

PRO-SYSTEMES

DECEMBRE 1987 - N° 81

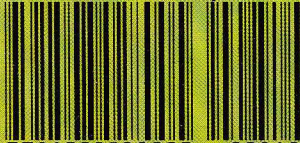
DOSSIER: **VEGETATION ASSISTEE PAR ORDINATEUR**

ATARI: **LA GALAXIE P.A.O.**

THEME DU MOIS: **LES ROBOTS SONT PARMIS NOUS!**



T 1508 - 81 - 28,00 F



3791508028005 00810

PROGRAMMEURS

- ▶ Database Toolbox: pour le développement d'applications de base de données.
- ▶ Editor Toolbox: pour construire votre propre traitement de texte ou incorporer un éditeur dans vos applications.
- ▶ Graphix Toolbox: pour construire des graphiques en haute résolution.
- ▶ Gameworks*: pour apprendre la théorie des jeux et créer votre propre logiciel ludique.



NOUVEAU! POUR LES SCIENTIFIQUES ET LES INGÉNIEURS: MÉTHODES NUMÉRIQUES* POUR TURBO PASCAL

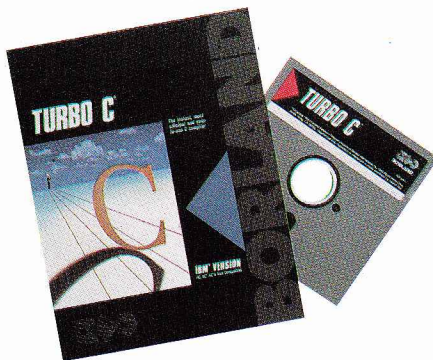
Un ensemble très complet de routines et de programmes pour doter vos applications de puissants outils d'analyse. Il comprend: résolution d'équations, interpolations, calculs de dérivées, calculs d'intégrales, inversions de matrices, équations, moindres carrés, graphiques et transformations de Fourier
Prix: 995 F.H.T.



TURBO C. LA RÉFÉRENCE DES PROFESSIONNELS.

Les programmeurs confirmés en C apprécient son étonnante vitesse, la qualité du code généré et l'adoptent sans hésitation. Les débutants, pour leur part, aiment sa facilité d'apprentissage et d'utilisation. En compilant plus de 10 000 lignes à la minute, Turbo C s'est immédiatement imposé. Résultat: 100 000 exemplaires vendus en trois mois.

Turbo C* 1295 F.H.T.



LE FORUM DES LANGAGES

- ▶ Sur 36 14 code Borland tapez FOR, "le Forum des langages" est ouvert sur abonnement à tous ceux qui souhaitent bénéficier de conseils, dialoguer, participer en direct à la vie de la programmation. Faites 36 14, code BORLAND, tapez BOR, vous saurez tout, tout de suite, sur Borland sans abonnement préalable.

PRINCIPALES CARACTÉRISTIQUES

- ▶ Compilateur: compile en une passe en générant du code natif, des modules objets ou des fichiers source assembleur. Le format des fichiers objet est compatible avec l'éditeur de liens PC DOS. Six modèles de mémoire mixables: tiny, small, medium, compact, large, huge. (Utilise le 8087/80287 si celui-ci est installé.)
- ▶ Éditeur interactif: le système comprend un puissant éditeur plein écran. Si le compilateur détecte une erreur, l'éditeur positionne le curseur automatiquement sur celle-ci dans le code source.
- ▶ Environnement de développement: une fonction Réalisation/Projet (Make) est incluse qui rend le développement en C particulièrement facile. Gestion des fenêtres et des menus déroulants.
- ▶ Edition de liens avec des modules objets relogeables créés par Turbo Prolog.
- ▶ Compatible avec le standard ANSI du C.
- ▶ Environnement intégré ou en lignes de commandes.
- ▶ Source de bibliothèques Runtime bientôt disponible.

"Turbo C reprend les concepts développés dans Turbo Pascal: une très grande rapidité de compilation et d'exécution, un environnement de développement particulièrement convivial..."

G. Sebarine.

Décision informatique. Fév. 87.

Tous les produits BORLAND sont des marques déposées de BORLAND INTERNATIONAL INC. Les autres marques ou noms de produits sont des marques déposées par leurs propriétaires respectifs.

Copyright Borland International 1987.



65, rue de la Garenne - Dépt. Z - 9
92318 SEVRES Cedex - France
Tél.: (33) (1) 45 07 15 11 - Téléc.: 632 162 F

S O M M A I R E

MICRODIGEST

Toute l'actualité du monde micro-informatique : les nouveaux matériels et logiciels, les livres, les rendez-vous de l'informatique..... 25

SOCIETE ET SOCIETES

Big brother nous regarde-t-il déjà ?..... 73

ESSAIS

- Un PC bon marché et performant : Vicki..... 79
- Télé PC 1200 : une nouvelle façon de communiquer..... 83
- Draw : une souris qui dessine..... 87
- La vision A4 avec l'écran Génius..... 91
- Un environnement SGDB complet pour développeurs : DOS ISAM..... 93

ANALYSE

Atari : la galaxie PAO..... 99

DOSSIERS

- Végétation assistée par ordinateur..... 112
- Les PGAs de la série 3000 de Xilinx..... 126

THEME DU MOIS

- La robotique : un mythe décisif..... 137
- Robotisation et PMI : le choix de la compétitivité..... 140
- Robotique et futur : les robots nouveaux sont arrivés..... 152
- Robotique musicale, la triple fonction : I.A., robotique, industrie..... 160

FICHE COMPOSANTS

- Fiche 48 : le SCC 68070 de RTC..... 167

DEVELOPPEMENT

La programmation sans panne (4) : 100 milliards d'instructions..... 175

INTELLIGENCE ARTIFICIELLE

- C++ : le nouveau langage objet..... 185
- Conception d'un système expert (4)..... 201

LEGISLATION

Les limites de protection de programmes par droit d'auteur : les clones de progiciels..... 211

ET AUSSI...

Cote de l'occasion..... 222
 Petites annonces..... 223
 Le bonus de Micro-Systèmes..... 228
 Index des annonceurs..... 238

POUR LES TIMB



200 000 IMPRIMANTES VENDUES EN 1986*

CHOISISSEZ VOTRE SEIKOSHA

Si Seikosha a acquis la deuxième place en Europe, ce n'est pas par hasard. Cette réussite, Seikosha la doit à la qualité et à la fiabilité de ses imprimantes, qui allient silence et esthétique. Et si Tekelec Airtronic a choisi de représenter Seikosha en France, c'est aussi pour l'étendue de la gamme (plus de vingt modèles), la multicompatibilité, et l'excellent rapport performances/prix.

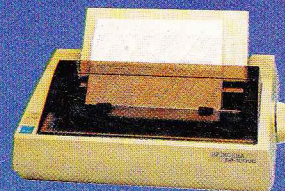
**TEKELEC AIRTRONIC ET SEIKOSHA : LA BONNE VOIE
POUR VOTRE INFORMATIQUE**

*Ventes en Europe (Source IDC)

GAYAL

TEKELEC TA AIRTRONIC

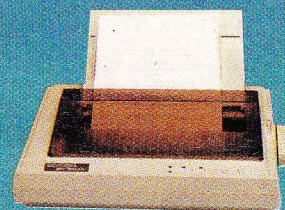
TEKELEC AIRTRONIC - rue Carle Vernet - B.P. 2 - 92315 SÈVRES CEDEX
TÉL. : (16) 1.45.34.75.35 - TELEX : TKLEC 204 552 F - FAX 45.07.21.91.



SP180 : 80 col - 100 c/s



SL80 : 80 col - 135 c/s - 24 aiguilles



SP1200 : 80 col - 120 c/s



BP1300/5300 AI : 80/136 col - 300 c/s



BP5420 : 136 col - 420 c/s



SBP10 : 136 col - A4 - 800 c/s



VEGETATION ASSISTEE PAR ORDINATEUR

Des palmiers poussent sur l'écran de votre ordinateur, un minitel se charge d'arroser votre jardin en votre absence, un robot tond votre pelouse, un autre taille les arbres ou cueille des fruits... le tout sous la conduite vigilante d'un système expert. Vision futuriste ? Pas tellement. Car, quand le monde de l'informatique rencontre celui des plantes, il apparaît une nouvelle technologie, où botanistes et physiologistes des plantes se retrouvent aux côtés d'électroniciens et d'informaticiens : c'est la « végétation assistée par ordinateur ».

La végétation, symbole de la nature vivante et éternelle, paraît être à l'antipode de la froide technologie moderne que représente l'informatique. Née et développée dans les zones fortement urbanisées et industrialisées, comme les mégapoles japonaises ou la californienne Silicon Valley, et confinée, à ses débuts, dans les calculs astronomiques pour les grands centres de recherches et universités, ou dans la gestion de très grandes entreprises, l'informatique investirait-elle désormais les champs, les forêts, les jardins ? Comment les plantes, dans leur diversité, pourraient-elles se prêter au traitement automatique, et les arbres, les fleurs, les feuilles se réduire à des séquences de zéros et de uns ?

Et pourtant, la branche et l'arbre ont fourni plus d'un modèle aux informaticiens. Les programmes les plus élémentaires comprennent des « embranchements » et « ramifications » ; les « arbres binaires » et « arbres de décision » servent à modéliser le raisonnement, les réseaux sémantiques utilisés pour représenter les connaissances dans les systèmes experts ont des formes arborescentes, d'autres réseaux « arborescents » relient les terminaux au centre de calcul... sans oublier le slogan publicitaire parodiant Saint-Exupéry : « Bull, dessine-moi un arbre ». Enfin, est-ce un hasard si

nos cellules nerveuses elles-mêmes arborent cette forme et si les terminaisons neuronales ont pour nom « dendrites » (du grec « dendron » : arbre) ?

Après les petites entreprises et les professions libérales, nombre d'agriculteurs ont vaincu les réticences que leur inspirait cet instrument d'un autre monde, puisqu'ils sont déjà quelque dix mille en France à être informatisés. L'ordinateur voisine aujourd'hui avec le tracteur, le tuyau d'arrosage et la tondeuse à gazon, tandis que des botanistes, associés à des informaticiens, font croître des palmiers, peupliers et autres conifères sur des écrans cathodiques, pour le plus grand profit des architectes, paysagistes, urbanistes, étudiants en botanique ou en horticulture, et pour la plus grande joie des créateurs d'images de synthèse.

Des mathématiques et des plantes

La modélisation ou la simulation numérique par ordinateur permet, de nos jours, de calculer, de reproduire et d'observer pratiquement tous les phénomènes naturels, depuis les interactions moléculaires jusqu'à la formation d'une galaxie, et ce grâce aux techniques et aux performances inégalées des processeurs dédiés au traite-

ment d'images et aux supercalculateurs à architecture parallèle (voir notamment *Micro-Systemes* n° 78, p. 158, septembre 1987).

Ces techniques reposent sur l'association d'une résolution numérique – impliquant un grand nombre de calculs – et d'une visualisation – nécessitant des processeurs d'images et des écrans à haute définition. Le système que l'on veut représenter est décrit par un modèle mathématique défini par un certain nombre de paramètres correspondant aux caractéristiques du système en question.

Dès les années cinquante, John von Neumann et Stanislas Ulam avaient inventé une méthode numérique pour modéliser les processus d'auto-organisation dans les systèmes biologiques, d'où leur nom d'« automates cellulaires » ou « jeu de la vie ». Cette méthode consiste à appliquer un maillage (réseau) sur le système étudié. A chaque nœud est associé un état discret qui ne peut prendre qu'un petit nombre de valeurs. La valeur d'un nœud dépend de son état antérieur et de celui de ses voisins ; inversement, elle influe sur l'état de ceux-ci.

Les plantes étant un exemple de système biologique, c'est-à-dire un automate cellulaire extrêmement complexe, il était possible, dès lors que l'on possédait des moyens de visualisation suffisamment élaborés, de

DOSSIER

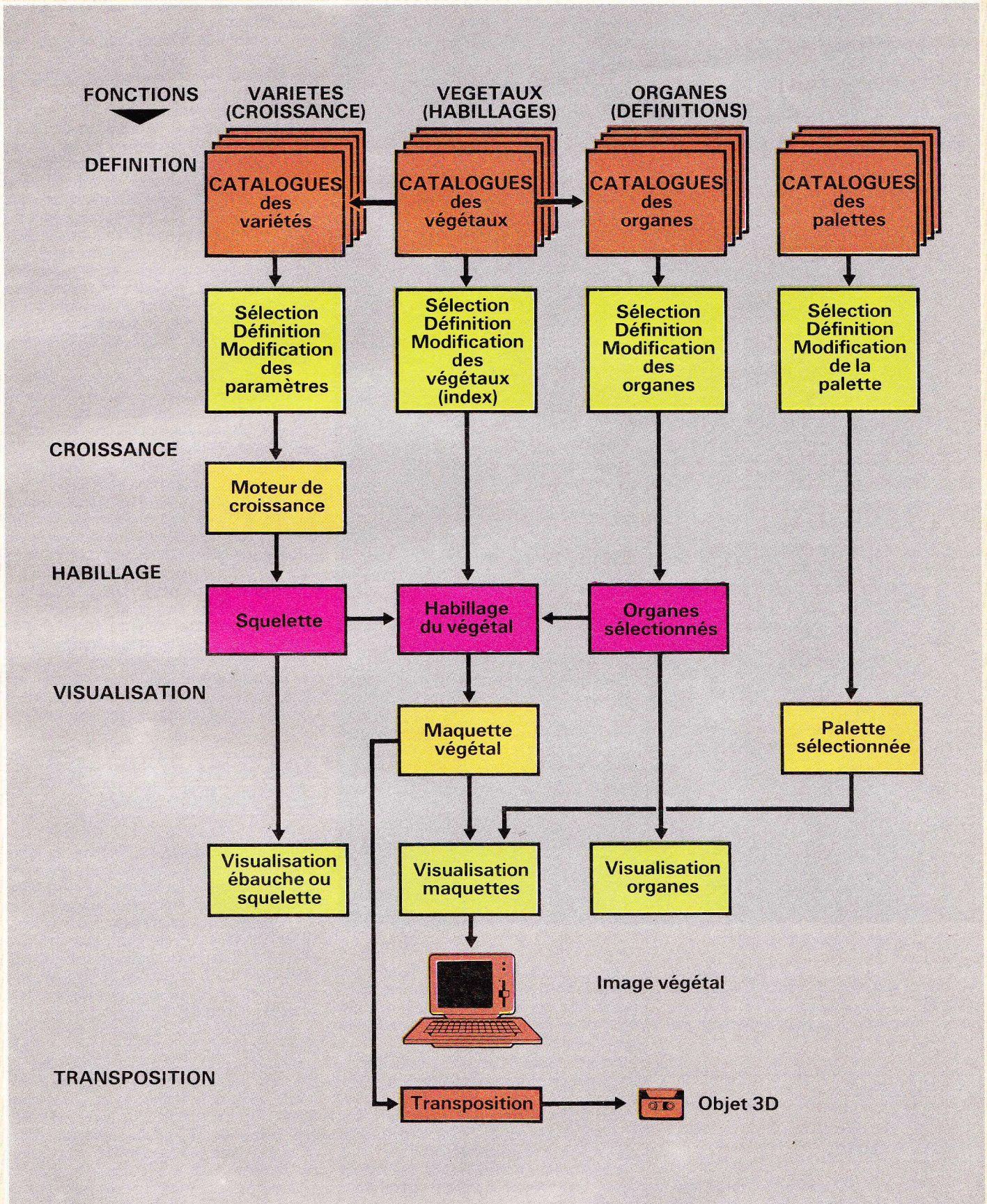


Fig. 3. - Architecture générale d'AMAP. (D'après doc. SESA.)



(c) S&S&A

Arbre taillé généré par le logiciel AMAP.

Louis Pasteur (3), « la modélisation doit se baser sur le maximum de connaissances en botanique, (...) une forme naturelle se comprend par sa croissance, et plus généralement par son évolution. »

Modéliser les lois botaniques

En effet, à la différence des autres méthodes connues de calcul d'images de végétaux, issues de procédés purement mathé-

matiques telles les fractales (que nous venons de voir), AMAP est fondé sur l'intégration fine des fonctions biologiques des végétaux, ce qui lui confère les caractéristiques suivantes :

- la polyvalence, le modèle pouvant s'appliquer en principe à n'importe quel végétal ;
- une fidélité botanique exceptionnelle, les images obtenues étant strictement conformes à une réalité possible ;
- la prise en compte du temps, le progiciel pouvant représenter le même végétal à des âges différents ;
- la prise en compte de l'aléatoire de la

vie, deux plantes ainsi modélisées n'étant jamais identiques ;

- la possibilité d'interaction avec un certain nombre de paramètres (présence du vent, saison, densité de feuillage, taille d'un arbre à des âges déterminés, etc.) ;
- la qualité graphique.

Les observations de Philippe de Reffye sur une centaine de caféiers lui ont permis de calculer les probabilités de ramification, de mortalité ou de pause, et d'établir ainsi le premier modèle mathématique de croissance d'un arbre (4). Mais auparavant, il a fallu recenser sur le terrain un grand nombre de paramètres (environ soixante-quinze pour chaque plante, tous n'étant pas utilisés simultanément). A partir de ces données, les chercheurs calculent l'influence, sur la structure de l'arbre, de facteurs tels que la pente du terrain, la densité de la végétation, l'attaque des insectes...

Appliqué en Côte d'Ivoire à des cultures de caféiers, ce modèle a permis d'étudier la verse (phénomène qui fait pencher ou tomber les arbres), ainsi que d'autres facteurs utiles, tels que l'influence de différentes doses d'engrais sur la croissance des plantes ou la résistance à diverses maladies.

Après le caféier, c'est au tour du palmier, du cotonnier, de l'arbre à lychees, de l'hévéa... d'être modélisés. Chaque année, le programme est étendu à de nouvelles variétés.

De plus, pour chaque variété, l'intervention de nombres aléatoires, simulant le hasard dans la nature, fait que des clones d'un même arbre ne croissent jamais de la même façon, même s'ils présentent un air de ressemblance. Une forêt entière peut ainsi être synthétisée, sans qu'aucun arbre ne soit exactement identique à son voisin.

Une nouvelle branche de l'architecture

P. de Reffye n'est certes pas seul à travailler dans ce domaine ; il existe, en effet, d'autres logiciels de modélisation des plantes, notamment celui réalisé par une équipe de l'université de Cornell (Californie) sous la direction du professeur Karl Niklas. Mais AMAP a ceci de particulier qu'il est très général et peut s'appliquer à n'importe quel végétal. « Il est possible de calculer et de représenter une plante aussi loin qu'on en a le courage », affirme P. de Reffye.

En effet, AMAP se fonde essentiellement sur la notion d'architecture des plantes, qui s'applique à tout le règne végétal. Ce n'est que très récemment que cette notion a été étudiée d'un point de vue scientifique. En examinant attentivement leur croissance, des botanistes ont découvert qu'une grande variété d'arbres et de plantes évoluent de la même façon, bien qu'ils diffèrent très nettement par leur apparence. Le professeur Francis Hallé (Laboratoire de botanique tropicale à Montpel-

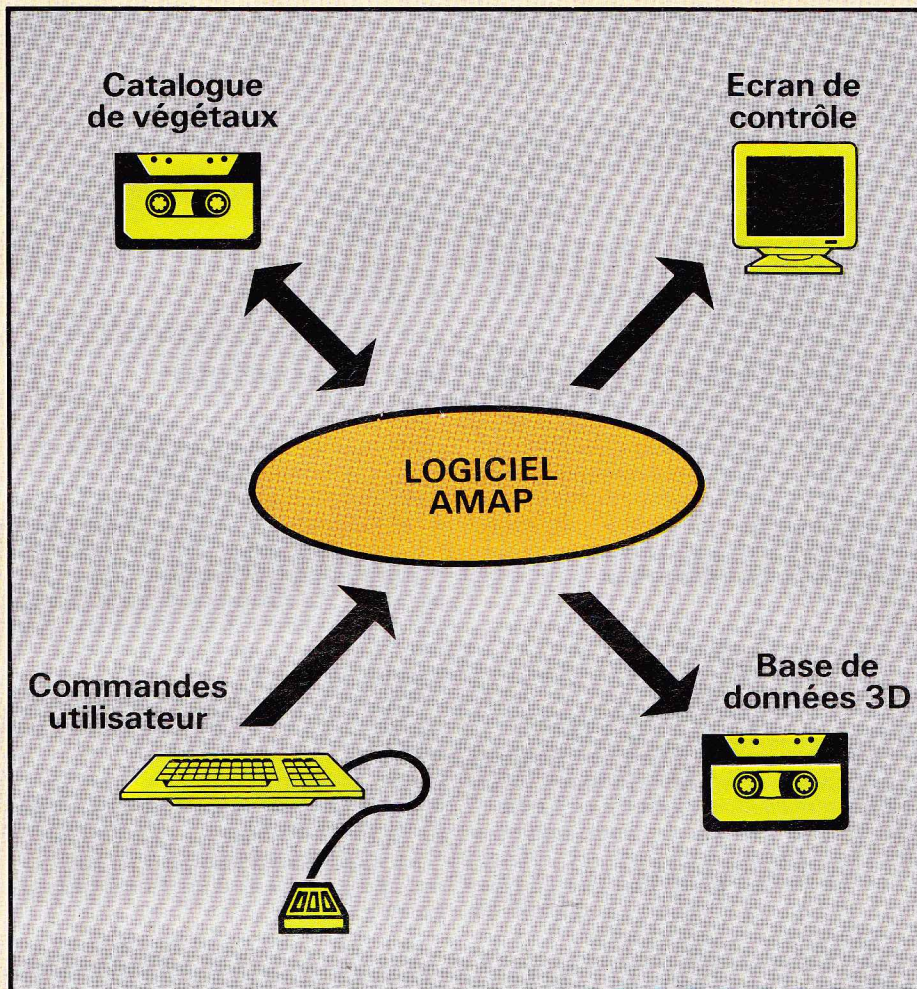


Fig. 4. — Architecture externe d'AMAP. (D'après doc. SESA.)

lier) a ainsi mis en évidence « une vingtaine de modèles architecturaux pour répertorier les quelque trois cent mille espèces d'arbres connues ».

Chaque modèle architectural est caractérisé à l'aide d'une douzaine de paramètres. Le plus simple est le modèle du palmier, comportant un tronc, pas de branches et de grandes feuilles, tandis que le pin maritime, au tronc droit et haut, aux branches dressées et disposées en étages, fournit un modèle architectural pour de très nombreux végétaux des pays tropicaux et tempérés. En fait, trois modèles seulement — assez bien représentés par le peuplier, le sapin et le hêtre — suffisent pour décrire toute la végétation de l'Europe, alors que la végétation tropicale offre un éventail beaucoup plus large. « Sans ce travail préalable des botanistes, je n'aurais rien pu faire », reconnaît P. de Reffye.

Le moteur de croissance

Par ailleurs, la croissance d'une plante est le résultat de l'évolution de ses tissus

cellulaires spécifiques, qui donnent naissance aux organes : tronc, branche, entrenœud, feuille, fleur, fruit. Cette croissance dépend essentiellement du temps. C'est pourquoi, pour modéliser les plantes, il a été nécessaire de créer un modèle de simulation de croissance, basé sur les lois générales de la biologie et appelé « moteur de croissance ».

