes. La population, en maîtrisant l'ordinateur, pourrait maîtriser individuellement ou collectivement tous les outils, machines ou appareils (au travail, chez soi ou ailleurs) donnant une signification effective au mot **autogestion**. Et l'on verrait alors le socialisme « mou » et extensible à l'œuvre !

M.C.B.

## Récapitulation

### Informatique dure

Centralisée  
Programation par des informaticiens  
Incompatible  
Standardisée  
Applications en langages évolués plus l'assembleur  
Compilation difficile  
Limitative  
Souvent inadaptée  
Travail « stérile » pour les informatisés  
En anglais (américain)

### Informatique molle

Décentralisée  
programmation par tous  
Compatible  
Personnalisée  
Applications en langages privés ou en langage évolués plus en langage extensible  
compilation facile  
Extensible  
Adaptable  
Travail « intelligent » pour tous  
En toutes langues