Après initialisation, ce moteur permet de générer une plante en prenant en compte simultanément la progression dans l'architecture de la plante et la construction des organes. La progression dans l'arbre s'effectue en s'aidant des lois botaniques. Les phénomènes de la vie d'une plante (élongation, mortalité, pause, réitération, élagage) sont pris en compte dans le modèle grâce à des probabilités de succès sur chacun de ces événements. Le moteur de croissance effectue un tirage aléatoire et compare le nombre tiré avec la probabilité correspondant au phénomène testé. Le temps nécessaire à chaque étape de croissance peut être rendu dépendant de phénomènes extérieurs (sécheresse, engrais, alternance des saisons, etc.).

Simultanément, à chaque fois qu'une

branche s'accroît, la disposition dans l'espace est immédiatement déterminée, en tenant compte des paramètres géométriques (longueurs, angles) et de l'influence des éléments extérieurs (pesanteur, vent, pente du terrain...) grâce à des calculs de résistance des matériaux. C'est à ce niveau qu'il est fait appel aux procédures simulant la croissance des feuilles, fleurs, fruits.

Dessine-moi un arbre

Pour « créer » un végétal, l'utilisateur d'AMAP procède en deux étapes : il faut d'abord construire son architecture, consistant à définir une « variété » caractérisée par le tronc, les branches... puis « habiller » ce squelette végétal à l'aide de fleurs, de feuilles, de fruits, autrement dit les « organes » (fig. 3).

Les différentes variétés et les organes, qui définissent une plante, sont disponibles dans un catalogue de végétaux dans lequel l'utilisateur peut choisir le spécimen correspondant totalement ou partiellement à son besoin. Une fois choisie, la maquette 3D générée par AMAP se présente sous forme de fichier ASCII. Le végétal peut, dès lors, être visualisé, modifié, archivé, intégré dans une base de données externe (fig. 4).

Des bibliothèques de formes sont livrées avec le logiciel commercialisé par la SESA (Société d'études des systèmes d'automatisation) pour mini ou micro-ordinateur (PC/AT ou compatible). A titre indicatif, sur un Data General MV 10000 SX, la simulation demande environ une minute pour un palmier à huile, et une dizaine de minutes pour un peuplier ; bien sûr, le temps de calcul dépend de la complexité du végétal et du degré de précision demandé.

De l'arbre à la feuille

AMAP a évolué non seulement en s'appliquant à des variétés de plus en plus nombreuses, mais aussi en étant capable de calculer et dessiner des détails toujours plus précis. Les feuilles, par exemple, au départ simples taches vertes, révèlent maintenant toutes leurs caractéristiques de forme et de texture, lesquelles évoluent à mesure que l'arbre croît.

La construction d'un organe se fait suivant les mêmes principes que ceux de l'arbre entier. Le moteur de croissance de ces organes a en effet la même structure que celui de l'arbre, mais avec un nombre de paramètres restreint. C'est pourquoi le modèle est aussi bien utilisable pour la forme générale de la plante que pour des vues détaillées par agrandissement d'une partie de la plante (c'est d'ailleurs ce qui a conduit à appliquer le modèle fractal aux plantes, comme nous l'avons vu précédemment).

C'est ainsi que P. Lienhardt et J. Françon se sont appliqués à modéliser les feuilles et l'évolution de celles-ci : croissance, déformations, nécrose et autres métamorphoses, par exemple les variations de coloration d'une feuille au cours des saisons, le dépliement de celle-ci au printemps ou son froissement en automne, variations qui correspondent à des modifications de la composition même de la feuille.

La méthode proposée par ces chercheurs s'appuie sur les notions botaniques relatives à la croissance et à l'évolution d'un système de nervures, ainsi qu'à celle du limbe, c'est-à-dire la surface s'appuyant sur ces nervures qui forment, en première approximation, un système analogue à un arbre bidimensionnel.

Ainsi, l'évolution dans le temps de la feuille peut être suivie sur ordinateur : initialement ($t=0$), la feuille est réduite à un point. A chaque instant ultérieur ($t>0$), le modèle instantané de la feuille est déduit de son modèle à l'instant précédent ($t-1$). Un certain nombre d'opérations sont possibles entre les instants t et $t+1$. Ce sont : la ramification (création d'une nouvelle arête de croissance), la fusion, la disjonction (création ou suppression de faces frontières, permettant une modification du contour de la feuille).

Cette simulation a été réalisée sur mini-ordinateur SM90. Le nombre de faces triangulaires composant les feuilles varie entre 300 et 3 000. Le temps de calcul nécessaire à la génération de ces images est compris entre 3 secondes et 4 minutes, sans tenir compte du temps de remplissage des facettes. La méthode s'est avérée applicable aussi bien aux images de fleurs, fruits et autres surfaces.

Représenter un champ de maïs

Le même P. de Reffye, associé à des chercheurs de l'INRA (Institut national de la recherche agronomique), a développé un modèle géométrique pour le maïs, afin de caractériser correctement la structure complète du couvert végétal (5). Partant du modèle mis en œuvre par AMAP, cette équipe s'est attachée à décrire très exactement le maïs, plante particulièrement simple, décomposée en une tige, 12 à 15 feuilles alternées, un épi, une panicule (fleur mâle) à l'extrémité de la tige. La représentation d'un pied met en jeu une centaine de paramètres. Pour passer de l'individu au peuplement, il suffit de calculer, puis de juxtaposer le nombre de pieds voulus. Cette implantation se fait en fonction des caractéristiques du peuplement désirées.

La simulation sur ordinateur permet de résoudre deux sortes de problèmes :
– biologique, d'une part : il s'agit de la compétition entre pieds pour le partage des ressources ; ce phénomène peut être modé-

lisé au travers de fonctions d'autocorrélations intra-rang, la compétition inter-rang étant négligée ;

– géométrique, d'autre part : c'est le problème, particulièrement complexe, de l'interpénétration du feuillage, où le déplacement d'une feuille peut engendrer de nouvelles collisions.

Une telle maquette est déjà disponible pour un petit peuplement, de quelques dizaines à quelques centaines de pieds. Il est envisagé d'intégrer ce modèle dans de nombreuses études, tels le calcul des ombres portées à l'intérieur du couvert végétal, la caractérisation des échanges radiatifs au sein de la végétation, et ultérieurement la modélisation spatiale de la propagation d'épidémies (déplacements d'insectes, diffusion sur un support complexe) ou la modélisation du microclimat au travers d'une meilleure prise en compte des échanges de vapeur d'eau et de gaz carbonique entre le couvert végétal et l'atmosphère.

Les fractales ou la logique des plantes

« En imitant la logique de la nature, les logiciels de simulations informatiques permettent de cerner l'essence des objets naturels », explique Peter Oppenheimer (6). L'originalité des travaux de ce chercheur du New York Institute of Technology, disciple de B. Mandelbrot, est de concilier, dans un programme de modélisation, la géométrie fractale avec les lois naturelles. « Au-delà de la simple modélisation visuelle, ce programme simule le processus de croissance de ces formes en imitant la logique génétique de ces organismes », ajoute-t-il.

Les paramètres du programme de modélisation d'arbre à l'aide des fractales (fig. 1 et 2) joueraient, selon ce chercheur, un rôle analogue au code ADN (acide désoxyribonucléique) qui décrit les caractéristiques des branches d'un arbre. De même que l'ADN contient des informations sur tout l'organisme sous une forme très économique, ces paramètres, peu nombreux, suffisent à déterminer entièrement la forme et l'allure de l'arbre. « Les ordinateurs, comme les organismes, appliquent des règles simples pour créer des structures complexes », constate P. Oppenheimer.

L'analogie avec l'ADN est encore plus flagrante, lorsque l'on sait que la forme de cette molécule géante est celle d'une spirale, qui serait, selon P. Oppenheimer, « en quelque sorte un ensemble autosimilaire dégénéré. C'est l'unité atomique qui constitue les arbres fractals ».

En fait, l'utilisation des fractales permet plutôt de créer des formes nouvelles, une botanique fantastique, une végétation surréaliste, que de modéliser des plantes existant dans la nature. Il est possible de partir

d'une plante réaliste et, en modifiant un à un les paramètres, d'assister à une métamorphose complète de celle-ci. P. Oppenheimer a ainsi réalisé une animation où, par interpolation de ces paramètres d'une espèce d'arbre réel à une autre, il effectue des transformations étonnantes comme, par exemple, la métamorphose d'un sapin en cerisier.

L'avantage de l'expérimentation sur ordinateur

La modélisation sur ordinateur ne remplace pas l'expérimentation en laboratoire, mais lui est complémentaire : « L'intérêt du numérique est d'ouvrir de nouveaux champs d'expérimentation », explique Jean-François Colonna, qui dirige le GSV-Lactamme (Groupe Synthèse Vidéo) à l'Ecole Polytechnique. La simulation permet, en particulier, d'accélérer le cours du temps : visualiser un jardin ou une forêt cinq ou dix ans après la plantation des arbres offre un avantage incomparable à un architecte paysagiste comme à un sylviculteur, et savoir, comme le fait remarquer P. de Reffye, « si la branche du cèdre planté dans votre jardin traversera un jour la salle de bains » n'est pas négligeable...

La modélisation ouvre encore bien d'autres perspectives, dont voici quelques exemples.

– Sélection : mise en œuvre de facteurs limitant la production de caféiers, liés à l'architecture des plants ; élimination précoce, par extrapolation, de nouveaux croisements d'hévéas susceptibles de casser au vent.

– Etude de nouvelles pratiques de plantation (gène entre plantes), de taille ou de bouturage ; l'INRA espère ainsi pouvoir optimiser la conduite d'exploitations forestières pour accroître le volume utile de bois.

– Optimisation de l'utilisation des insecticides et des engrais, par mise en corrélation de la croissance de la population avec l'évolution de la surface foliaire.

– Création de plantes « mutantes » et autres végétaux fantastiques n'existant pas dans la nature, mais ayant une apparence de réalité.

Le laboratoire de modélisation du CI-RAD à Montpellier assure les développements liés à la création de nouvelles variétés de modèles, le calcul d'éléments végétatifs (feuilles, fleurs, fruits), le développement de la banque de formes pour ces éléments... et poursuit des recherches sur l'utilisation scientifique du modèle.

En outre, précise P. Oppenheimer, « l'avantage économique d'un programme est qu'une structure hautement complexe est générée à partir d'un ensemble très concis de données qu'il est facile de produire. C'est d'ail-



© Sesa/Cirad

Cerisier généré par le logiciel AMAP à partir des paramètres de croissance.

leurs ainsi, suggère-t-il aussi, que la nature a résolu le problème de l'évolution et du conflit entre complexité et simplicité. »

L'intelligence vient aux plantes

L'apport de l'intelligence artificielle est particulièrement important dans les phénomènes relatifs à la végétation, étant donné leur complexité. Elle intervient notamment dans la synthèse d'images végétales, où beaucoup de paramètres entrent en jeu, comme la taille d'une plante, la proximité avec d'autres végétaux, la pente du terrain, la nature du sol, etc.

Les techniques d'intelligence artificielle font passer à l'arrière-plan la question de la géométrie visuelle des objets et de la modélisation de leur allure physique dans l'espace, pour s'attacher d'abord à leur représentation symbolique. Les connaissances introduites dans la machine sont des « notions symboliques » de ce qu'elle devra ensuite dessiner, et non les paramètres définissant les éléments de dessin (lignes, surfaces), par exemple « la gravité », « l'équilibre », « éviter une autre forme »... Le programme du peintre Harold Cohen,

de l'université de Californie San Diego, génère des dessins en fonction de ces notions symboliques, ignorant toute interaction graphique directe. Une telle démarche simplifie les processus de composition et d'animation des images.

Nadia et Daniel Thalman (Miralab, université de Montréal, Québec) développent un système interactif de création et animation d'images tridimensionnelles, dont chaque séquence est décrite par un script caractérisant les divers objets de la scène, appliquant ainsi à l'imagerie électronique les techniques les plus évoluées de représentation des connaissances (cf. *Micro-Systèmes* n° 75, p. 108, mai 1987).

Des experts de la végétation

Mais l'apport de l'intelligence artificielle va bien au-delà de l'aide au graphisme. Les premiers systèmes experts opérationnels concernaient le diagnostic de maladies de végétaux, comme le fameux « Tom », conçu par Alain Bonnet à l'initiative de l'Institut national de la recherche agronomique (INRA) en collaboration avec la société Cognitech, appliqué aux plants de tomates et actuellement en cours de test.

L'INRA développe par ailleurs une vingtaine d'outils du même genre pour des végétaux différents. Après la réalisation d'un prototype sur grosse informatique, ces systèmes sont portés sur micro-ordinateurs. L'étape suivante consiste à exploiter un vidéodisque pour illustrer les questions posées au système à l'aide de photographies représentant des variétés de plantes, des systèmes typiques de maladies, etc. Afin de les mettre à la portée de tous, certains de ces systèmes seront mis sur minitel, ce qui permettra aux cultivateurs de prendre immédiatement les bonnes décisions, sans avoir besoin de recourir à un spécialiste, souvent éloigné du lieu de culture.

Quelques programmes, tel « Zea » pour la conduite de la culture du maïs, sont issus des projets agricoles du feu Centre mondial informatique, regroupés sous le nom d'ARPEGE (Aide à la résolution de problèmes pour l'élevage et la gestion des exploitations). Développé pour partie en Lisp avec le moteur d'inférence OPS5, Zea est censé remplacer plusieurs experts avec ses quatre modules spécialisés en travail du sol, irrigation, pathologie et variétés de maïs. Un tel système peut faire passer le rendement à l'hectare du simple au triple. A terme, Zea pourra être interrogé par terminal digital VT 100 via Transpac, par les centres d'en-

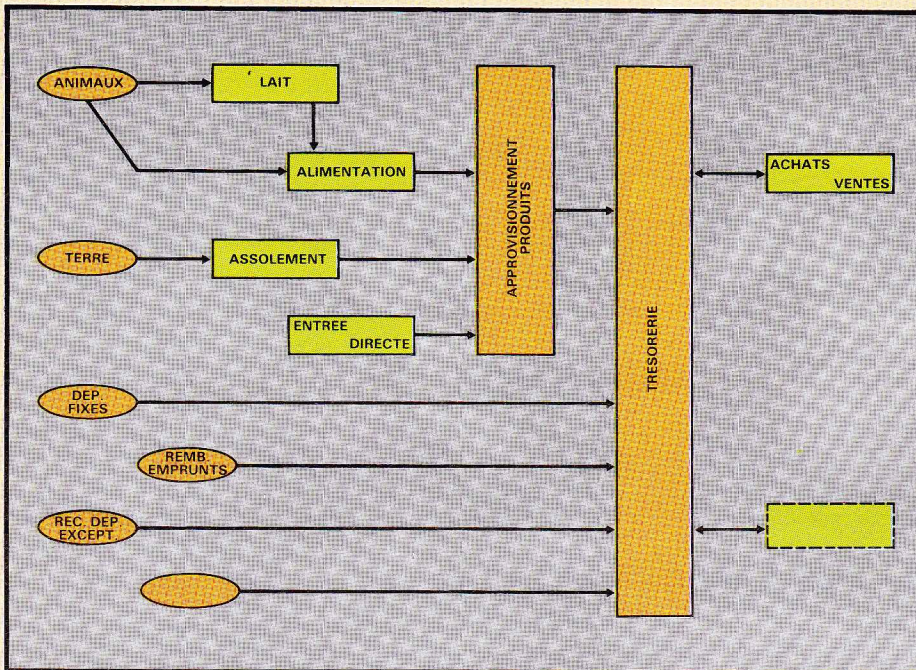


Fig. 5. — Le logiciel d'aide à la décision sur micro-ordinateur, produit par le Laboratoire d'économie rurale de Grignon (INRA) a pour objectif d'effectuer une prévision des recettes et des dépenses sur une période de 18 mois au maximum, pour des exploitations de polyculture-élevage. Ce programme comporte un module de choix d'assolement et calcul des produits et charges végétales.

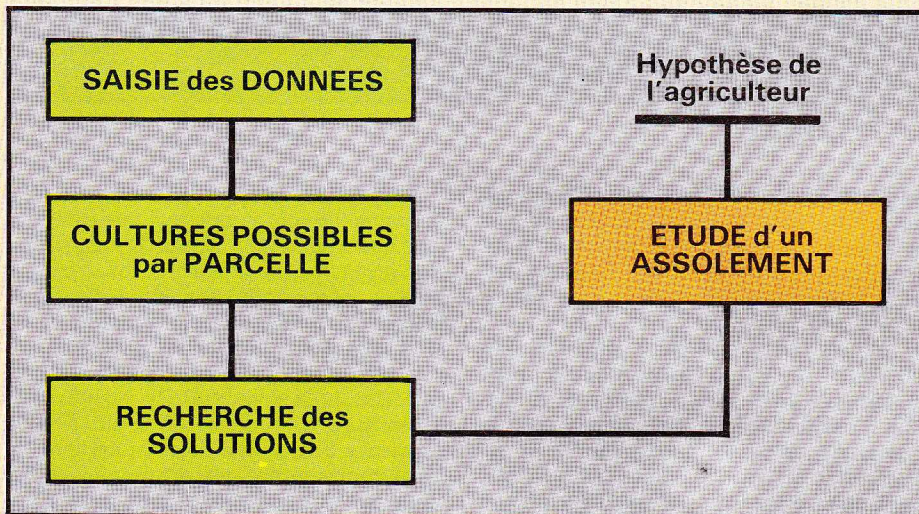


Fig. 6. — Organisation du logiciel de recherche d'assolements par parcelle (Laboratoire d'économie rurale INRA.) :

— Une fois l'ensemble des données saisies, le programme détermine, dans un premier temps, pour chaque parcelle, les cultures possibles et leurs marges ; cette phase tient compte de l'histoire de la parcelle, des successions autorisées et interdites, et des règles de répétitions.

— Ensuite, il recherche, sur l'ensemble des parcelles, les assolements qui vérifient les limites maximales de surface et qui apportent les marges totales les plus grandes.

— Enfin, il est possible de visualiser et de modifier ces solutions ou de les comparer avec les propres hypothèses d'assolements de l'utilisateur.

seignement et les instituts agricoles. Un accès par minitel est prévu afin de le mettre à la portée de tous les agriculteurs.

D'ores et déjà, le minitel met à la disposition de tous, les services offerts par Edimatic (par le 36 15, code EDIMK). Grâce à la « télématique agricole », l'utilisateur est aidé dans le choix de produits phytosanitaires,

dés herbants et autres fongicides. Un système interactif, baptisé « Ivraie », est capable d'identifier les mauvaises herbes, compte tenu d'une certaine incertitude (choix multiples) : le cultivateur décrit la plantule et le système lui fournit son nom.

Le Laboratoire d'économie rurale de l'INRA a développé un moteur d'inférence

pour micro-ordinateurs (Mimi) dont la caractéristique principale est d'être suffisamment simple pour ne pas devoir faire appel à des ingénieurs de la connaissance et autres spécialistes, afin que les conseillers agricoles et les agriculteurs puissent créer eux-mêmes, au niveau de la région, un système expert bon marché, facilement diffusable et tenant compte des particularités régionales : influence du terroir, de la région, des traditions... Lors d'une première expérience de formation de conseillers, ceux-ci ont réussi à réaliser en quelques jours un système expert appliqué au diagnostic et à l'aide à l'organisation du travail en agriculture.

Le même moteur d'inférence, associé au graphisme, a permis de réaliser un programme d'identification des arbres à partir des particularités de leurs formes, organes, couleurs... Chaque question posée est illustrée d'un dessin aidant l'utilisateur à y répondre. La conclusion — c'est-à-dire l'arbre identifié — est également assortie d'une image de cet arbre.

L'ordinateur aux champs

Sans parler des nombreux progiciels destinés à la gestion des exploitations agricoles (gestion des parcelles, aide à la décision...), il existe encore beaucoup de produits appliqués à l'agriculture et l'arboriculture. Citons, aux États-Unis, « Wheat Councillor » (ICI) pour la prévention des maladies du blé, « Pomme » (Virginia Polytechnic Inst.) pour la gestion des champs de pommiers, « Comax » pour la récolte du coton. Ce dernier, installé dans une ferme du Mississippi, a déjà fait ses preuves : en suivant ses propositions, notamment en commençant à récolter le coton à la date préconisée, la production totale peut être sensiblement augmentée, réalisant un gain de 60 dollars par acre de terrain. Aux Pays-Bas et au Danemark, des microprocesseurs équipent les serres et en contrôlent les divers paramètres (température, hygrométrie, luminosité...).

Bien que la plupart des logiciels actuellement disponibles et concernant les végétaux s'adressent aux agriculteurs, ces systèmes pourraient probablement être adaptés à la culture de n'importe quel arbre ou plante. L'un des pionniers de la micro-informatique agricole en France, Jean-Marie Attonaty, directeur de recherche au Laboratoire d'économie rurale de Grignon (INRA), a fait deux paris simultanés : créer des systèmes experts utilisables sur micro-ordinateur et donner les moyens aux techniciens de terrain de créer eux-mêmes les systèmes experts dont ils ont besoin.

Soulignant l'intérêt du traitement « sur place, en temps réel et en mode conversationnel », ce chercheur est, de son propre

aveu, l'un des « premiers à mettre un ordinateur dans la voiture pour l'amener à la campagne » (7). Mais cette utilisation pose de grosses difficultés : un grand nombre de données concernant la végétation, et l'écologie en générale, sont essentiellement qualitatives ; des indicateurs comme la couleur de la surface du terrain, la réaction au coup de botte de l'agriculteur, lui permettent, d'instinct, d'apprécier l'humidité ou la consistance du sol. De plus, si l'on peut faire des prévisions de récolte de fourrage, « rien ne permet de savoir si le printemps sera précoce ou tardif, sec ou pluvieux, cette année », remarque J.-M. Attonaty. Aussi les programmes d'aide à la culture devront tenir compte de l'incertitude de l'avenir, du caractère flou de certaines données.

Les logiciels de végétation assistée par ordinateur peuvent être utilisés en amont de programmes de gestion ou d'organisation de l'exploitation agricole (fig. 5). Les travaux de J.-M. Attonaty portent en particulier sur de telles applications. Cette gestion se fait en trois phases, correspondant à des échelles de temps différentes : « Certaines décisions se prennent au jour le jour. Vous choisissez un produit phytosanitaire (décision) puisque vous vous êtes aperçu que vous aviez des pucerons (diagnostic), et puis vous regardez ce qui va se passer après (suivi). »

Il existe aujourd'hui beaucoup de produits consacrés au suivi. L'aide à la décision fait le plus souvent appel à des programmes complexes de recherche opérationnelle, implantés sur gros systèmes. Quant au diagnostic, nous avons vu qu'il pouvait mettre en œuvre des systèmes experts, afin de tirer parti de toutes les données en présence.

Ainsi le logiciel d'aide au choix des cultures, conçu au Laboratoire d'économie rurale de Grignon et commercialisé sous le nom de « Plainlac » par Telciel, a pour objectif de fournir à un agriculteur une série d'assolements possibles dans son exploitation pour une année donnée en fournissant une culture pour chaque parcelle, et de rechercher la configuration qui maximise la marge brute totale.

Ce programme, très convivial et facile d'emploi, demande à l'agriculteur de saisir lui-même les données relatives aux parcelles (nom, surface, type de terre, historique des cultures), aux cultures (espèce, prix de vente, rendement, règles de succession et de répétition...), ainsi que les contraintes globales de l'exploitation (maxima de surface pour une culture donnée). A partir de là, le logiciel fournit par parcelle les cultures possibles en les classant par marge brute décroissante, leur rendement, leur marge à l'hectare, et la production et la marge totale de la parcelle.

Un certain nombre de solutions sont proposées à l'utilisateur pour la totalité de ses parcelles. Il peut aussi modifier ou définir lui-même un assolement et obtenir la marge brute correspondante (fig. 6).

Robots et végétaux

Forts des bons conseils qui leur auront été fournis par les ordinateurs, les agriculteurs, sylviculteurs et autres jardiniers doivent aussi les mettre en pratique au moment opportun, c'est-à-dire semer, repiquer, arroser, irriguer, tailler, récolter, cueillir...

Si l'idée des robots cueilleurs de fruits n'est pas une nouveauté (elle remonte à plus de vingt ans), elle n'a pris forme qu'au début des années quatre-vingts, simultanément au Japon, en Floride et en France. C'est ainsi que la machine développée au Centre d'étude du machinisme agricole, du génie rural et des eaux et forêts (CEMAGREF), la première du genre, répondant au doux nom de « Magali », a été testée dans les vergers de pommes du Centre expérimental de Marsillagues à partir de septembre 1986. Automotrice et téléguidée, Magali est capable de récolter environ 50 % de la production des arbres, à raison d'un fruit toutes les quatre secondes, ce qui correspond au rythme de deux cueilleurs manuels.

La machine est équipée d'une double commande, manuelle et automatique, ainsi que d'un module de vision (caméra et analyseur d'image) permettant le repérage des fruits d'après leur couleur. Un projecteur lui permet, en outre, de travailler de nuit. Pour rendre commercialisable un tel robot, il faut encore améliorer ses performances afin d'assurer une récolte à 80 %, précise Alain Bourelly, du Cemagref.

Cet organisme a d'autres projets dans ce domaine ; il étudie notamment la robotisation de la cueillette de fruits fragiles tels que la tomate et la fraise, et a prévu de réaliser des robots pour l'irrigation, ainsi qu'un bras robotisé capable d'arracher des arbres en milieu forestier. Des prototypes de matériels de débroussaillage ont déjà été réalisés, et les études actuelles portent sur la robotisation de l'exploitation des taillis et des premières éclaircies forestières.

D'autres machines, munies de systèmes de vision, sont capables d'effectuer le tri qualitatif des fruits, la récolte sélective des plantes mûres (asperges et tomates, par exemple). Des opérations délicates comme la tonte du gazon ou la taille des vignes s'effectueront également par des engins robotisés.

D'ores et déjà, un prototype de robot pour la transplantation de plantules cultivées *in vitro* a été développé par le professeur Yoshiyuki Miwa (université de Waseda, Japon). Pour cette opération extrêmement délicate, ce chercheur a réalisé un robot actionné par alliage à mémoire de forme (cf. *Micro-Systèmes* n° 70, p. 94, décembre 1986). Il permet une préhension douce et une plantation de semis

de 0,5 mm de diamètre et de 20 mm de longueur sur le milieu de culture, sans endommager les plantules. Un système de détection utilisant la plante comme une antenne a été conçu et développé ; la plantule est ainsi capable d'indiquer directement sa position, ce qui permet une détection précise dans le cas où l'espacement entre les plantules est supérieur à 5 mm. Y. Miwa a également mis au point un prototype reconnaissant l'état de croissance des semis, grâce à des détecteurs de couleur, lesquels peuvent, dans certaines limites, distinguer les plantules d'après leur taille en intervalles de 5 mm.

Un système entièrement automatisé de transplantation de plantules, intégrant tous ces éléments, a été réalisé avec succès. Il faut actuellement 60 secondes pour transplanter une plantule, mais l'auteur pense réduire ce temps à 30 secondes. Cette réalisation a été récompensée par un prix attribué par le groupe Moët-Hennessy, lors des journées « L'électronique et le pilotage des plantes », qui rassemblaient 300 scientifiques de 40 nationalités différentes, à Monaco, du 14 au 16 septembre 1987.

Ce colloque a, par ailleurs, permis à de nombreux biologistes, physiologistes des plantes, électroniciens et physiciens d'explorer ensemble les voies d'une coopération dans les techniques suivantes : programmation-régulation, automation-robotique, système experts, simulation sur écran, etc.

Le minitel arroseur

Depuis plusieurs années déjà, on sait maîtriser l'automatisation et la télésurveillance de l'irrigation. Des capteurs placés sur les branches ou les fruits peuvent être reliés à un système déclencheur de l'irrigation. Une PME provençale, Automatic Pilote Systèmes (APS), a mis au point un tel dispositif : le « Pepista ». Une centrale informatique dotée d'un logiciel enregistre et intègre les mesures, effectuées toutes les minutes pour la culture maraîchère ou toutes les demi-heures pour les vergers. La centrale commande de manière optimale l'ouverture des électrovannes du système d'irrigation, en fonction de la courbe journalière cumulée des micro-variations.

La même société a développé un système complémentaire, « Transagri Puls'Abs », qui offre l'intérêt d'utiliser un minitel comme terminal d'accès. Il suffit donc de disposer d'un minitel et, pour le prix d'une conversation téléphonique, il est possible d'interroger ou de modifier le programme permettant de déclencher à distance l'arrosage d'un jardin, d'un champ ou d'une pelouse. Le pilotage s'effectue par l'intermédiaire d'un système à microprocesseur muni d'un modem. Cette connexion fait aussi office d'alarme à dis-

tance : en cas d'anomalie du système d'irrigation (blocage de l'électrovanne, par exemple), Transagri Puls'Aps appelle automatiquement l'un des quatre numéros de téléphone qu'il est capable de mémoriser. Toutes les informations sont visualisables sur l'écran du minitel, et la programmation s'effectue à partir du clavier, l'accès au système étant protégé par un mot de passe.

S'il est possible de télécommander l'arrosage du jardin, on peut également télécommander une tondeuse à gazon, comme s'il s'agissait d'un jouet d'enfant. C'est le cas du robot PX 2100 de la firme japonaise Kubota. Quant au LARFRA, à Bordeaux, il étudie un robot tondeur de gazon automatique qui fonctionnera par mémorisation préalable du trajet à parcourir, avec l'aide d'un capteur ultrasonore.



Fleur générée par AMAP et visualisée à l'aide du logiciel Wavefront Technologies.

La végétation vue de satellite

Au début de cet article, nous avons vu comment on peut visualiser la végétation et son évolution sur l'écran d'un ordinateur. Mais il est aussi essentiel de l'observer en réalité, afin de détecter certaines caractéristiques : variations de la croissance en fonction du terrain, influence de la pollution, évolution de maladies des arbres...

L'apport de l'électronique et de l'informatique ne sont pas négligeables dans ce domaine. Ainsi le groupe de Télé-détection du Centre national d'études des télécommunications (CNET), qui cherche à développer l'utilisation des techniques hyperfréquences embarquées sur avion ou satellite pour l'observation des ressources terrestres, a lancé une campagne, en liaison avec l'INRA et le Cemagref, pour observer la végétation. Le radar ERASME (Etude radar des sols et de la mer), mis au point par la division RPE, était destiné à observer la forêt tropicale en Guyane française du 21 septembre au 10 octobre 1986, afin de déterminer ses caractéristiques et sa variabilité (variation diurne, effet de la pluie...). Ces études expérimentales ne sont pas encore exploitées, mais elles indiquent que l'onde radar, capable de pénétrer la forêt jusqu'au niveau du sol, peut fournir une information sur la structure verticale de la végétation à l'intérieur de la forêt.

L'une des applications de SPOT (Satellite pour l'observation de la terre) vise également la végétation. Les spécialistes considèrent, en effet, comme très positif l'apport du cliché spatial pour avertir de phénomènes non visibles depuis le sol, évaluer la qualité des récoltes, dresser un inventaire des ressources ligneuses, procéder à des comparaisons entre espace cultivable et espace cultivé...

Des logiciels sont développés pour interpréter ces images de télé-détection. C'est le cas de Multiscope (Cap Gemini Sogeti) qui,

doté de fonctions de gestion des données images, de visualisation, d'analyse, de traitement et de classification d'images, peut être appliqué aux recensements et suivi de l'évolution et de la santé des cultures, prévision de récolte, analyse de catastrophes naturelles (inondation, sécheresse, maladie des végétaux), au suivi de l'évolution d'une forêt, de la santé des arbres, de leur recensement... A l'heure actuelle, aucune application de ce type n'a encore été développée pour Multiscope.

Halte aux feux de forêts

L'informatique constitue également une arme contre les feux de forêts. Au début de l'été 1987, les services de lutte contre les incendies des Alpes-Maritimes ont installé un des tout premiers systèmes experts permettant une analyse plus rapide et plus sûre des risques.

Développé par le groupe Intelligence artificielle et systèmes experts de l'École des Mines de Paris, à Sophia Antipolis, Expertgraph simule, en effet, le risque potentiel de propagation du feu si celui-ci devait se déclarer, et le présente sur écran à l'aide de cartes du département comportant des zones de couleurs différentes selon les degrés de danger. Les pompiers peuvent alors intervenir plus efficacement. Ecrit en Pascal et tournant sur ordinateur MicroVax de Digital, il prend en compte la végétation, la topographie, les risques particuliers tels que les habitations et sites industriels, le vent, l'humidité, la température, etc. Pour ces dernières connaissances, le système est relié par le réseau téléphonique à des banques de données de la météorologie nationale.

Afin de lutter plus efficacement contre les incendies de forêt, un étonnant système anti-incendie a été mis au point par Protect Forest, une PME varoise, avec l'aide de

l'Anvar. Pégase, tel est son nom, est un réservoir d'eau sous pression, enterré et prêt à jaillir à la première alerte fournie par un réseau de capteurs de température fixés aux arbres voisins et gérés par une centrale d'alarme à microprocesseurs. Quatre de ces installations fonctionnent actuellement au voisinage de l'autoroute du Sud, et deux autres sont à l'étude pour le parc industriel de Sophia-Antipolis.

Des plantes, des jeux et des hommes

Pour conclure sur une note moins grave, signalons que de nouveaux jeux sur ordinateur mettent en scène des végétaux. C'est le cas de « Canadair », logiciel commercialisé par France Image Logiciel, qui simule le rôle et les fonctions d'un « pompier du ciel ». Dans « Garden Party », un jeu d'arcade de Free Game Blot, le joueur doit aider le jardinier dans ses différentes tâches : plantation, arrosage, récolte, lutte contre les insectes et contre les caprices de la météorologie.

Enfin, nous pouvons, avec J.-M. Attonaty, nous poser la question : « Pourquoi les informaticiens sont-ils arrivés plus tard en agriculture que dans certains autres secteurs ? » Probablement parce que les données sont beaucoup plus complexes, qualitatives, changeantes, incertaines, et qu'il a fallu attendre l'avènement de l'intelligence artificielle et des systèmes fondés sur la logique floue (cf. *Micro-Systèmes* n° 64, p. 93, mai 1986) pour voir apparaître des bouleversements dans les rapports entre les agriculteurs, botanistes, instituts de recherche fondamentale et appliquée, techniciens agricoles, bref, l'ensemble des professions gravitant autour des végétaux.

Claire Rémy

Bibliographie

- (1) « L'image de synthèse en quête de réalisme », par Michel Fantin, *La Recherche*, n° 167, p. 830, juin 1987.
- (2) « Simulation de l'architecture des arbres », par P. de Reffye, C. Edelin, M. Jaeger, C. Cabart, Colloque international *L'arbre*, Montpellier, 9-14 septembre 1985.
- (3) « Synthèse d'images de feuilles végétales », par P. Lienhardt et J. Françon, *Mari* 87, 18-22 mai 1987.
- (4) « Modélisation de l'architecture des arbres par des procédés stochastiques », thèse de doctorat d'Etat, par P. de Reffye, Université de Paris-Sud, Orsay, 1979.
- (5) « Modélisation de la géométrie d'un couvert végétal : cas du maïs », par F. Aries, P. Monestiez, P. de Reffye et L. Prevot, *Mari* 87, 18-22 mai 1987.
- (6) « Les fractales, les ordinateurs et l'ADN », par P. Oppenheimer, *Mari* 87, 18-22 mai 1987.
- (7) « Pourquoi les systèmes experts ? » par J.-M. Attonaty, *Purpan* 438, 1986, p. 1-4.

ORDINATEURS COMPATIBLES IBM-PC XT/AT/386*



Bios legal AWARD*
 licence CONTROL RESET*



Bios legal AWARD*
 licence CONTROL RESET*



Compatibles de table

PC/XT*

- Coffret et clavier AZERTY.
- Alimentation 150 Watts.
- Carte mère **TURBO** équipée 256 K.
- Lecteur 360 K et contrôleur.
- Port parallèle.
- Carte couleur graphique.

~~3620 F ttc~~
3420 F ttc

AVEC
 CARTE ECRAN,
 PORT //

Réf. : IEETX

AT 286*

- Coffret et clavier AZERTY.
- Alimentation 165 Watts.
- Carte 512 K **BABY TURBO**
- Lecteur 1.2 Mo **NEC*** + contrôleur.
- Port parallèle.
- Carte couleur graphique.

~~7590 F ttc~~
7090 F ttc

AVEC
 UNE SOURIS
 GRATUITE

Réf. : IEEEX

AT 386*

- Coffret, clavier et Alim.
- Carte mère **BABY** 16 MHz 512 K.
- Lecteur 1.2 Mo **NEC*** + contrôleur.
- Disque dur 20 Mo et contrôleur.
- Ports série et parallèle.
- Carte couleur graphique.
- Ecran monochrome 12".

~~31900 F ttc~~
27990 F ttc

Réf. : AX386

Portables compatibles

PC/XT*

- Coffret ABS, clavier et Alim.
- Ecran 9" intégré.
- Carte mère **TURBO** équipée 640 K.
- Carte couleur graphique.
- Lecteur 360 K et contrôleur.
- Disque dur 30 Mo + contrôleur.
- Ports série, parallèle, joystick.
- Horloge calendrier.

~~13990 F ttc~~
12990 F ttc

Réf. : IEEEPX

AT 286*

- Coffret ABS, clavier et Alim.
- Ecran 9" intégré.
- Carte 640 **BABY TURBO**
- Carte couleur graphique.
- Lecteur 1.2 Mo **NEC*** + contrôleur.
- Disque dur 20 Mo et contrôleur.
- Ports série et parallèle.

~~19990 F ttc~~
17600 F ttc

EN CADEAU
 UNE SOURIS

Réf. : IEEEXA

AT 386*

- Coffret **ABS***, clavier et Alim.
- Ecran 9" intégré.
- Carte mère **BABY** 16 MHz 640 K.
- Carte couleur graphique.
- Lecteur 1.2 Mo **NEC*** + contrôleur.
- Disque dur 20 Mo et contrôleur.
- Ports série et parallèle.

~~37990 F ttc~~
32990 F ttc

Réf. : 386PA

Portables LCD compatibles

PC/XT*

- Coffret **ABS***, clavier et Alim.
- Ecran plat LCD 640 * 200.
- Carte mère **TURBO** équipée 640 K.
- Carte couleur graphique + LCD.
- Lecteur 360 K et contrôleur.
- Disque dur 30 Mo + contrôleur.
- Ports série, parallèle, joystick.
- Horloge calendrier.

~~16790 F ttc~~
14450 F ttc

AVEC
 UNE SOURIS
 GRATUITE

Réf. : IEEEXC

AT 286*

- Coffret **ABS***, clavier et Alim.
- Ecran plat LCD 640 * 200.
- Carte 640 K **BABY TURBO** 10 MHz.
- Carte couleur graphique + LCD.
- Lecteur 1.2 Mo **NEC*** + contrôleur.
- Disque dur 20 Mo et contrôleur.
- Ports série et parallèle.

~~23490 F ttc~~
16990 F ttc

Réf. : IEEEXA

AT 386*

- Coffret **ABS***, clavier et Alim.
- Ecran plat LCD 640 * 200.
- Carte mère **BABY** 16 MHz 640 K.
- Carte couleur graphique + LCD.
- Lecteur 1.2 Mo **NEC*** + contrôleur.
- Disque dur 20 Mo et contrôleur.
- Ports série et parallèle.

~~47390 F ttc~~
34990 F ttc

Réf. : 386CA

OPTIONS

| | | |
|--------------|--|----------|
| SMALL | Coffret "mini" pour IEEE TX, AX ou 386 | + 500 F |
| LOOK | Coffret métal look AX* pour IEEE TX | + 300 F |
| VCAS | Option clavier étendu avec curseurs séparés | + 300 F |
| VCMNG | Option carte type HERCULES* (reprise CCLRG) | + 100 F |
| VCEGA | Option carte type EGA* (reprise CCLRG) | + 1050 F |
| VCMIO | Option carte multifonction avec reprise XTFD pour IEEE TX | + 700 F |
| ATMIO | Carte multifonction RS232 et parallèle pour IEEE AX et 386 | + 550 F |
| DS2 | Dos 2.11 + GWBASIC* + manuel français | + 490 F |
| TLD12 | Ecran 12" bifréquence (type HERCULES* et composite) | + 990 F |
| TLD14 | Ecran 14" bifréquence (type HERCULES* et composite) | + 1290 F |
| EGA14 | Ecran 14" couleur EGA* 650 * 400 | + 3550 F |

| | | |
|--------------|---|----------|
| FD5T | Lecteur 360 K supplémentaire made in Taiwan | + 990 F |
| FD5J | Lecteur 360 K supplémentaire made in Japan | + 1160 F |
| FD12 | Lecteur 1.2 Mo supplémentaire NEC* made in Japan | + 1250 F |
| FD3S | Drive 3" 1/2 made in Japan 720 K | + 1190 F |
| FD3D | Drive 3" 1/2 made in Japan 1,44 Mo | + 2090 F |
| FD2X | Kit 20 Mo pour IEEE TX (FD 20 M + CHD) | + 3095 F |
| FD2A | Kit 20 Mo pour IEEE AX et 386 (DF20M + FDH) reprise ATFD | + 3499 F |
| FD3X | Kit 30 Mo pour IEEE TX (FD30M + CRL) | + 3999 F |
| FD4A | Kit 40 Mo pour IEEE AX et 386 (FD40M + FDHD) reprise ATFD | + 7590 F |
| ST40X | Streamer archive 40 Mo + câbles et soft pour IEEE TX | + 3990 F |
| ST40A | Streamer archive 40 Mo + câbles et soft pour IEEE AX et 386 | + 4290 F |

NOMBREUSES AUTRES OPTIONS DISPONIBLES. CONTACTEZ-NOUS (1) 42.93.47.32

ACCESSOIRES

| | | |
|---------------|---|-------|
| STDUC | Support vertical pour systèmes | 480 F |
| STDCA | Tiroir amovible pour clavier | 630 F |
| COVER | Capot de protection pour clavier | 170 F |
| CLOCK | Carte horloge pour XT* avec batterie | 290 F |
| MOUSE | Souris graphique avec soft pour XT* ou AT* | 390 F |
| MDAO | Mouse + logiciel de DAO pour XT*, AT* ou 386* | 540 F |
| JSKIBX | Joystick autocentreur pour XT* ou AT* | 190 F |

2 nouveaux points de vente au cœur de PARIS

PARIS 12^{eme}
 60, cours de Vincennes
 75012 PARIS

Métro : Porte de Vincennes

PARIS 15^{eme}
 près de la Porte de Versailles
 44, rue Cronstadt
 75015 PARIS

Métro : Convention

**BON A DÉCOUPER ET A
 Renvoyer à CONTROL RESET
 34 RUE DE TURIN 75008 PARIS
 SERVICE-LECTEURS N° 272**

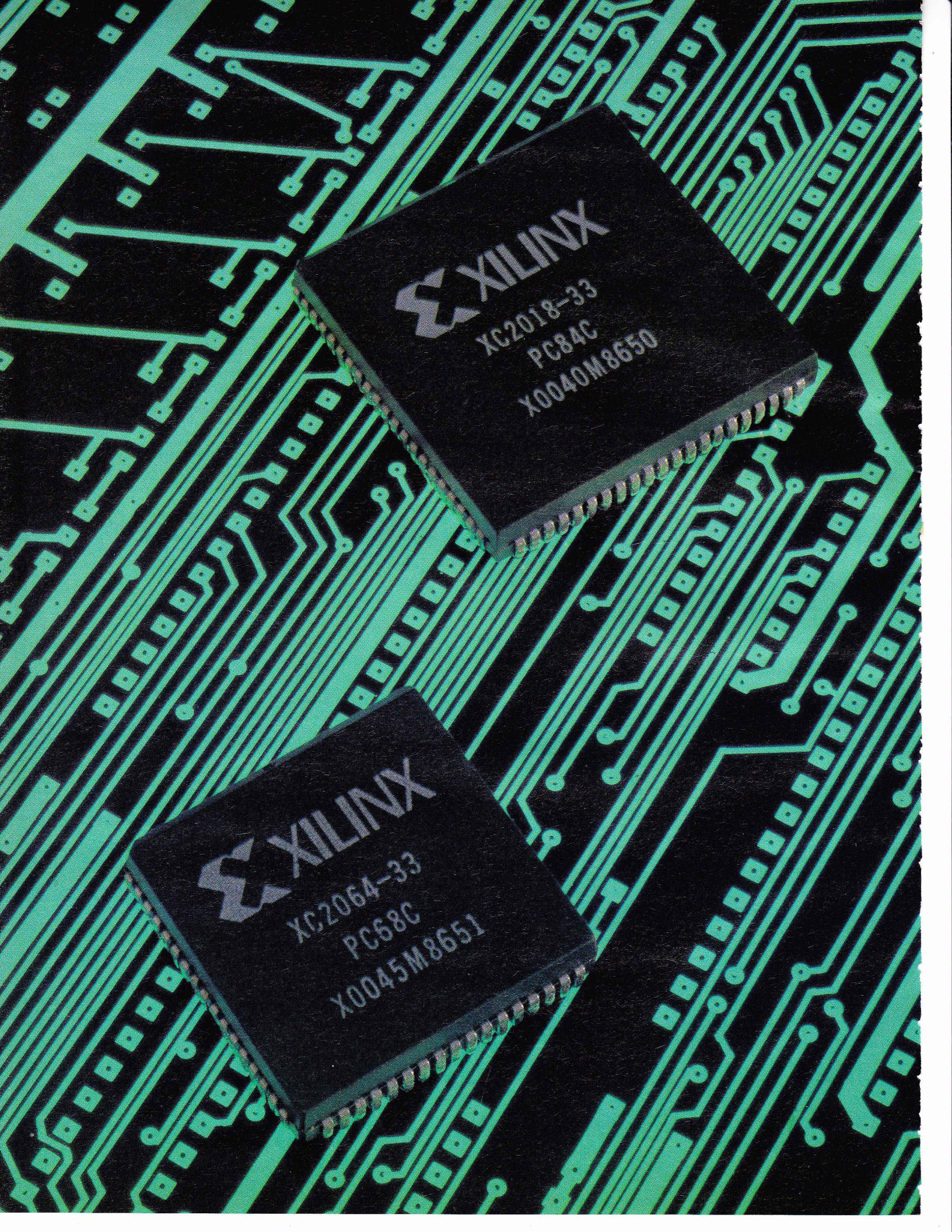
Marque d'ordinateur _____
 Signature : _____

OUI, je commande le matériel suivant :

Frais de port
 Total TTC

VITE! envoyez-moi votre catalogue gratuit

Nom : _____
 Prénom _____
 Adresse _____
 Code postal _____ Ville _____



XILINX

XC2018-33
PC84C
X0040M8650

XILINX

XC2064-33
PC68C
X0045M8651

LES PGAs DE LA SERIE 3000 DE XILINX

Les LCA allient densité et maniabilité des réseaux de portes classiques aux avantages certains d'une programmation *in situ* par l'utilisateur. Avec une architecture du type réseau et une complète autonomie de l'utilisateur, ils représentent une nouvelle génération de réseaux logiques dont les performances et la facilité de mise en œuvre dépassent de loin celles des circuits conventionnels. Leur introduction en 1985 par la société Xilinx préfigurait déjà une nouvelle génération de circuits programmables.

Le but principal des concepteurs de systèmes a toujours été d'intégrer toute la logique qui entoure leurs microprocesseurs et mémoires au sein d'un produit standard. A la limite, un système pourrait être composé de tous les produits VLSI standards (fig. 1) avec l'ensemble des caractéristiques logiques propres au système définies de façon logicielle. Ce vieux rêve devient maintenant réalité grâce à l'avènement des PGA (Programmable Gate Arrays). En effet, les circuits de la série 3000 de Xilinx permettent d'intégrer toutes les fonctions logiques d'un système au sein d'une seule et même famille de circuits, avec des performances et un gain d'espace accrus.

Malgré les qualités des réseaux logiques traditionnels, les concepteurs avaient toujours du mal à y intégrer un certain nombre de fonctions telles que décodeurs complexes, multiplexeurs, compteurs larges ou autres. Ils se résignaient la plupart du temps à les loger dans des MSI spécifiques externes aux LCA (Logic Cell Array, pour réseaux logiques de cellules) dont les coûts de développement et de production étaient élevés. La nouvelle génération de PGA permet maintenant une intégration plus complète et une souplesse d'emploi sans précédent.

L'architecture PGA

L'architecture de base de la famille de produits PGA de Xilinx se décompose grossièrement en trois parties (fig. 2) : les fonctions d'entrée-sortie, les ensembles logiques programmables et les éléments d'interconnexion.

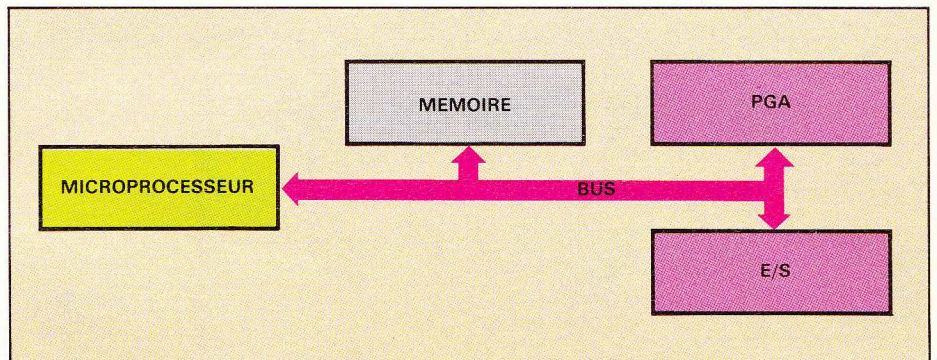


Fig. 1. — Un PGA au sein d'un système à microprocesseur.

Conçu en technologie C.MOS faible consommation, cette architecture de seconde génération employée dans les circuits de la série 3000 utilise les concepts de base des produits 2064 et 1018. Entrées-sorties, cellules logiques configurables et interconnexions programmables par l'utilisateur garantissent un maximum de souplesse d'emploi et de performances. Chacun de ces trois éléments a eu une évolution qui a tenu compte des souhaits des utilisateurs de la précédente génération de circuits.

Les blocs d'entrées-sorties (fig. 3) disposent de fonctions et d'options améliorées au point de pouvoir programmer le comportement et les caractéristiques électriques de chacune des broches du circuit. Ils sont en quelque sorte l'interface programmable entre les éléments internes du PGA et les broches externes du boîtier.

Signaux d'entrée

Sur la circuiterie d'entrée, l'utilisateur a

accès soit directement au signal présent sur la broche, soit à la sortie d'un élément de stockage (bascule). Ceci permet d'allouer deux fonctionnalités d'entrée différentes pour une broche, particularité intéressante pour un environnement microprocesseur.

Enfin, l'élément de stockage pourra être soit un latch transparent, soit une bascule du type flip-flop, ce qui permet de traiter des flots de données multiplexées.

Signaux de sortie

La circuiterie de sortie (qui aboutit bien sûr à la même broche du circuit) pourra être soit latchée, soit directe. Sa polarité pourra être inversée ou non selon la programmation qu'aura mise en place l'utilisateur. De même, la connexion du buffer trois états de sortie et la polarité du signal qui le pilote est définissable.

La mise en place optionnelle d'une bascule permet d'envisager des sorties pipées. La programmation du temps de montée du buffer de sortie autorise non seulement une

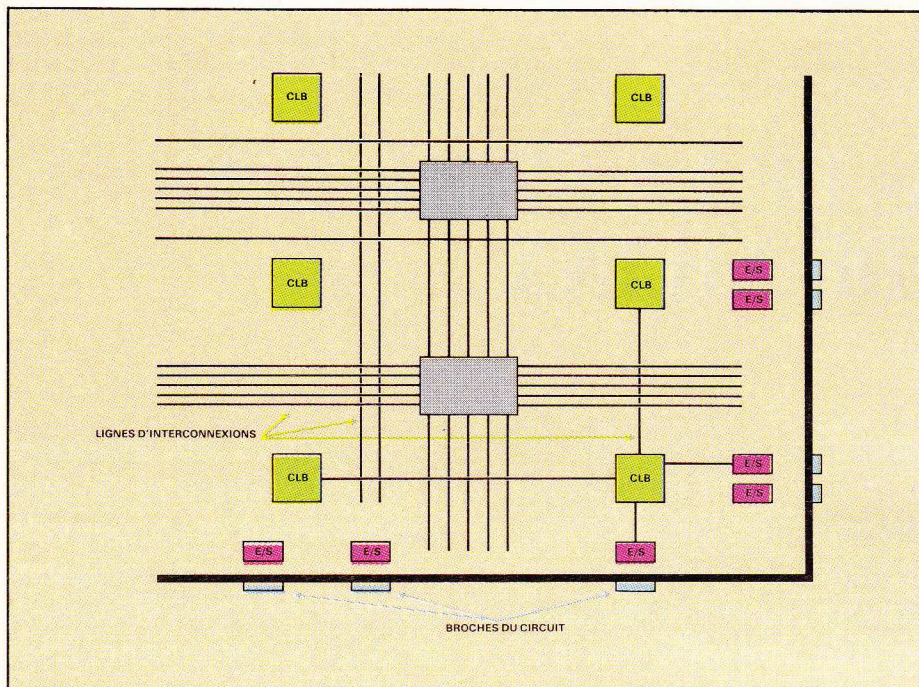


Fig. 2. - Structure générale d'un circuit PGA.

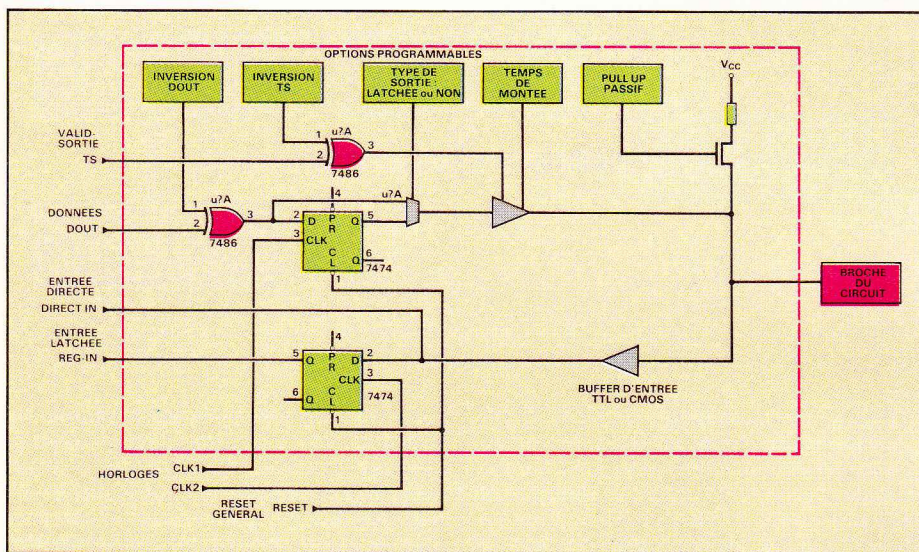


Fig. 3. - Bloc d'entrées/sorties.

modulation des temps de réponse en fonction de l'application mais également une consommation moindre et une réduction notable du bruit.

Pour en terminer avec les fonctionnalités des éléments d'entrée-sortie, notons pour les lecteurs non familiers de ce genre de circuits que comme pour les générations précédentes de LCA, la sensibilité des niveaux d'entrée est compatible TTL ou CMOS. Cela fait partie des options de programmation de base du circuit.

Cellules logiques configurables

Elles constituent en quelque sorte des blocs de construction de fonctions logiques

de base que l'on retrouve au sein de tout système numérique. Chacune d'entre elles contient un ensemble de fonctions combinatoires et des éléments de stockage dont les fonctionnalités et la structure d'ensemble est programmable par l'utilisateur. Plusieurs cellules associées peuvent évidemment former des blocs de traitement plus complexes. Elles sont disposées en matrice au sein du boîtier.

En comparaison avec les versions antérieures de circuits, la série 3000 conserve les mêmes caractéristiques de structure à cela près que cinq fonctions combinatoires sont maintenant à disposition de l'utilisateur (au lieu de quatre) et que les restric-

tions sur les signaux d'entrée n'existent plus. De plus, grâce aux deux éléments de stockage (fig. 4), une fonction à sept variables peut être réalisée.

Comme pour les circuits 2064 et 1818, l'ensemble combinatoire à cinq entrées peut être scindé en deux ensembles de traitement à quatre variables. Dans cette configuration, les deux fonctions F et G pourront piloter soit les bascules de stockage, soit les sorties des cellules logiques.

Deux bascules de stockage par cellule sont à disposition de l'utilisateur. Chacune dispose d'une entrée de données propre, mais elles se partagent une horloge, un signal de validation d'horloge et un reset asynchrone communs. La donnée en entrée (DIN) peut être directement reliée à ces bascules sans recourir au bloc combinatoire. Ceci permet de simplifier les opérations de transactions de bus et d'assurer au sein d'une même cellule un traitement de donnée et son stockage en vue d'une utilisation ultérieure.

Les sorties peuvent provenir de l'élément de stockage ou de l'élément combinatoire. Les signaux F et G ainsi que Q₁ et Q₂ disposent des mêmes fonctionnalités, ce qui permet une gestion des sorties à partir des lignes d'interconnexions plutôt que de la cellule elle-même. Cette symétrie entre combinatoire et séquentiel réduit les temps de réponse de la cellule et simplifie la gestion de ses signaux d'entrée.

Enfin, le cadencement de ces cellules peut se faire à des fréquences avoisinant 70 MHz selon la version, et, en exemples de configuration, le constructeur fournit une bibliothèque de fonctions qui regroupent la plupart des composants actuels.

Les lignes d'interconnexion sont composées de métallisations et interconnectées entre elles par des commutateurs électroniques sont la base de toute la liberté laissée à l'utilisateur de composer comme il le souhaite un agencement plus ou moins complexe des cellules entre elles.

Trois types de lignes d'interconnexion sont disponibles : les lignes d'usage général, les grandes lignes et les connexions directes.

Lignes d'usage général

Elles sont au nombre de cinq pour chaque ligne et chaque colonne de cellules et parcourent l'ensemble du circuit de façon orthogonale. Ce sont elles qui effectueront le routage des signaux locaux.

Les métallisations qui les forment peuvent être interconnectées par une matrice de commutateurs se trouvant à l'intersection de chacune des lignes et des colonnes (fig. 5). Il est possible de créer n'importe quelle combinaison de connexions puisque chacune de ces lignes dispose d'un élément de commutation dans la matrice (fig. 6).

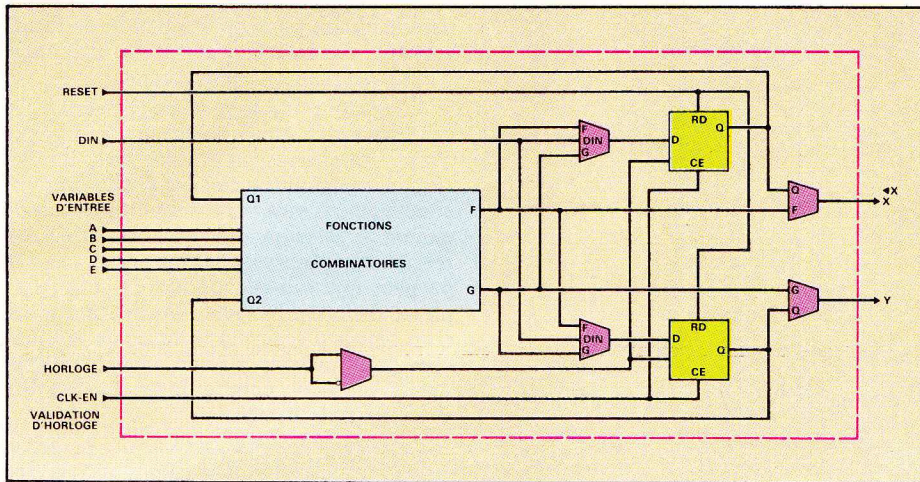


Fig. 4. - Structure d'une cellule logique configurable.

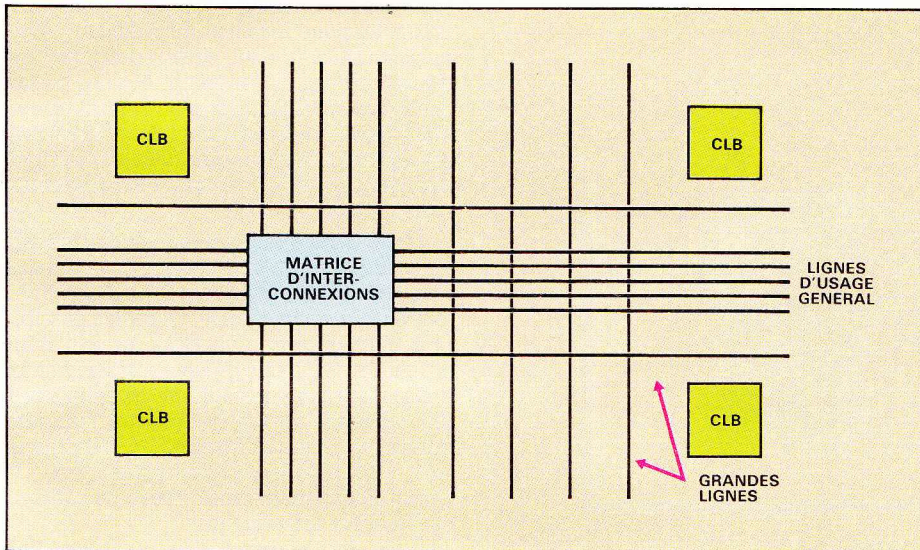


Fig. 5. - Lignes d'interconnexion.

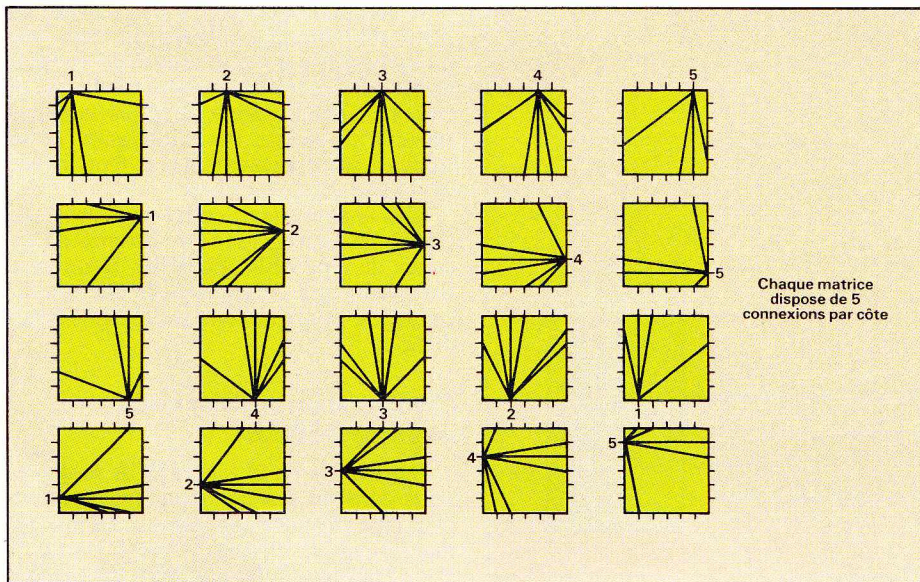


Fig. 6. - Possibilités de connexion offertes par une matrice.

Grandes lignes

Les grandes lignes (au nombre de deux horizontalement et de quatre verticalement) seront plutôt destinées à router des signaux dont les temps de propagation sont plus critiques (horloges, etc.). Des tampons rapides peuvent éventuellement être utilisés sur deux des lignes verticales qui peuvent également servir de routeurs généraux au même titre que celles décrites plus haut. Evidemment, des commutateurs programmables sont présents pour les relier aux lignes d'usage général, aux cellules logiques et aux blocs d'entrée-sortie, mais ces lignes ne transitent pas par les matrices de commutation.

Lignes de connexion directe

Ces lignes ultra-rapides fournissent la possibilité de connecter directement n'importe quel bloc d'entrée-sortie ou cellule à un élément adjacent, ceci dans les quatre directions. La conséquence directe de ces routages rapides est de permettre le « montage » d'une structure de cellules dont l'interdépendance est critique, par exemple, la réalisation de registres à décalage ou de compteurs dont la fréquence d'horloge est élevée. De même pour des structures du type pipeline, les données pourront évoluer dans le sens vertical ou horizontal avec une même facilité.

Pour finir, ces lignes de connexion directe forment le routage le plus performant du circuit.

Particularités

Chaque colonne d'interconnexion dispose de deux tampons trois états contrôlables sélectivement qui les relient aux lignes horizontales adjacentes. Chacun d'eux pilote une grande ligne (fig. 7). L'entrée et la commande de ces tampons sont programmables par l'utilisateur. Ceci permet la réalisation de bus internes trois états sur lesquels peuvent être multiplexées des données de nature différente. Des résistances de pullup sur ces grandes lignes horizontales permettent, par exemple, de réaliser des ET câblés dont les variables d'entrée sont programmables.

Implantation et mise en œuvre

L'implantation d'un PGA au sein d'un ensemble de circuits se fait évidemment après avoir défini quelles peuvent être les différentes fonctionnalités qu'il sera amené à effectuer. Notons au passage qu'un PGA peut remplir une multitude de rôles différents selon la configuration logicielle qu'on lui soumet. Cette configuration reste bien entendu à l'entière discrétion de l'utilisateur qui n'a plus besoin de moyens de développement inabornables.

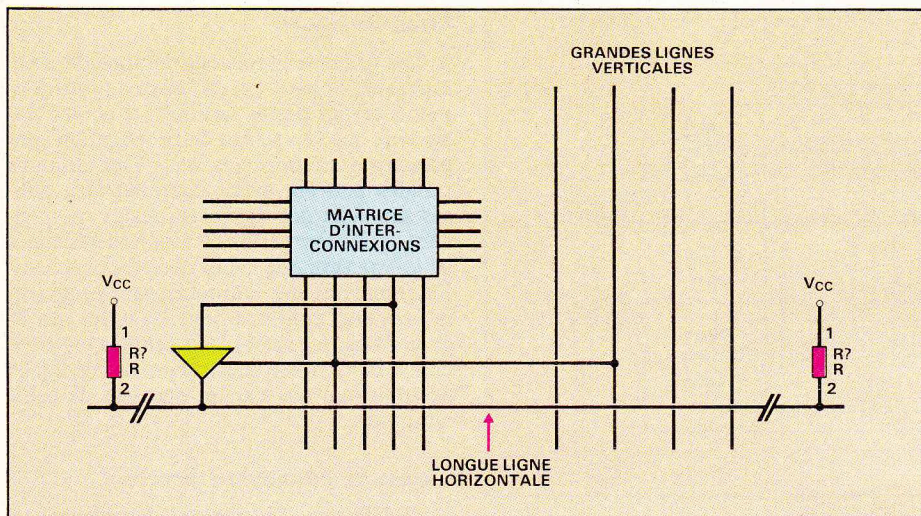


Fig. 7. - Les tampons 3 états programmables.

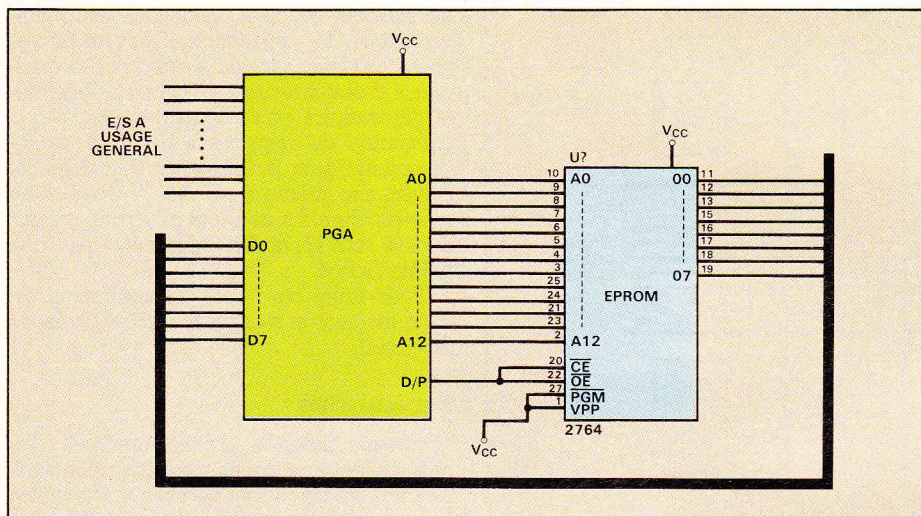


Fig. 8. - Autoconfiguration à partir d'une Eprom.

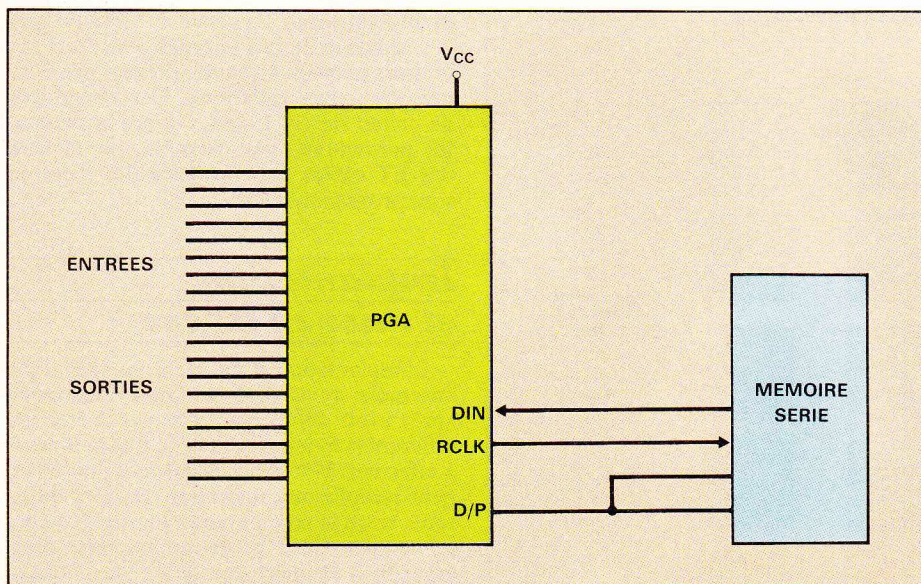


Fig. 9. - Autoconfiguration par mémoire série.

En effet, il suffit de télécharger un fichier vers le circuit pour qu'il réponde de façon adaptée. Ce téléchargement ne demande qu'une poignée de millisecondes.

Voilà donc un tout autre aspect de la question en ce qui concerne toute la génération des circuits programmables. La dynamique engendrée par ces possibilités est effectivement sans précédent et, à une époque où le développement des matériels est réservé aux rares sociétés bien équipées, on ne peut que saluer l'arrivée de cette souplesse de mise en œuvre. Dorénavant, plus n'est besoin d'énormes volumes de production pour justifier l'étude, le développement, la mise au point et la production des circuits à la demande.

Procurez-vous un circuit de la série 3000, entourez-le d'un minimum d'éléments vitaux (microprocesseur, mémoires, etc.), câblez une maquette en fonction de vos besoins et il ne vous reste plus qu'à mettre au point le fichier de données nécessaire au PGA (un PC suffit), pour qu'il se transforme en un mouton à cinq pattes dont vous n'avez jamais osé rêver. Bien entendu, les choses ne sont jamais vraiment aussi simples mais la procédure de développement est devenue considérablement moins lourde.

Nous n'examinerons qu'ultérieurement les procédures de constitution des données nécessaires à la programmation d'un PGA. Passons en revue pour l'instant les possibilités d'intégration d'un PGA au sein d'un ensemble. Pour cela il faut considérer deux points fondamentaux : d'une part le circuit doit être câblé de façon à pouvoir effectuer les fonctions que l'on attend de lui, d'autre part, il doit pouvoir être programmé. Il existe globalement trois modes de câblage qui assurent cette dualité.

Autoconfiguration

Illustrés en figure 8 et 9, les deux modes d'autoconfiguration nécessitent une mémoire parallèle ou série connectée au PGA et qui contiendra les données destinées au circuit. Le PGA génère les adresses nécessaires à la lecture de la mémoire et se trouve immédiatement prêt à fonctionner dès que la lecture a eu lieu.

Bien entendu, la mémoire peut être aussi une RAM sauvegardée par batterie ou un circuit du type ZEROPOWER (RAM avec batterie intégrée) dans laquelle une modification de configuration peut être immédiate.

Configuration en daisy chain de plusieurs PGA

Dans le cas où plusieurs PGA sont présents dans un système, leur configuration peut être effectuée par l'un d'entre eux (le maître) qui se chargera de lire la mémoire et de transmettre les données sous forme série aux PGA esclaves.

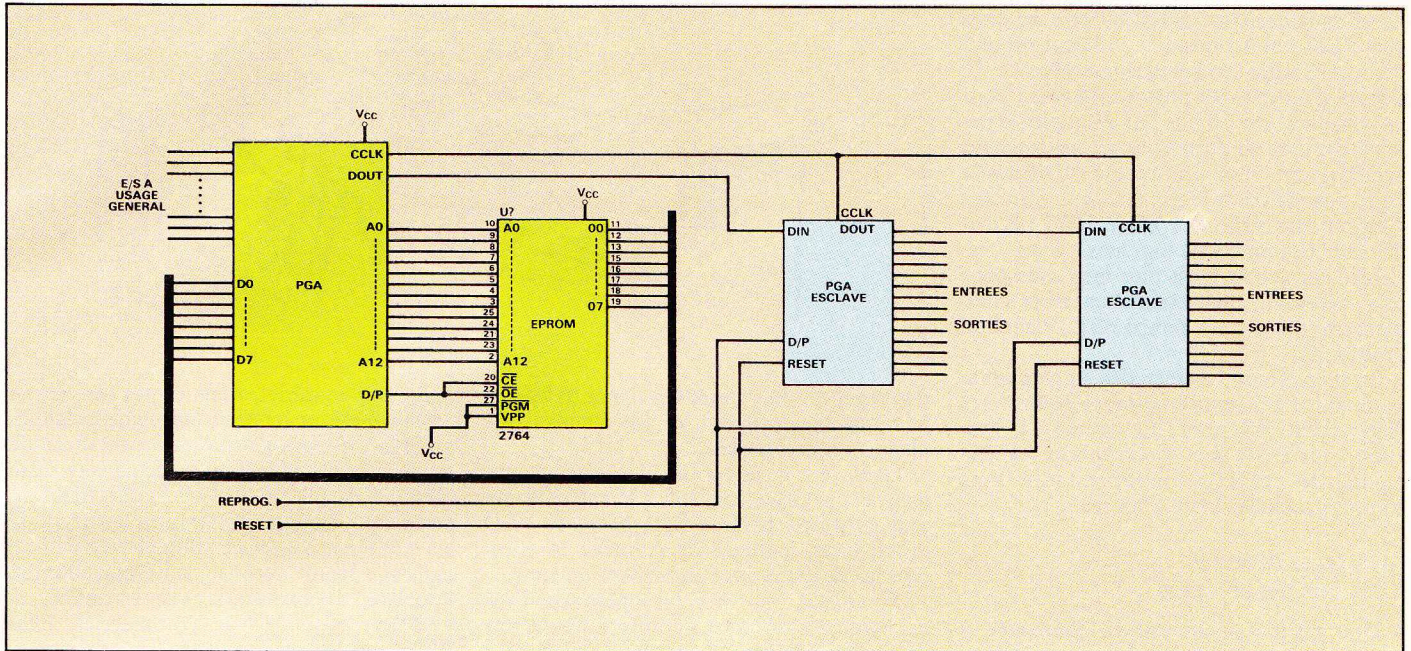


Fig. 10. – Configuration maître/esclave.

Un exemple de circuit à trois PGA est illustré en figure 10.

Configuration en mode périphérique

Dans ce mode, le PGA est connecté au bus d'un microprocesseur ou d'un microcontrôleur. C'est le mode le plus séduisant car le circuit reste dynamiquement reconfigurable par le processeur à tout moment et peut selon les exigences de l'environnement présenter un comportement adapté (fig. 11).

Utilisation

Pour présenter le plus pragmatiquement possible les diverses applications des PGA, nous énumérerons des exemples typiques d'utilisation qui concerneront des fonctions classiques difficilement intégrables sous les précédentes versions de circuits.

Décodage d'adresses

Tous les systèmes à microprocesseurs nécessitent une logique de décodage d'adresses pour l'accès aux mémoires et aux entrées-sorties. Cette logique devient rapidement une glu lorsque l'espace d'adressage augmente, lorsque plusieurs boîtiers doivent éventuellement être activés pour une même position ou lorsque les différents espaces mémoires sont distincts (caches, E/S non projetées en mémoire, etc.).

De plus, lorsque des mémoires dynamiques sont présentes, une logique de multiplexage et de rafraîchissement supplémentaire est à prévoir.

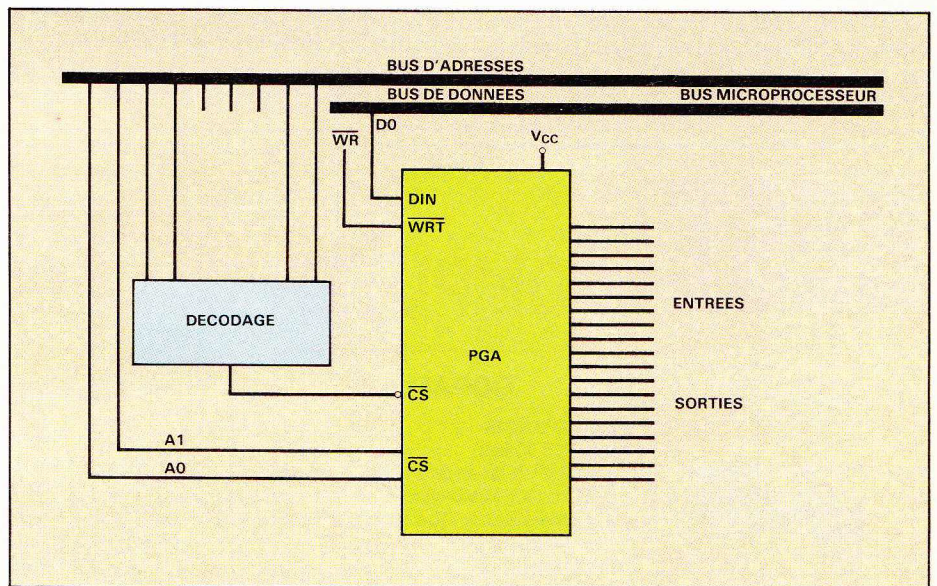


Fig. 11. – Configuration par un processeur.

La série 3000 dispose de particularités spécialement adaptées à ce genre d'applications. Une judicieuse combinaison des cellules logiques pour le prédécodage des signaux et lignes d'adresses, puis une utilisation des ET câblés et connexions trois états sur les grandes lignes pour la sélection des boîtiers suffisent à réaliser les décodages les plus acrobatiques.

Une utilisation optimale des fonctionnalités décrites plus haut permet généralement de gagner en temps de propagation ou au pire des cas d'obtenir un temps équivalent à celui d'une logique réalisée à l'aide d'éléments discrets.

Multiplexage et comptage

Dans ce cas encore, où par définition le multiplexage est la transmission de données de sources diverses vers une même destination, les tampons trois états pilotables par les cellules *via* les grandes lignes internes du PGA offrent toutes les fonctionnalités nécessaires à de telles applications.

Pour les fonctions de multiplexage, le gain espace sur une carte est considérable surtout lorsqu'il s'agit de structures du type bus.

Concernant le comptage binaire (le plus critique) sur un grand nombre de bits, les

implémentations les plus courantes se sont faites dans des circuits SSI ou MSI ou dans des PLD conventionnels. L'inconvénient des uns se situe au niveau du nombre de variables d'entrée des portes logiques : pour un compteur sur 10 bits par exemple, le positionnement du dernier étage est fonction de l'état des neuf précédents et nécessite donc une fonction logique à neuf variables d'entrée. La série des PGA 3000 permet de créer des fonctions logiques à 6 ou 7 variables par cellule (voir plus haut sans pour cela nécessiter des structures trop complexes ni d'avoir des temps de propagation prohibitifs.

L'inconvénient majeur des implémentations sous PLD reste le nombre de bascules disponibles (10 pour un circuit 24 broches); un PGA 8 x 8 comporte 64 éléments de stockage indépendants sans compter ceux des blocs d'entrée-sortie.

Interfaces diverses à des bus processeurs

Une quelconque interface connectée à un bus processeur nécessite des registres de stockage des données en entrée ou de données à transmettre au périphérique. Le choix habituel se porte sur des circuits intégrant des fonctions classiques, et le principal problème se situe au niveau des fonctionnalités figées de ce genre d'interface et

Illustration Colin-Thibert



des registres de données et d'état (8 bits en général) qui occupent le plus souvent une place non négligeable.

Dans le cas des PLD, les registres se trouvent presque toujours à la sortie du réseau et/ou on préfère faire gérer au PLD les diverses logiques de contrôle et effectuer les stockages sur un registre externe.

En utilisant un PGA, il est plus aisé d'enregistrer des données en entrée ou en sortie car les éléments de stockage se trouvent dans les blocs d'entrées-sorties et leur gestion peut se faire aussi bien par programmation que par une structure logique bâtie à partir des cellules configurables.

Il ne s'agissait pas ici de présenter les PGA comme la panacée à tout développe-

ment logi-matériel, mais plutôt de mettre en évidence un certain nombre de possibilités mises à disposition des concepteurs de systèmes. Il ne fait pas de doute que cette génération de circuits va bouleverser les techniques de construction de systèmes et les stratégies commerciales qui leur sont attachées. Nous aurons vraisemblablement l'occasion d'y revenir dans ces colonnes.

D'un point de vue commercial, c'est l'étendue des applications, les performances et le degré d'intégration, aussi bien que le prix de revient au développement et la facilité de mise en œuvre des PGA qui en constitueront les atouts essentiels.

J.-L. Leonetti

BEST COMPUTER

16, rue de Turenne
75004 PARIS
Tél. : 48.04.31.84
Métro St-Paul, Chemin

DISQUE DUR SEAGATE



20 Mo **1990F**
40 Mo **3990F**

COPAM COMPACT XT TURBO



- CPU 8088
- TURBO 6, 8, 10 MHz
- 512 K RAM + Horloge
- Interface Parallèle + série
- Lecteur 360 Ko
- Carte couleur graphique
- MS DOS 3.2 + manuel + Basic

3990F

COPAM AT

- CPU 80286
- 6, 8, 10 MHz
- 512 K (1 Mo sur carte Mère)
- Interface parallèle et série
- Sortier R, G, B, TTL et composite
- Lecteur 1,2 Mo
- Disque dur 20 Mo
- Clavier avec flèches séparés
- Moniteur monochrome
- MS DOS 3.2 + Manuel



12990F

DISQUETTES 5" 1/4

- 5" 1/4 DF/DD **3,40F**
- 5" 1/4 DF/DD
- Boîte plastique **4,20F**
- 5" 1/4 DF/DD
- pour AT **13,50F**

DISQUETTES 3" 1/2

- 3" 1/2 DF/DD **9,9F**
- 3" 1/2 DF/DD
- Boîte plastique **11F**
- 3" 1/2 DF/DD
- pour PS/2 **49F**

MONITEURS



- GOLDSTAR Monochrome **649F**
- SAMSUNG TTL **790F**
- THOMSON couleur **2290F**

LISTING

- 240 x 11" 2500 feuilles **149F**
- 380 x 11" 2500 feuilles **249F**

DIVERS

- STEAMER AT/XT 40 Mo **3990F**
- Table traçante A3 6 coul. **3990F**
- Câble parallèle, blindé **69F**
- Carte parallèle Centronics **230F**
- Boîtes de rangement
- 5" 1/4 130 disquettes TH 177 **89F**
- KORTEX KX-TEL **1590F**

IMPRIMANTE

- CITIZEN 120 D - 120 CPS **1590F**



- EPSON LX 800 - 180 CPS **1990F**

Le meilleur rapport Qualité/Prix
ATARI 520 STF
2890 F TTC

Tous nos prix sont HT

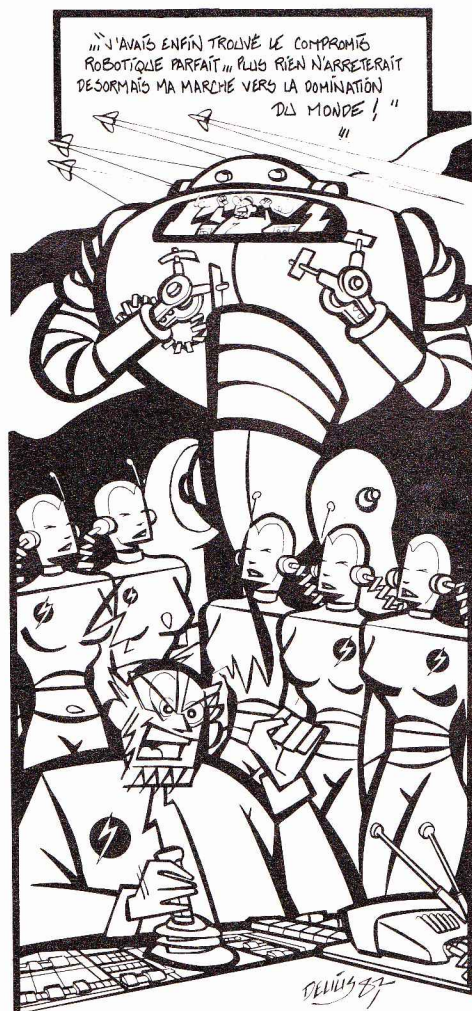
LA ROBOTIQUE: UN MYTHE DECISIF

Homme machine ou machine homme, le robot a depuis toujours relevé du mythe. Au départ, simple copie approximative d'êtres vivants, le robot d'aujourd'hui accède à l'intelligence, mais y a perdu une partie de son identité, c'est-à-dire sa forme.

Les premiers robots, ou plutôt automates, furent basés sur un concept anthropomorphique et conservaient un aspect magique, rituel. On retrouve ainsi dans l'Égypte ancienne ou dans l'antiquité grecque des mécanismes fonctionnant à l'eau ou au mercure, qui n'ont d'ailleurs laissé que des traces écrites. Ces statues plus ou moins mouvantes, aux têtes animées qui feraient sourire aujourd'hui, mais au degré d'évolution et de culture technique en rapport avec celui des sociétés concernées qui éveillent peut-être plus d'émerveillement ou de crainte qu'un robot monteur de chez Renault en 1987. Au-delà de ces réalisations frustes, l'horlogerie au Moyen Âge va permettre de faire naître des machines qui se meuvent plus délicatement et exécutent quelques gestes. Cette vélocité, alliée à une apparence toujours plus proche de la réalité, forcent l'admiration. Les ressorts et les vérins autorisent alors des opérations confondantes de réalisme pour l'époque. Cette philosophie de l'automate imitateur, simple défi des hommes devant l'Éternel, et finalement d'un usage limité va se perpétuer jusqu'au début du vingtième siècle. C'est durant cette période qui s'étend sur près de quatre siècles que de grands noms vont s'affirmer dans cette « science ». Au début, quelques-uns y laissent leur tête car on ne plaisante pas avec la magie. Les références du genre s'appellent Léonard de Vinci, Vaucanson, F. von Klaus...

Le robot esclave

Le vingtième siècle, qui invente le mot « Robot », matérialise un profond changement dans la philosophie des automates, et ce grâce à l'apport de la technique. Le dé-



but du siècle voit les machines cybernétiques fleurir. Plus rien à voir avec les automates joueurs de flûte, écrivains, oiseaux battant des ailes. La nouvelle génération intègre un nouveau concept de dépendance à l'environnement et de captage d'informations modifiant leur comportement, ce qui n'est possible que grâce à l'électricité et surtout l'électronique même un peu rustique. Ainsi, une machine cybernétique (on leur donnait souvent à l'époque des apparences ou des noms d'animaux) peut être attirée par la lumière, le bruit, la chaleur ou au contraire les fuir, se diriger dans un labyrinthe... Certes la production de machines cybernétiques va bon train, jusqu'au milieu du vingtième siècle. Peu à peu, il devient inutile de prouver que le robot (terme d'origine tchèque) dispute à l'homme des lambeaux d'intelligence, considérée alors comme manière autonome de se déterminer. Signe des temps et d'un rationalisme qui domine la science et la production, les robots vont enfin devenir plus utiles. A toutes les époques, les évolutions scientifiques et culturelles dans un secteur donné ne forment pas une sorte de génération spontanée, mais sont dues au développement global d'un tissu culturel qui les induit. C'est ce qui explique le fait que certaines inventions sont faites quasi simultanément aux antipodes sans que les chercheurs n'aient eu le moindre contact. La cybernétique est née d'une modélisation qui va permettre, avec l'aide de l'électronique, de l'électromécanique, la naissance de robots esclaves destinés à la production ou à diverses tâches utiles.

Le robot au travail

Qu'il soit à l'usine ou à la maison, le robot d'aujourd'hui n'a plus forme humaine,

mais n'est qu'un assemblage de capteurs et de mécanismes destinés à remplir une action plus ou moins précise, complexe, voire dangereuse mais qui détermine intégralement sa forme. C'est l'un des avantages du robot de pouvoir fonctionner en atmosphère polluée, radioactive, etc., et de pouvoir répéter indéfiniment le même geste avec une précision constante, sans états d'âme, lassitudes, mauvais rhume... Et pour cette tâche, la forme humaine n'est pas forcément la mieux adaptée à une activité de précision. Pour souder une pièce, il n'est pas nécessaire de créer la reproduction métallique d'un homme assis tenant une soudeuse à arc, un simple bras terminé par une électrode fait l'affaire. Pour que le bras d'un robot saisisse avec un maximum de précision une pièce disposée de manière aléatoire sur un tapis roulant, il vaudra mieux que les capteurs remplaçant notre œil et dirigeant cette action soient à hauteur de la pince, que placés à un mètre au-dessus comme le sont des yeux d'un humain par rapport à ses mains.

Les robots néfastes ?

Derrière les robots industriels couve une grande polémique et les avis sont partagés quant à l'intérêt de nos petits frères de métal. Le principal grief qui leur est fait revient comme un leitmotiv et concerne bien entendu le chômage, un robot étant capable dans une entreprise de remplacer plusieurs ouvriers. La réalité est plus complexe, car à supposer que toutes les entreprises soient pleines de robots, il ne fait aucun doute que la quantité de travail restante se trouverait peu à peu répartie entre les individus aptes à travailler, autrement dit qu'à terme, chacun travaillerait beaucoup moins. Ce propos n'est pas utopique, le même problème s'est posé il y a un siècle à l'arrivée massive des machines industrielles. Après une phase d'adaptation, les machines ont continué leur travail et les hommes en ont fait d'autres, moins durs. Comme disent les économistes, l'une des causes du chômage (il y en a malheureusement d'autres) est la désadaptation structurelle et passagère entre de nouveaux moyens de production et des habitudes anciennes. Donc, à terme, ces petites mécaniques devraient faire notre bonheur.

Les robots à la maison

Si l'on utilise la classique définition du robot, machine destinée à remplacer l'homme dans l'exécution d'une action, une machine à laver est un robot. En fait, beaucoup de chroniqueurs et chercheurs ont vu dans la vie domestique le dernier bastion du robot aux allures humaines, portant



l'apéritif au maître de maison, et lui débitant quelques fadaises sur le contenu du réfrigérateur ou la température intérieure. Là encore, la réalité est bien différente. Il ne fait aucun doute, à l'aube de la dernière décennie du vingtième siècle, que la seule intelligence domestique (excepté la maîtresse de maison) sera omniprésente et contrôlera tous les appareils de la maison sans aucun mouvement ni propos aussi spectaculaire qu'inutile. Ce sera une simple centrale électronique, raccordée à tout ce qui pourra l'être (four, machine à laver, portes, fenêtres, chaîne Hi-Fi, téléviseurs...), à condition de tomber d'accord sur un moyen standard de réaliser l'opération, le fameux bus domotique.

Des robots pour se divertir

Il reste néanmoins un domaine où le robot conserve son apparence humaine, le divertissement. Et il y en a pour tous les

goûts. Cela commence au robot jouet, qui, comme les animaux cybernétiques d'il y a quelques décennies, marche ou roule, contourne les obstacles et parle. Mais au-delà de ces figurines animées, certains ont perdu leur identité. Les joueurs d'échecs ne ressemblent plus au spectaculaire automate du baron von Kempelen, qui cachait d'ailleurs un humain dans son socle, mais sont constitués d'un simple bras solidaire du damier, et mu par un ordinateur situé, lui aussi, dans le socle. Point de magie, mais une simple utilisation de la technologie de l'instant. Quant à la représentation anthropomorphe, elle se porte merveilleusement bien sur les grands écrans. De *la Guerre des étoiles* à *Terminator*, les robots et androïdes mécanisés en tout genre réveillent un mythe indestructible. Les parcs d'attraction sont également un de leur domaine de prédilection, en général plus bon enfant. De Mirapolis à Disneyland, ces mécaniques parfois géantes et toujours ingénieuses forcent l'admiration des enfants, même si leur intelligence est souvent comptée.

De l'organe à l'homme

Dans les domaines « sérieux », la victoire de la fonction sur la forme n'est cependant pas aussi définitive qu'il y paraît. Si l'on considère l'évolution des robots industriels en une vingtaine d'années, on constate que ceux-ci sont passés par différentes phases d'une montée en intelligence toute relative. Les premiers robots industriels ne réalisaient en effet que des manipulations selon des séquences figées. Les séquences variables, c'est-à-dire intégrant des variations suivant les cas, ne vinrent que plus tard, avec en prime la liaison à des systèmes informatiques. Ce n'est que très récemment que les robots sont devenus un tant soit peu intelligents. Les derniers modèles sont munis de capteurs qui leur permettent de saisir certaines informations sur l'environnement pour se déterminer dans leur action ou pour cumuler ces informations de manière à modifier leur comportement. Là encore, c'est la technologie qui domine. Il ne fait aucun doute que le prix des composants qui baisse, leur miniaturisation, la puissance des processeurs et donc des logiciels, et l'acuité croissante des capteurs permettront dans les décennies à venir d'affiner les processus de capture et de traitement des informations, c'est-à-dire de réaliser des robots de plus en plus intelligents. Et il arrivera bien un jour où un industriel fabriquant divers types de ces machines souhaitera, pour des raisons d'économie toujours d'actualité, ne réaliser qu'un seul modèle de robot parfaitement polyvalent. Il y a alors gros à parier qu'il ressemblera à un homme !

A. Cappucio

PLEIN FEU SUR LES NOUVELLES

stair



NB 24-10 / 216 CPS / 80 COLONNES



NR 10/15 / 240 CPS / 80-136 COLONNES

HENGSTLER

DÉPARTEMENT IMPRIMANTES ET PÉRIPHÉRIQUES

B.P. 71 - 93602 Aulnay-sous-Bois Cedex - FRANCE

Téléphone : (1) 48.66.22.90 - Télex : hcn 212486 F

Nom _____ Prénom _____

Société _____

Rue _____

Ville _____ Tél. : _____

MS 12/87

ROBOTISATION ET PMI: LE C

Les robots ont quitté le domaine de la science-fiction pour venir peupler les ateliers, lorsqu'ils décident de robotiser la fabrication, est l'augmentation des rendements, duits finis, l'amélioration des conditions de travail, l'économie de main-d'œuvre. être considérés : l'organisation générale de l'unité de production, la formation du per

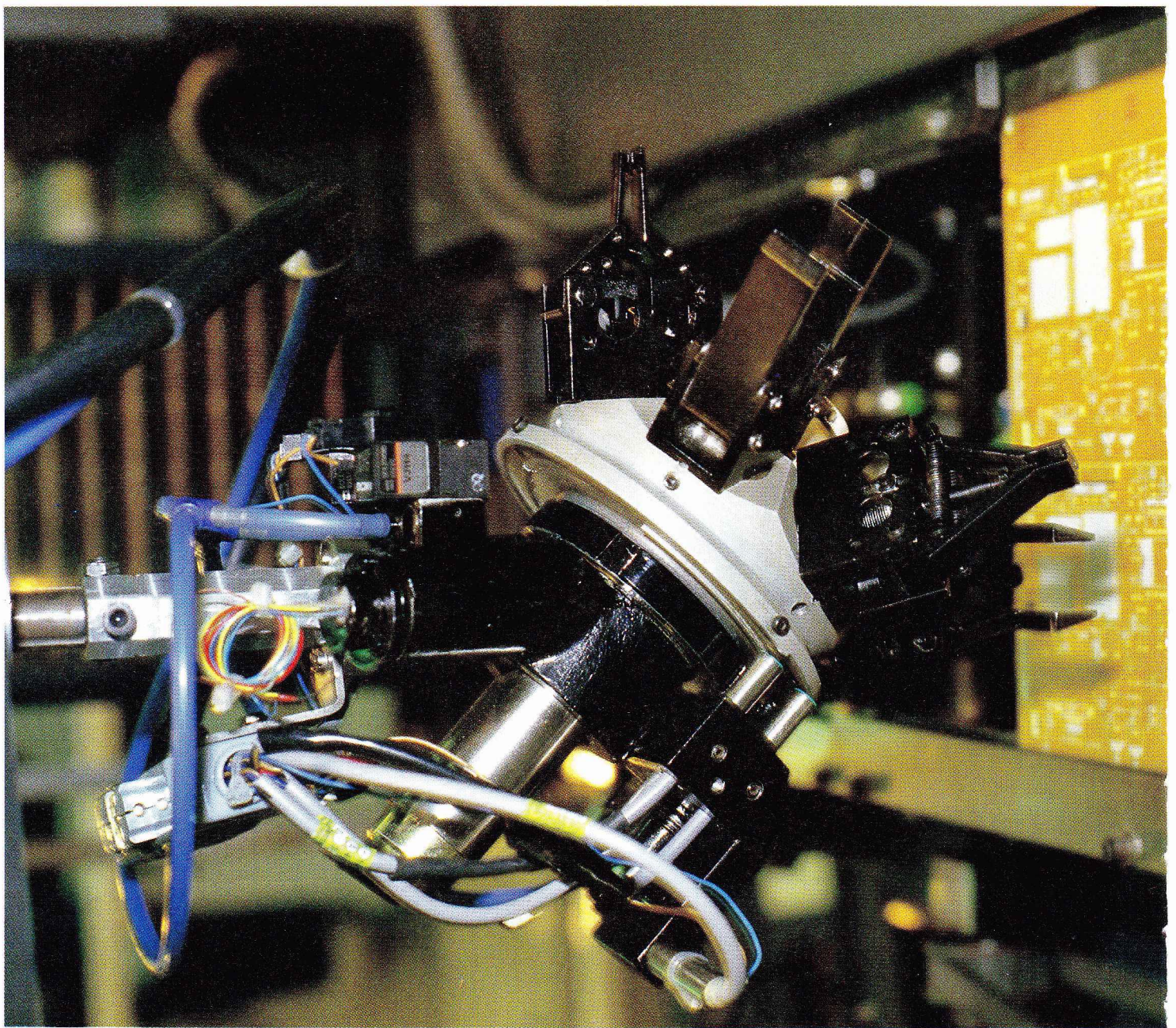
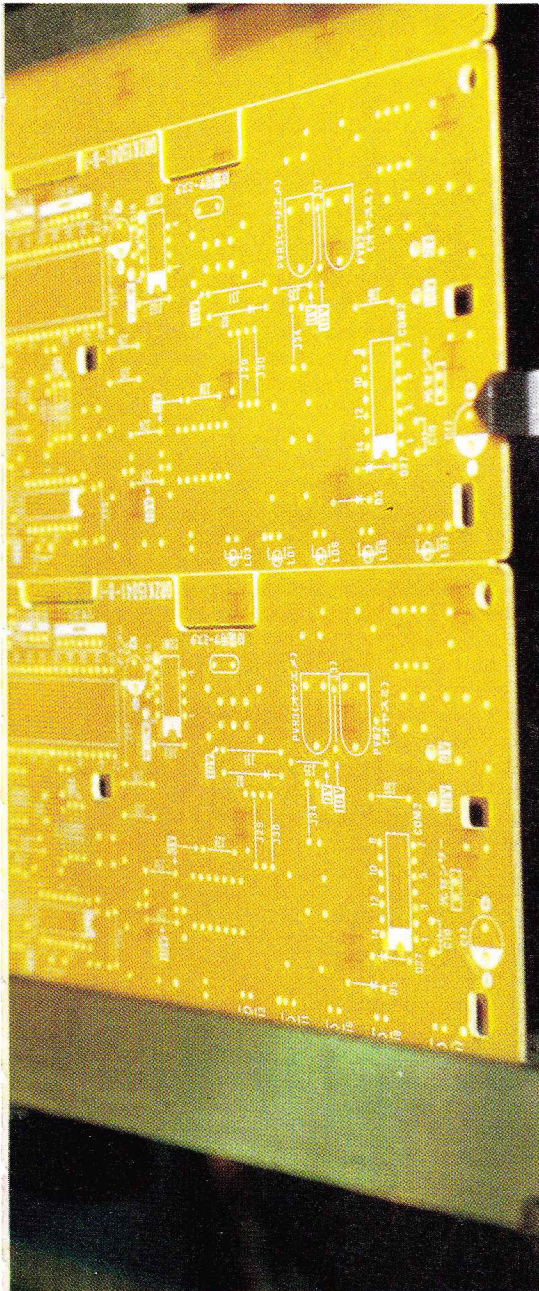


Photo © Benoit Decont/REA

HOIX DE LA COMPÉTITIVITÉ

les usines, les unités de production. Si la principale motivation des industriels, d'autres facteurs interviennent dans leur décision : la qualité et la régularité des pro-
 Mais, avant de décider de robotiser son entreprise, de nombreux paramètres doivent
 sonnel, l'investissement en matériel et logiciel...



Transporter, charger, décharger, souder, visser, coller, peindre... le marché potentiel des robots est considérable, et pourtant leur implantation, surtout en Europe, est encore relativement faible. Malgré une forte croissance de l'équipement robotique, au cours de ces dernières années, la France reste encore, dans ce domaine, à la traîne des pays développés, notamment l'Allemagne fédérale, les Etats-Unis et, évidemment, le Japon, qui fait figure d'exemple en la matière (tableau 1).

Nombreux sont ceux qui ont pu admirer les performances extraordinaires des robots exposés à Tsukuba. Outre cet aspect spectaculaire, de nombreuses usines nippones sont entièrement robotisées et tournent vingt-quatre heures sur vingt-quatre, sans la présence d'hommes *in situ*. Les premières applications industrielles de la robotique datent des années soixante. L'automobile, avec 56 %, et l'électronique avec 24 %, sont les domaines les plus concernés, aujourd'hui, par la robotisation.

Un robot n'est théoriquement limité ni à une tâche particulière ni à un produit ou à un secteur industriel. Pour la petite entreprise, qui travaille sur de faibles séries, le robot, par sa flexibilité, est la seule vraie alternative au travail manuel (encadré 1).

La robotisation permet d'abaisser les prix de revient des produits, d'améliorer la qualité, de maîtriser les délais de fabrication. Si elles veulent rester compétitives, il est temps que les entreprises françaises prennent conscience de l'enjeu que constituera la robotique dans les années à venir.

Le robot n'est qu'un outil

Jusqu'à la fin des années soixante-dix, les robots industriels ne jouaient qu'un rôle d'automate mécanique programmable. Depuis 1980, la robotique est associée à la

◀ Exporobots 87 à Paris : robot multifonction d'assemblage A 4010 S.

productique, c'est-à-dire l'ensemble des techniques d'automatisation de production : la conception, la fabrication et la gestion de production (fig. 1). Philippe Coiffet, auteur d'une série de traités sur la robotique, constate : « On a pris conscience que le robot n'était qu'un outil de l'automatisation parmi d'autres, même s'il est récent et qu'on ne doit plus le considérer comme une machine isolée à tout faire, mais bien comme un élément à intégrer à un système de production » (voir bibliographie).

Chacun des sous-systèmes constituant la productique remplit un certain nombre de fonctions automatisables grâce à des modèles mathématiques et des robots industriels. Ces différentes phases ont tout intérêt à être associées dans un système informatique global, depuis la conception, en amont, jusqu'à la gestion de production, en aval.

La robotique est étroitement liée à l'informatisation dont elle est destinée à devenir l'un des principaux débouchés dans la prochaine décennie. Ses progrès accompagnent ceux de la CAO (conception), FAO (fabrication), GPAO (gestion de production) et, d'une façon générale, de toutes les techniques assistées par ordinateur, ainsi qu'à ceux de l'électronique (mémoires plus vastes, vitesses de traitement supérieures), des techniques de l'intelligence artificielle, des systèmes de vision et autres capteurs.

Il existe une grande variété de robots industriels et non industriels – c'est-à-dire intervenant dans les milieux naturels (agriculture, océan, espace, mines...) ou dans la vie courante (robots domestiques, pédagogiques, de service). Nous nous intéresserons ici exclusivement à la première catégorie, qui représente déjà une assez grande diversité.

Les robots peuvent être classés par leurs fonctions. C'est ainsi que l'on distingue les robots d'assemblage, de montage de composants en surface, de chargement et déchargement, de manipulation d'outils, de projection (peinture, colle, etc.), de soudage par point ou à l'arc, de découpage et usinage, de moulage, de traitement thermique, de finition... (tableau 2).

L'Association française de robotique in-

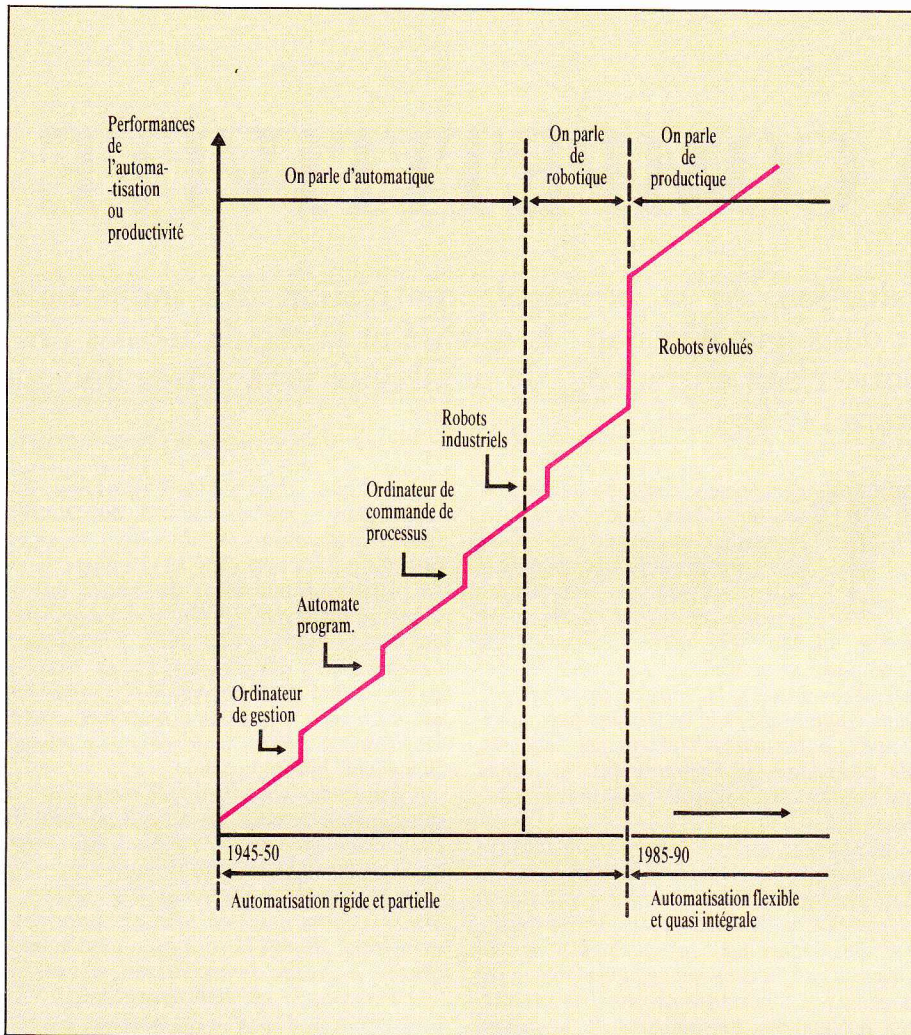


Fig. 1. - Evolution du concept de l'automatisation vers la robotique et la productive. D'après Philippe Coiffet (1).

dustrielle (AFRI) a établi une classification en fonction du degré d'évolution des robots, depuis les manipulateurs jusqu'aux robots dits « intelligents ». Tous ces dispositifs ont en commun le fait d'être adaptables à de multiples applications et de comporter un moyen de préhension ou un outil pour effectuer des tâches à caractère industriel, y compris le contrôle (encadré 2).

La première idée qui vient à l'esprit lorsqu'on parle de robotisation est la suppression d'emplois. L'économie de main-d'œuvre est, en effet, une raison importante pour prendre la décision de robotiser. Une étude portant sur 200 entreprises japonaises robotisées indique que plus des deux tiers ont réduit de 40 % le nombre de travailleurs. Mais il faut aussi tenir compte du fait que la robotisation implique une élévation de la qualification : les ouvriers sont remplacés par des ingénieurs et des techniciens spécialisés.

Préparation et formation du personnel

Pour beaucoup d'entre nous, les robots représentent encore mystère et inquiétude. La robotisation nécessite donc une préparation et une formation du personnel. Celle-ci ne sera pas restreinte à l'utilisation exclusive du robot, mais elle englobera toute l'organisation du travail dans l'unité de production. Il est important, tout d'abord, que le robot soit perçu par le personnel comme une amélioration de ses conditions de travail. La robotisation permet, en effet, d'automatiser les tâches répétitives, ennuyeuses, pénibles, dangereuses (industrie chimique ou nucléaire, par exemple). Elle supprime donc un certain nombre de risques. Mais il ne faut pas oublier qu'elle peut en créer d'autres : des accidents peuvent être causés par une collision entre un homme et un robot, par la projection de pièces ou de matière, par le coincement dans un robot. La formation devra aussi comprendre cet aspect de la robotisation.

La décision de robotiser

Contrairement à une opinion répandue, la robotisation ne s'adresse pas exclusivement aux très grandes industries (automobile, électronique...). Elle peut aussi bien intéresser les petites et moyennes entreprises qui cherchent à améliorer leur productivité et maintenir un haut niveau de qualité. Benoît Weymuller, sous-directeur de Machines et Automatismes à la direction générale de l'industrie du ministère de l'Industrie, déplore que « le démarrage de la robotique chez les PMI n'a pas encore réelle-

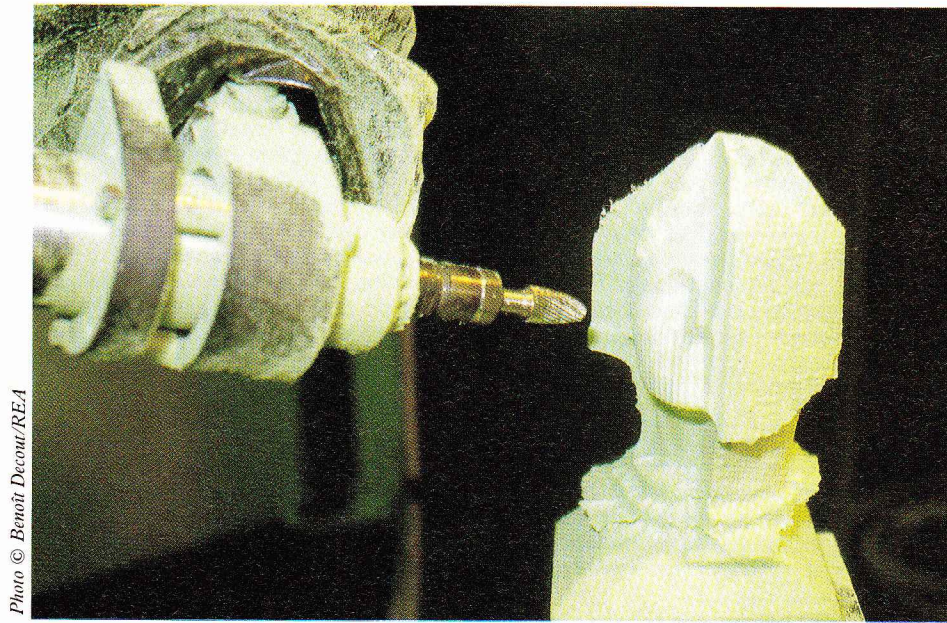


Photo © Benoît Decout/REA

Système Vision 3D associé à la RATP.

Tableau 1 Estimation du nombre de robots dans sept pays industrialisés

| | Parc total de robots 1984-1985 | Robots de soudage | Robots de chargement/déchargement | Estimation du total des robots en 1990 |
|--------------------|--------------------------------|-------------------|-----------------------------------|--|
| France | 2 300-2 700 | 1 050 | 600 | 6 700 |
| Etats-Unis | 13 000 | 4 000 | | 31 500 |
| Japon | 21 000-25 000 | 13 000 | | 32 000 |
| Allemagne fédérale | 6 600 | 3 200 | 1 000 | 19 000 |
| Grande-Bretagne | 2 600 | 850 | | 9 000 |
| Suède | 2 400 | | | |
| Italie | 2 600 | | | |

Source : Bail Equipement.

Tableau 2 Marché mondial de la robotique : perspectives

| Segments | 1986 | 1987 | 1988 | 1989 | 1990 | 1991 |
|---------------------|-----------|-------|-------|-------|-------|-------|
| Régions | | | | | | |
| USA | 600 M\$ | 660 | 825 | 1 030 | 1 250 | 1 530 |
| Japon | 780 M\$ | 910 | 1 025 | 1 150 | 1 300 | 1 550 |
| Europe | 450 M\$ | 550 | 650 | 800 | 950 | 1 100 |
| Autres | 50 M\$ | 60 | 100 | 100 | 150 | 150 |
| Applications | | | | | | |
| Assemblage | 390 M\$ | 490 | 645 | 790 | 970 | 1 170 |
| Chgt/Manut. | 430 M\$ | 530 | 645 | 760 | 890 | 1 050 |
| Peinture | 190 M\$ | 210 | 210 | 210 | 220 | 260 |
| Soud. point | 400 M\$ | 390 | 420 | 440 | 490 | 560 |
| Soud. arc | 250 M\$ | 315 | 405 | 560 | 700 | 840 |
| Usin./autres | 220 M\$ | 245 | 275 | 320 | 380 | 450 |
| Total | 1 880 M\$ | 2 180 | 2 600 | 3 080 | 3 650 | 4 330 |

Source : Dataquest.

ment eu lieu. Ce retard singularise la France par rapport à ce que l'on observe dans les autres pays développés, comme l'Allemagne. Ses conséquences, si ce retard se poursuivait, seraient graves pour la compétitivité des industries manufacturières ».

Mais trop souvent, faute de temps, d'informations objectives et de moyens, ces PMI diffèrent une décision d'automatisation au risque d'être distancées par leurs concurrents. C'est pour répondre à ces interrogations qu'une association s'est créée dans la ville nouvelle d'Evry (encadré 3).

Un robot ne s'achète pas sur catalogue. Comme nous l'avons vu, c'est un élément qui doit s'insérer à la fois dans une structure matérielle et logicielle, et dans un contexte social. C'est l'ensemble qui va résoudre le cas particulier de l'industriel. Celui-ci cherche généralement une solution « clés en main », ou tout au moins une pré-étude sérieuse.

En outre, le coût des robots est encore très élevé par rapport à celui de l'informatique traditionnelle. C'est pourquoi les chefs d'entreprise choisissent généralement d'automatiser d'abord les outils comptables et financiers, avant de se pencher sur les moyens de production. Il est difficile de donner une évaluation du coût moyen d'un

robot, car cette valeur varie beaucoup en fonction des tâches que peut accomplir ce dernier. Elle se situe le plus souvent entre 200 et 500 000 F. L'achat d'un robot constitue donc un investissement important que beaucoup de PMI hésitent à faire, car souvent elles ignorent l'existence de solutions adaptées à leur cas particulier (encadré 4).

La robotisation peut être considérée de différents points de vue : elle répond à un besoin économique urgent face à la compétition internationale ; elle est un moyen de modernisation de l'appareil de production ; elle peut devenir une plate-forme de collaboration entre la recherche et l'industrie. Son apport est donc perçu différemment par les différentes directions de l'entreprise : le service production s'attachera essentiellement aux performances et à la flexibilité qu'apporte le robot ; le bureau d'études s'intéressera au progrès technologique qu'il représente ; quant au directeur financier, il y verra surtout un investissement important et en attendra des retours d'investissements. Les avis peuvent être contradictoires.

L'industriel ou le chef d'entreprise devra examiner ces différents points de vue. Le problème est généralement très complexe,

Encadré 1

LE ROBOT IDEAL EN 14 POINTS



L'Américain Joseph Engelberger, l'un des pères de la robotique moderne, a défini le cahier des charges du robot idéal, en s'inspirant de la pièce de l'écrivain tchèque Karel Capek, l'inventeur du mot « robot ». Voici les quatorze attributs de ce robot idéal :

1° Une main capable d'aggriper, de tenir, puis de relâcher une pièce.

2° Un bras capable de mouvoir la main en trois dimensions.

3° Un poignet au bout du bras avec des articulations, permettant à l'ensemble main-poignet de se mouvoir dans n'importe quelle direction à l'intérieur de la zone de travail.

4° Des muscles suffisants pour soulever une pièce de 225 kg.

5° La capacité de reprendre une même position avec une marge d'erreur de 0,3 mm.

6° Des contrôles manuels permettant à une personne d'utiliser toutes les fonctions des membres du robot.

7° Une mémoire intégrée capable d'apprendre les instructions d'un professeur humain.

8° Des systèmes automatiques permettant à la mémoire de contrôler les opérations en l'absence du professeur humain.

9° Une vitesse d'exécution au moins égale à celle d'un humain.

10° Une banque de programmes, permettant au robot de se ramener à des situations qu'il a déjà rencontrées dans le passé et donc de réagir de façon identique.

11° Des dispositifs de sécurité et des procédés de connexion avec l'usine, l'atelier ou avec la machine sur laquelle le robot travaille.

12° Une interface pour dialoguer avec un ordinateur.

13° Une fiabilité d'au moins 400 heures de temps moyen entre deux pannes, dans l'environnement de travail réel.

14° Une structure facilitant la maintenance, l'accès rapide aux composants et leur interchangeabilité en cas de panne, avec en plus des systèmes d'autodiagnostic.

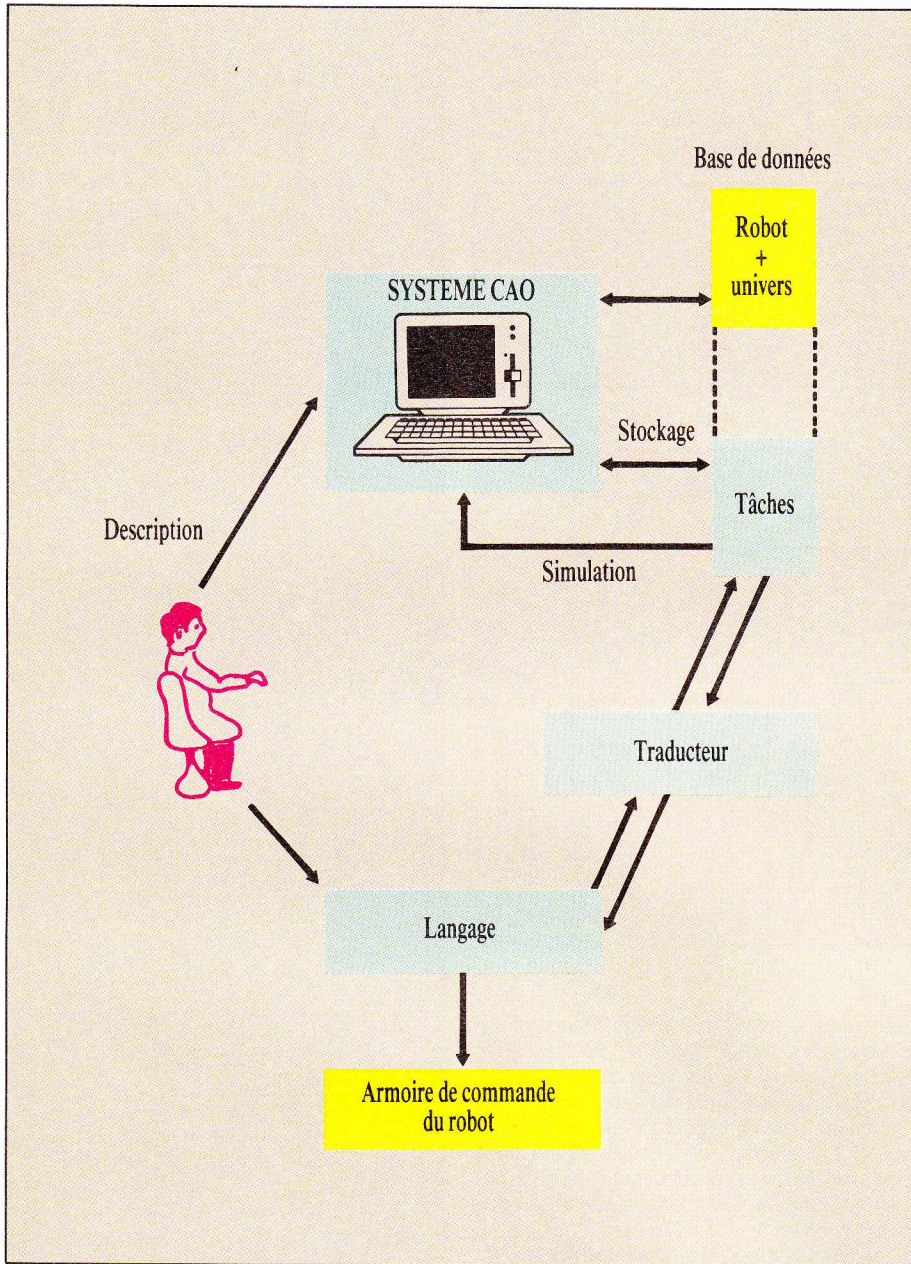


Fig. 2. – Schéma d'un système de CFAO pour la robotique. D'après Philippe Coiffet (1).

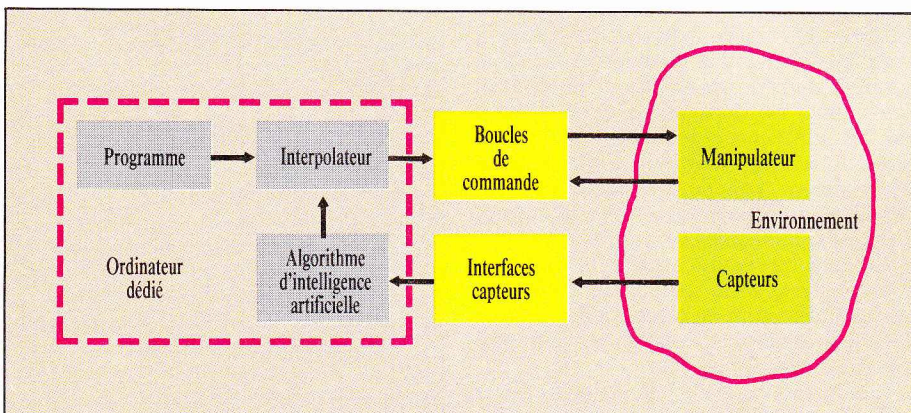


Fig. 3. – Schéma d'un système robotique intelligent. D'après Yoram Koren (3).

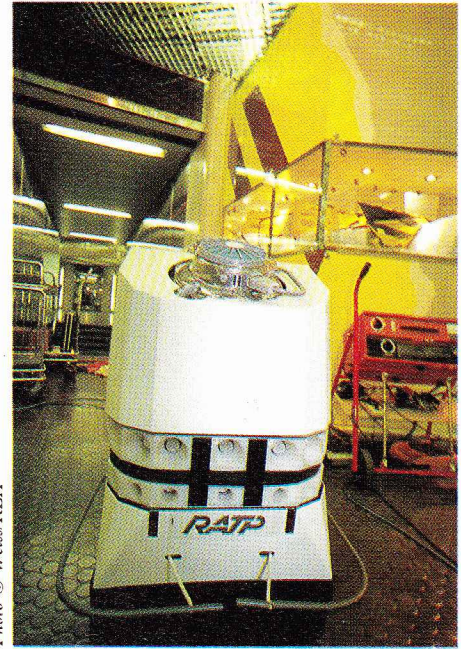


Photo © Weiss/REA

Midi, robot nettoyeur de la RATP.

et bien peu de PMI disposent en leur sein des sources d'expertise nécessaires pour le résoudre dans sa globalité. Les systèmes experts et les systèmes informatiques d'aide à la décision (SIAD) pourront lui apporter une aide utile. Le décideur pourra également avoir recours à un conseil extérieur : ce sont les sociétés de services et d'ingénierie en productique (SSIP) qui seront les partenaires de l'industriel dans son projet d'investissement.

Robotiser progressivement

Une fois prise la décision de robotiser, il n'est pas nécessaire d'automatiser d'emblée toutes les tâches. La robotisation peut fort bien être progressive. Parfois le manipulateur peut suffire à résoudre un problème, plutôt qu'un robot surdimensionné.

Il est souvent possible de commencer par une petite cellule robotisée, comprenant, par exemple, le robot et son positionneur, un opérateur étant chargé de mettre les pièces sur le positionneur. Ainsi, un robot soudeur supprimera une opération délicate effectuée auparavant par un homme dont la tâche sera remplacée par une simple manutention. Les opérations de soudage sont, en effet, parmi celles qui se prêtent le mieux à la robotisation.

Dans un deuxième temps, la manutention peut aussi être automatisée par des robots supplémentaires venant alimenter les robots de soudage. La troisième étape consistera à remplacer les robots de soudage et d'alimentation par un seul robot qui aura pour fonction, à la fois, d'alimen-

Encadré 2

CLASSIFICATION ET TERMINOLOGIE DES ROBOTS SELON L'AFRI

L'Association française de robotique industrielle (AFRI) distingue quatre grandes classes de robots :

1° Manipulateur manuel : engin de manipulation motorisé, commandé par l'homme, ayant au moins quatre degrés de liberté.

Deux sous-catégories :

- Manipulateur d'assistance musculaire : la commande se fait par action directe sur la charge ou dans son voisinage (exemple : girafe de déplacement de charge ; exclusion : treuils, palans, chèvres, équilibreurs à ressorts, etc.).

- Télémanipulateurs : la commande se fait à distance à l'aide de leviers ou de boutons (exemples : manutention en ambiance dangereuse, forge, industrie nucléaire, subaquatique ; exclusion : engin de génie civil ou agricole, pont roulant, grue, treuil, chariot élévateur, chèvre, etc.)

2° Manipulateur automatique : engin de manipulation de deux axes et plus, non asservi, à cycle automatique.

Deux sous-catégories :

- Manipulateur automatique à séquence fixe : réglé mécaniquement par butées, cames et portant un organe terminal (exemples : machine à peindre, chargement/déchargement de presse, chargeur d'outils, etc. ; exclusion : transfert rigide ou convoyeur).

- Manipulateur automatique à séquences variables : programmable par l'utili-

sateur à l'aide de matrices à diodes, automate programmable, microprocesseur, et où chaque axe est réglable mécaniquement par butée ou came (exemples : chargement/déchargement de machine-outil, assemblage ; exclusion : machines-outils, transtockeurs, chariots automoteurs, etc.).

3° Robot programmable : manipulateur automatique de trois axes et plus, dont au moins deux axes sont programmables par apprentissage ou par langage symbolique.

Deux sous-catégories :

- Manipulateur automatique comportant moins de cinq axes programmables.

- Manipulateur automatique comportant cinq axes et plus programmables. (Exemples : robot vertical, horizontal, portique, scara, etc. ; exclusion : machines-outils à commande numérique, centres d'usinage, chariots autoguidés, tables traçantes, etc.).

4° Robot dit « intelligent » : manipulateur automatique programmable capable d'analyser les modifications de son environnement et de réagir en conséquence, à l'exclusion des modifications triviales données par les capteurs tout ou rien (exemple : manipulation avec reconnaissance de forme, assemblage avec contrôle d'effort, soudage avec suivi de joint, etc.).

ter le positionneur et de souder la pièce. Enfin, dans une quatrième phase, le changement d'outils et de programmes pourra se faire automatiquement, constituant une cellule flexible.

La décision de robotiser doit toujours être un compromis entre l'avantage qui peut être retiré d'une automatisation et le coût que représente cet équipement. Ainsi, il vaudra mieux, dans certains cas, continuer à effectuer certaines tâches manuellement : à l'entrée de la ligne robotisée, un poste d'intervention manuelle permettra d'éliminer les éléments sortant des normes prévues. Les contraintes les plus importantes concernent la mécanique, car c'est l'élément qui coûte le plus cher : une fois que la mécanique a été choisie, elle est difficilement modifiable, et toute modification nécessite une immobilisation de la ligne pendant quelque temps.

Même si la robotisation ne doit concerner, au début, que certaines tâches et un périmètre très limité, l'ensemble de l'organisation doit être reconsidéré en fonction de cette implantation.

La coordination des tâches automatisées

Les machines automatisées constituent souvent des « îlots d'automatisation ». Une robotisation bien intégrée implique une communication entre ces îlots. Les robots, qui peuvent être répartis dans un large périmètre à l'intérieur de l'usine, sont commandés par des ordinateurs. Pour qu'il puisse y avoir communication entre eux, et que les différentes tâches soient coordonnées, il faut relier les robots *via* un réseau local industriel.

Le réseau est un élément très important dans l'usine équipée par IRCI (*Ingénierie Recherche Conseil Informatique*). Il s'agit d'une chaîne d'emballage et d'expédition de matériel informatique (ordinateurs et périphériques). Cette unité de « pick and pack » reçoit en entrée les commandes clients et doit fournir en sortie des cartons contenant tout ou partie de la commande. Les consommables sont apportés par des chariots filoguidés qui alimentent les stocks et repartent avec les commandes.

L'informatique permet d'interfacer les différentes familles de machines automatisées : robots (bras articulés), moyens de transport (navettes, bras élévateurs, tapis...), automatismes. Les micro-ordinateurs commandant les autres parties de la chaîne permettent de répartir l'« intelligence » en tous les points de la ligne robotisée et de leur conférer une autonomie. Mais il est nécessaire de pouvoir dialoguer et de centraliser les informations.

La synchronisation des tâches et des mouvements (accélérations, freinages,



Photo © Maillat/REA

Robots utilisés dans l'industrie nucléaire (CEA Fontenay ISPN).

Encadré 3

UNE ACTION EXEMPLAIRE EN FAVEUR DE LA ROBOTISATION A EVRY

A 25 kilomètres de Paris, au centre de la ceinture technologique du Sud parisien formé par Saclay, Orsay, Palaiseau, Orly, Rungis, une ville nouvelle a décidé d'encourager le développement des technologies avancées, et en particulier de la robotique. Déjà plus d'une centaine de firmes des secteurs électronique, informatique, productique, robotique se sont installées dans ce technopôle axé sur l'informatique industrielle : Evry.

Afin de propager cette dynamique, issue d'entreprises de pointe, comme Snecma, Digital Equipment, IBM, Hewlett-Packard, Automation, pour n'en citer que quelques-unes, à tous les niveaux de l'économie régionale, des chefs d'entreprises, enseignants, chercheurs et élus locaux se sont réunis pour créer, en 1983, l'Association pour le développement à Evry du centre d'informatique et de robotique (ADECIR).

Vecteur du rapprochement de la recherche et de l'enseignement avec le milieu des entreprises, ADECIR informe, conseille, accompagne la PMI pour faciliter son accès aux automatismes et à la robotique. Centre de compétence au service de ses adhérents, elle met à la disposition des entreprises une capacité d'expertise, d'évaluation des projets, d'organisation des étapes d'un « plan productique » et des besoins en formation.

Association loi de 1901, indépendante des fournisseurs de matériels et de logiciels, l'ADECIR offre des informations et des services hors de tout contexte commercial. Elle propose notamment à ses adhérents une journée gratuite de diagnostic et un service d'informatique et de conseils techniques, ainsi qu'une assistance au montage de leurs dossiers d'aide financière. Elle propose également un cycle de formation à la productique comprenant la conception et la fabrication assistées par ordinateur, une large place étant réservée à la robotique industrielle (machines-outils à commande numérique, robots et manipulateurs, cellules et ateliers flexibles).

C.R.

changements de direction) est réalisée par des processus câblés ou précâblés qui peuvent être gérés par les automatismes.

L'informatique a une vue macroscopique de la ligne. Elle ne connaît que certains points, dits « points clés ». Toute charge apporte une information capitale à la gestion de la ligne. Sur une ligne, de petits articles placés dans des casiers plastifiés sont amenés au robot ; celui-ci pioche des éléments dans le casier et va les ranger dans un carton destiné à l'utilisateur. Par exemple, dans le cas d'une commande d'un micro-ordinateur, le robot prendra un écran, un clavier, une unité centrale et une documentation pour les mettre dans le carton. Cette activité, dite « pick and pack », est intelligente, d'une part par la nature de l'objet à prendre, d'autre part par la tâche de mise en boîte : le robot doit « connaître » la taille de l'objet à prendre afin de le ranger à tel endroit de la boîte pour optimiser le rangement (en haut ou en bas du carton).

L'architecture de cette ligne robotisée est une architecture dite de réseau, comportant un aspect fonctionnel (gestion de la ligne) et un aspect opérationnel (mécanique, robotique). Les commandes reçues de l'extérieur sont traduites à l'aide d'ordinateurs fonctionnels — des micro-ordinateurs industriels, c'est-à-dire d'une fabrication spéciale, adaptée aux conditions sévères qui règnent en milieu industriel (poussières, vibrations, chocs, températures extrêmes...).

Toutes les informations critiques relatives à la ligne sont centralisées dans une base de données. Ainsi, l'informatique assure l'interface entre la gestion de la commande et la ligne comprenant quatre robots à usage différent : deux robots prennent les objets, l'un les petits, l'autre les gros ; un robot est chargé de mettre les cartons en forme de palette pour les rendre transportables ; un autre a pour fonction de les dépalettiser.

Robotiser pour mieux gérer

L'automatisation des tâches implique la connaissance et le suivi, à tout moment, des matières premières, pièces détachées, produits semi-finis et finis. Tout au long de la chaîne, il est possible de savoir où se produisent les retards, quel poste est responsable de perte de matière première, etc. Ainsi, dans une issue de meuble, les chutes de bois et autres seront automatiquement pesées et récupérées. Cet avantage, qui n'est généralement pas considéré au moment où la décision de robotiser est prise, peut générer des bénéfices supplémentaires.

A l'usine de tubes-image couleur RTC (La Radiotechnique-Compélec) de Dreux, la robotisation a apporté des améliorations considérables tant du point de vue de la

qualité des produits que des conditions de travail. Elle s'adapte, en effet, aux impératifs industriels spécifiques aux tubes-image :

- changement de technologie des tubes, des procédés de fabrication et des contrôles ;

- fabrication simultanée de plusieurs formats de tubes et adaptation rapide des programmes de production pour répondre au mieux à la demande.

Les automates programmables qui équipent l'usine de Dreux sont doués de comportements « semi-intelligents », tels que la reconnaissance de types à l'aide de la lecture de codes à barres par exemple, la reconnaissance de forme, la possibilité de régulation des flux avec stockage tampon en fonction des quantités et des types...

A tous les niveaux de l'usine, des terminaux, des consoles de visualisation et des imprimantes permettent d'assurer la saisie en temps réel dans l'atelier, de dialoguer et d'exploiter les différents paramètres sous forme de tableaux ou de statistiques. Cette structure permet de responsabiliser le personnel au niveau de son poste de travail, de favoriser le travail d'équipe et de contrôler la qualité à tous les stades de la production. De plus, l'ensemble des informations, centralisées, permet aux responsables de production d'avoir des réactions appropriées et rapides sur tous les éléments de décision.

La flexibilité, un argument économique

Les robots industriels que l'on trouve le plus couramment dans les usines sont des manipulateurs automatiques programmables. Ils ont tendance à évoluer vers une plus grande polyvalence et adaptabilité, et deviennent de plus en plus « intelligents ».

La « flexibilité » est la capacité de produire en fonction d'une demande variable avec un minimum d'effort et de coût. Un même robot peut, en effet, être utilisé pour différentes tâches. Ainsi, certains paramètres (la taille par exemple) de l'objet à fabriquer peuvent varier ; le robot est également capable de prendre des outils différents. La flexibilité est l'argument économique le plus important ; grâce à cette qualité, le robot résiste à l'obsolescence. Une entreprise souhaitant fabriquer un nouveau produit, si elle possédait déjà un robot, n'a qu'à modifier le programme de celui-ci pour l'adapter aux caractéristiques de ce produit, ce qui limite considérablement le temps d'improductivité de l'atelier.

Les robots les plus perfectionnés ont un grand nombre de degrés de liberté (*) et sont donc capables d'accomplir des tâches très complexes et variées à l'aide du programme enregistré dans leur mémoire. Il suffit de modifier ce programme pour que

le robot exécute les tâches dans un ordre différent, adapté à un autre produit.

Le fait de réduire les temps de fabrication et de pouvoir utiliser le même robot pour différents types de produits permet de diminuer les en-cours et le stock des produits semi-finis, tout en réduisant les délais nécessaires entre la commande d'un produit et sa livraison.

Actuellement, un robot est dit « intelligent » lorsqu'il possède certains systèmes pour entrer en action ou modifier son comportement sans intervention humaine. Pour cela, il doit posséder des capteurs qui le renseignent sur le monde extérieur et sur sa position par rapport à celui-ci.

Les robots les plus évolués sont souvent équipés de systèmes de vision qui leur offrent la possibilité de « choisir » parmi un ensemble d'objets celui qui convient à la tâche qui leur a été demandée. Les systèmes les plus élaborés permettent la vision en trois dimensions et la reconnaissance d'objets en vrac (cf. article « La vision par ordinateur » dans *Micro-Systèmes* n° 71). D'autres, munis d'un faisceau laser, se contentent de lire des codes à barres pour déterminer le type de l'objet qui passe devant leur champ de vision.

Demain, les robots intelligents

A mesure qu'ils deviendront plus « intelligents », les robots ne seront plus seulement capables d'exécuter une tâche définie par une séquence d'instructions, mais d'accomplir celle-ci en fonction de circonstances qui peuvent être variables et en l'absence de toute intervention humaine. Pour cela, ils disposent non seulement de capteurs d'environnement, dits « extéroceptifs », comme la vision ou le toucher, mais de capteurs de l'état interne du robot, ou « proprioceptifs », qui permettent à celui-ci de contrôler sa trajectoire.

La robotique sera de plus en plus liée aux phases amont et aval de la fabrication, en particulier la conception et la fabrication assistées par ordinateur. Un système de CFAO appliqué à la robotique pourra à la fois produire des programmes et simuler leur exécution (fig. 2). Selon P. Coiffet, « vers 1990, 300 000 robots seront opérationnels dans le monde, dont un quart dotés de systèmes de vision et un quart reliés à un système de CFAO ».

Les robots sont actuellement capables de mouvement, de locomotion, parfois de perception. Demain, ils pourront également, à partir de l'information perçue grâce à leurs capteurs, extraire l'information pertinente, l'interpréter et prendre une décision. Cette dernière fonction implique la mise en œuvre de techniques d'intelligence artificielle (fig. 3).

Les systèmes experts permettront la supervision « intelligente » de processus industriels où il est nécessaire d'acquérir les informations des capteurs en temps réel et de dialoguer avec les routines de contrôle et commande. La firme grenobloise ITMI a développé un robot intelligent, piloté par un système expert et doté d'un système de vision, qui est capable d'assembler des pièces disposées en vrac dans le champ de la caméra de vision.

Les PMI commencent à être sensibilisées à l'enjeu que représente la productivité et à la nécessité de disposer des équipements de haute technologie que sont les ro-

bots, pour faire face à la concurrence internationale. Par ailleurs, les robots connaissent trois axes d'évolution qui tendent à les mettre à la portée des PME/PMI : la diminution des prix des matériels, l'accroissement de flexibilité et le développement d'applications orientées vers cette catégorie d'utilisateurs.

Claire Rémy

(*) Degré de liberté : c'est une direction dans l'espace que peut prendre un axe, ou élément articulé, indépendamment des autres. A chaque axe correspond donc un nouveau degré de liberté. Les robots les plus sophistiqués en possèdent une dizaine.

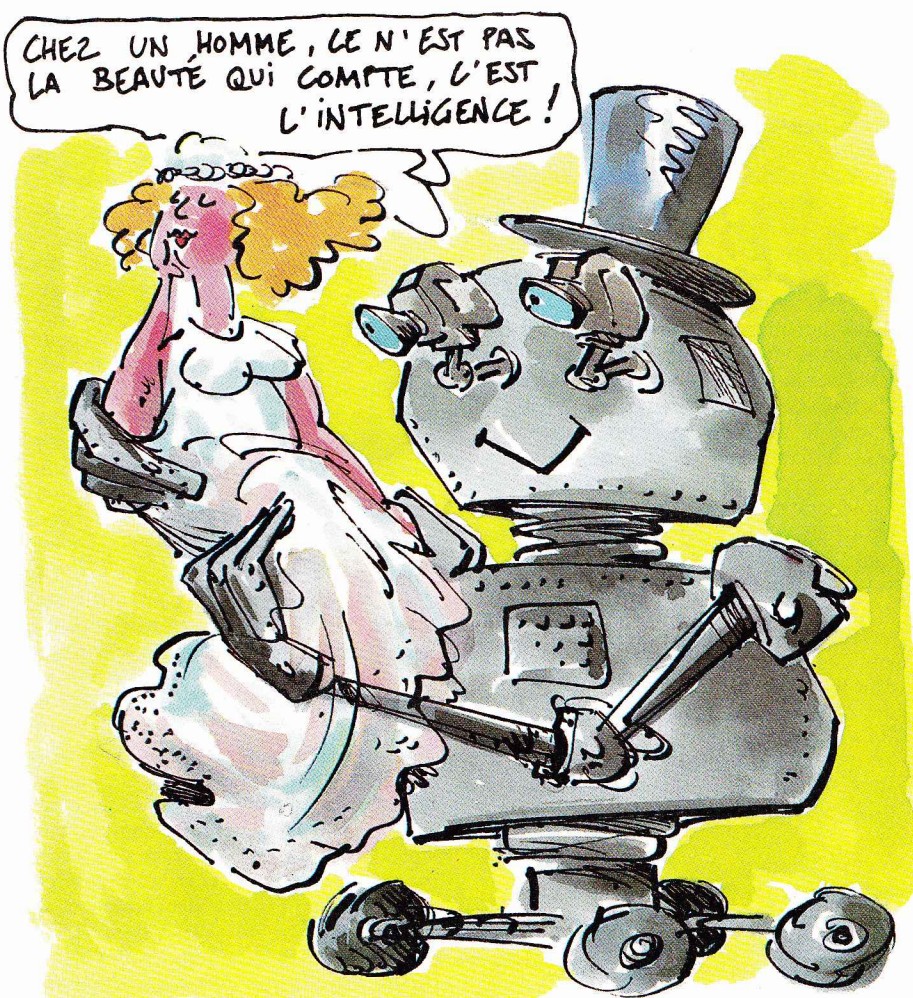


Illustration Colin Thibert

Bibliographie

- (1) « La robotique. Principes et applications », par Philippe Coiffet, Hermès, 1987.
- (2) « Des robots et des hommes », par Léon F. Wegnez, Eyrolles, 1986.
- (3) « La robotique pour ingénieurs », par Yoram Koren, McGraw-Hill, 1986.
- (4) « Le guide Robots Ingénierie 86/87 », G2H Publications, 1986.
- (5) « Robots. Le guide de l'utilisateur 1986/87 », Axes-Communication, 1987.

- (6) « Annuaire de la robotique 1987 », AFRI, Hermès, 1987.

Revue spécialisée :

- Robots Ingénierie Hebdo (G2H Public.)
- Axes robotique (Axes-Communication).
- Robots (CESTA et éditions SELF).
- Le Journal de la robotique (Saincy Communication).
- La Lettre de la robotique (Saincy Communication).

LE FINANCEMENT D

Non, le financement n'est plus ou ne peut plus être le frein majeur à la robotisation des entreprises. C'est un faux problème. L'investissement de « productique » n'est pas spécifique et il serait plutôt facile à financer.

Jusqu'au début des années quatre-vingt, le problème du financement des robots ne semble pas s'être vraiment posé en France car les équipements restaient le privilège des majors de l'industrie – automobile essentiellement – en raison de leurs coûts, directs et indirects, ou de leur délicatesse de mise en œuvre.

Aujourd'hui l'importance et la nécessité croissantes des équipements de robotique ou automatisés, leur prix et une diffusion accélérée dans les PMI conduisent utilisateurs et constructeurs à penser, pour les uns, leurs besoins de financement pour mieux contrôler des dépenses de plus en plus élevées, et, pour les autres la nécessité de préparer l'approche de leur marché, afin de mieux le pénétrer, grâce à l'argument que représentera le financement sans lequel la vente ne pourra se faire.

Afin de prendre la meilleure décision en ce domaine, le financement robotique devra tenir compte de trois facteurs :

1° Risque lié au choix du matériel ou du système : ce choix implique la nécessité de faire réaliser une étude d'implantation.

2° Analyse des coûts liés au projet :

- Le matériel : à mesure que la fiabilité et les performances des robots croissent, leurs prix tendent à diminuer, ce qui privilégie des engagements de courte durée qu'il faudra rapprocher du temps de retour sur investissement ou de sa durée probable d'obsolescence.

- La mise en œuvre des équipements et l'incidence sur l'environnement de l'entreprise qui sera amenée à modifier son processus de production.

3° Définition du « retour sur investissement » à partir des gains et perspectives de rentabilité qui sont attendus.

Actuellement plusieurs techniques de financement sont offertes aux entreprises qui envisagent l'équipement de leurs ateliers avec un robot.

Quatre moyens de financement

Les moyens de financement sont au nombre de quatre :

- l'autofinancement,
- l'emprunt à moyen ou long terme,
- le crédit-bail,

- la location.

L'autofinancement

L'autofinancement ou financement par fonds propres peut se faire :

- soit par prélèvement sur la trésorerie de l'entreprise (donc sur les fonds de roulement) ;

- soit avec un apport en argent frais (versement sur les comptes courants des associés ou augmentation de capital). Or aujourd'hui, dans les PMI, la possibilité de prélever sur la trésorerie est de plus en plus limitée, et l'apport qui va amputer le patrimoine personnel de l'entrepreneur permettra au mieux de rattraper un retard, mais tout à fait exceptionnellement de franchir un bond en avant.

L'autofinancement est donc peu adapté à des investissements de croissance que sont les robots, puisque l'entreprise aura à faire face aux besoins nouveaux induits par ces équipements : modification de l'atelier, plan de formation, modification des produits, attaque de marchés nouveaux, mise en route, constitution des stocks et encours, modification du cycle de production et toutes incidences sur la trésorerie de l'entreprise.

L'emprunt

En règle générale, les banques et établissements financiers, qui consentent à leurs clients des prêts pour leurs équipements en matériels – intervention en général à moyen terme de 2 à 7 ans –, cherchent à limiter la durée de leur intervention à la durée d'amortissement des immobilisations acquises au moyen de ces emprunts. *A contrario* de ce principe, et pour faciliter l'évolution technologique des entreprises vers des systèmes automatiques tels que robots, cellules et ateliers flexibles, informatisation de gestion de production, CAO-CFAO, les pouvoirs publics ont développé plusieurs procédures dites « bonifiées », mais dont la durée de vie fut éphémère, tels les « prêts efficacité des équipements en productique », le FIM (Fonds d'investissement et de modernisation), intervenant avec les PPT (Prêts participatifs technologiques) et autres PSI (Prêts spéciaux à l'investissement).

Aujourd'hui, priorité donc aux PMI et fin du rôle des fonctionnaires dans l'attribution des prêts.

Pour les entreprises n'appartenant pas à un grand groupe et réalisant un chif-

fre d'affaires inférieur à 500 millions de francs, il existe plusieurs procédures :

- les PBE (Prêts bancaires aux entreprises) directement pratiqués par les banques – durée de 5 à 10 ans, taux de 8,85 % à 9,15 % ;

- les PBPME (Prêts bonifiés aux petites et moyennes entreprises) distribués par le CEPME et Crédit national – durée de 5 à 20 ans avec des taux de : 9,95 % pour un prêt inférieur à 1 MF, et 9,45 % pour un prêt supérieur à 1 MF ;

- les PLT (Prêts à long terme du CEPME) – de 8 à 20 ans, taux de 11,20 %.

- les crédits Article 8 – crédits professionnels réalisés par les banques sur leurs ressources et garantis par le CEPME en liaison avec les organismes de caution mutuelle – durée de 2 à 5 ans, taux moyen 11,50 %.

Bien que pour certaines de ces procédures, il s'agisse de « financements bonifiés », il faut en souligner quelques inconvénients :

- Quotité d'intervention limitée à 70 % de la valeur H.T. des investissements, ce qui peut rendre illusoire l'attrait d'un taux nominal puisqu'il faudra prendre en considération l'incidence, sur le coût réel du crédit, qu'auront la part d'autofinancement et la sortie de TVA.

- Procédures lentes dans leurs décisions et plus encore dans leur mise en place.

- Durées d'intervention ne tenant aucun compte du temps de retour de l'investissement ni de son temps probable d'obsolescence.

Le crédit-bail

De façon simple, le crédit-bail garantit à toute entreprise l'utilisation d'un robot, comme si elle en était propriétaire : choix du fournisseur, du matériel dont elle négocie les conditions d'achat : délais, prix, en contrepartie du paiement d'un loyer fixe pendant un délai convenu et sans obligation de rachat de la valeur résiduelle à l'issue de la location.

Principes de l'intervention :

- 100 % de l'investissement du robot et de son environnement, et accessoirement du coût de l'étude d'implantation.

- Durée variable de 3 à 5 ans.

- Valeur résiduelle faible, en général 2 %.

- Délais de décision et de mise en place courts : de 24 heures à 10 jours.

- Coût apparent qu'il faut rapprocher

E LA ROBOTISATION

Illustration Colin Thibert



des avantages qu'il procure :

1° Financement total, TVA comprise, il préserve le fonds de roulement et l'indépendance financière de l'entreprise, laquelle peut ainsi tirer un profit immédiat d'un investissement productif, en consacrant ses ressources propres à la satisfaction des besoins de son exploitation : recherche, stocks, encours, prospection, ventes, formation.

2° Financement à taux ferme, il permet de connaître précisément le coût d'utilisation d'un robot et facilite donc le contrôle de gestion dans une période de crise et d'argent cher, où la moindre erreur sur le calcul de la rentabilité d'un investissement peut avoir de graves conséquences.

3° Financement économique, il permet de mesurer exactement la charge financière de l'équipement face à sa rentabilité, et tient compte du retour de l'investissement et de son obsolescence ; il peut intégrer la totalité de l'équipement, frais d'étude compris.

A ces avantages, le crédit-bail allie l'adaptabilité, la souplesse par la modulation de ses durées d'intervention et la grande variété des schémas qu'il peut mettre en œuvre.

La location

L'incertitude conjoncturelle et l'évolution rapide des technologies, ajoutées au développement des besoins locatifs des utilisateurs, ont incité les entreprises à rechercher des durées d'utilisation « à la carte », c'est-à-dire pour des durées correspondant le plus rapidement possible à leurs moyens de prévisions.

Au sens strict, la location n'est pas un financement, mais elle peut se définir comme une formule souple et discrète de mise à disposition d'un robot équipé, choisi librement par le futur locataire-utilisateur, pour une période déterminée à court terme, de 12 à 24 mois, ou à moyen terme, de 24 à 60 mois comme le propose Technibail, filiale spécialisée du groupe Bail Equipement.

Le prix de ce service est un loyer sans surprise, puisque non indexé.

Pour l'industriel, la location peut être un moyen de s'adapter à l'évolution technologique des systèmes robotiques et d'en prévoir sans risque les extensions possibles. Ainsi, dans le cas d'une extension matériel et/ou logiciel, le locataire aura le choix entre deux solutions :

- conserver comme date de fin de location la date initialement prévue au premier contrat, avec modification de son loyer d'origine, augmenté du complément de loyer correspondant au nouvel équipement ;
- ou refondre totalement le premier contrat de location afin de définir un allongement de la période de location, sans modification du loyer, malgré le coût supplémentaire entraîné par l'extension.

Procédé discret, car n'ayant d'incidence que sur le poste Services extérieurs du compte de résultats, et non sur le bilan de l'entreprise, la location Technibail offre la souplesse et la discrétion.

Si l'évolution rapide des technologies accentue l'importance du « bon choix »

de l'investissement, donc du robot, le « bon choix » du financement revêt aussi la même importance pour le chef d'entreprise ou son responsable financier. Il leur appartiendra, avant de se déterminer, de peser chaque formule en respectant quelques règles essentielles :

- équilibre entre la durée du retour sur investissement et la durée du financement retenu ;
- définition exacte des charges de remboursement et de l'incidence financière totale ;
- équilibre entre les capitaux de l'entreprise et ses emprunts ;
- définition des évolutions prévisibles de son équipement robotique et des futurs besoins de l'entreprise.

En règle générale, le choix devra être guidé non par le seul attrait d'un taux nominal d'un prêt ou par la durée de celui-ci, mais par une analyse qui intégrera aux éléments économiques :

- L'investissement sera-t-il opportun dans l'environnement économique de l'entreprise ?
- Dans quels délais sera-t-il rentable ? les éléments d'une étude financière ?
- Quel taux de profit va-t-il dégager ?
- Quelles incidences le choix du financement entraînera-t-il sur les comptes de l'entreprise ?

Les résultats de cette analyse seront significatifs pour amener le chef d'entreprise à préparer sa décision face à son projet.

Joseph Mary*

* Chef de marché Productique du groupe Bail Equipement.

MacIntosh Plus

OPERATION REPRISE DE VOTRE MAC 128 K ET MAC 512 K ☎ 48.78.38.01

Que vous soyez artisan, étudiant, consultant ou profession libérale, que vous soyez une PME ou gérant d'une association; MacIntosh Plus répondra intelligemment à votre attente. MacIntosh a déjà conquis le cœur de centaines de milliers d'utilisateurs professionnels. Simple, puissant, fonctionnel, il est doté d'une bibliothèque de logiciels exceptionnelles.

Microprocesseur 16/32 bits MC 68000 à 7,8 MHz.
1 Mo de mémoire centrale extensible à 4 Mo.
128 Ko de mémoire morte.
1 lecteur de disquettes 3 1/2 intégrés d'une capacité de 800 Ko.
1 clavier AZERTY accentué avec bloc numérique intégré et touches de déplacement.

1 écran graphique vidéo noir et blanc très haute résolution.
2 interfaces série.
1 connecteur pour un lecteur de disquettes et/ou pour un disque dur.
1 connecteur SCSI pour chaîne de périphériques à haute vitesse.
1 connecteur pour la souris. 1 souris.

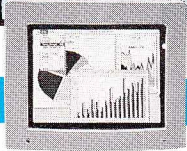
MacIntosh SE

TARIF COMPLET SUR NOTRE SERVEUR MINITEL 48 78 79 92

C'est le MacIntosh complet, compact et puissant. Les lecteurs de disquettes et disque dur sont des éléments standards intégrés sur le MacIntosh SE; de plus, il contient un connecteur interne apte à recevoir une carte d'extension. Le résultat c'est un outil rapide, puissant et souple qui va satisfaire la plupart des professionnels.

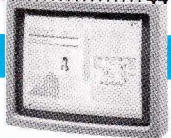
L'espace-conseil Ma

Périphériques/Imprimantes



La meilleure définition au meilleur prix !

Moniteur Couleur EIZO pour Mac II, 14" (820 x 620) **6 990 TTC**
Moniteur Couleur Taxan pour Mac II, 14" (640 x 480) **5 600 TTC**
Moniteur couleur Super Mac, 19", pour Mac II (hte déf. 1024 x 763) avec interface, 256 couleurs **44 900 TTC**



Ecran monochrome Monitern 19" (Mac SE/Mac II, hte résolution 1280 x 960) avec carte vidéo **17 900 TTC**

Ecran Radius Ecran pleine page A4, résolution 640 x 864) **19 900 TTC**

Ecran tactile Microtouch (Mac SE) 9", la grande nouveauté de Mac World expo!!!, supprime l'utilisation de la souris **26 900 TTC**

Mega Screen : (Ecran pleine page résolution 1024x900) **21 900 TTC**

Agfa-Geveart (Digitaliseur 200 DPI) **28 900 TTC**

Data Copy (Digitaliseur 300 DPI) **4 485 TTC**

Mac Vision (Interface Caméra Vidéo) **4 695 TTC**

Thunderscan (Interface Image Writer I, II) **3 950 TTC**

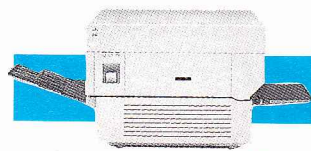
Kurta (Tablettes à Digitaliser) **5 200 TTC**

Pen Mouse 6" x 9" **4 990 TTC**

Penmann (Table traçante) **2 900 TTC**

Image Writer II + cables **2 900 TTC**

Seikosha **2 900 TTC**



Seikosha 120 Cps (Matricielle) **2 990 TTC**
Laser Writer Plus (Canon LBP-CX) (Résolution 130 pts) **2 900 TTC**

Communications/Réseaux

MAC 286 Pour Mac II (carte Emulation MSDOS) **13 900 TTC**
Tops (Pour IBM PC) **3 800 TTC**
Tops (Pour Mac) **1 600 TTC**
Tops Print **1 600 TTC**
Mac Link Plus (V. Europe) **2 200 TTC**
Mac Transier (Transférer l'univers Apple II et Mac) **695 TTC**
Carte MS/DOS SE (MACSE) **7 500 TTC**
PC Mac Bridge (Relie des IBM PC ou Compatible au réseau Apple Talk) **10 200 TTC**



Versa Term (Emulation Dec VT100, Data Général 200, etc...) **2 550 TTC**
Multi Talk (Partage de Modem, Imprimante sous Apple Talk) **5 900 TTC**
Mac Tell (Logiciel de Communication) **1 990 TTC**
Microphone (Logiciel de Communication) **1 450 TTC**
Symb Talk 3 (Serveur de bloc) **14 200 TTC**
Symb Share (Serveur de fichiers) **18 500 TTC**
Apple Share (Serveur de fichiers) **5 900 TTC**
Modem Diapason intelligent (vitesse 300/300 et 1200/75) **4 990 TTC**
Modem Universel Apple (300/300 & 1200/75) **2 600 TTC**

Stockage/Extension



Lecteur 800 K (Compatible/Chinon) **2 190 TTC**
Lecteur 800 K Apple **2 900 TTC**
Lecteur 400 K Apple **900 TTC**

Transformation Mac 512 -> Mac Plus
Port SCSI pour Mac 512 K
ROM pour Mac 512 K
Extension Mac Plus 2 Mo + 68020 à 16 MHz

CIRRUS (IMPORT USA)

Disque Dur 20 Méga SCSI **6 990 TTC**
Disque Dur 40 Méga SCSI **10 900 TTC**
Disque Dur 60 Méga SCSI **15 900 TTC**
Sauvegarde 40 Méga (micro cassette 3M) **10 900 TTC**

LA CARTE PRIVILEGE

Devenez membre du CLUB PRIVILEGE. Vous bénéficierez ainsi de nombreux avantages. Renseignez-vous vite 48.78.38.01.

Disque Dur SCSI, 20 Mo (mécanique Seagate)
Prix Club **5 990**

3 Macintosh, 2 Lasers, 1 Scanner Agfa sont à votre disposition Scanner Service à votre disposition, venez éditer, composer, mettre en page vos documents.

TARIFS CLUB :

50 F/heure et 4 F la copie Laser Writer
200 F/heure pour scanner.
Prix spéciaux pour les membres du Club.
ACCE met à votre service son équipe pour la formation, dans nos locaux ou sur site, et le développement/paramétrage 4D/EXEL ainsi que la création de serveur sur MacIntosh. Consultez-nous au 48.78.38.01.

Disque Dur 20 méga **9 900 TTC**
Disque Dur 20 méga Scsi
Apple **9 600 TTC**
Disque Dur 40 méga Scsi
Externe Apple **19 900 TTC**
Disque Dur 80 Méga Scsi
Externe Apple **27 500 TTC**
Disque Dur 145 Mo interne pour Mac II **24 900 TTC**



Extension 2.5 Méga pour Mac SE **5 700 TTC**
Extension 4 Mo
Mac Plus/SE/Mac II **11 500 TTC**
Extension 8 Mo/Mac II **22 900 TTC**
Extension MAC 512 -> 1 Mo **2 300 TTC**
Extension Mac 128/512 **2 500 TTC**

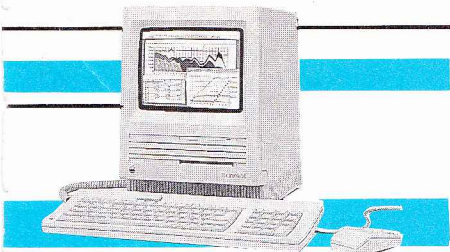
Logiciel



Gestion/ Base de Données/ Fichiers/Tableurs

EXEL : C'est le tableur le plus évolué existant sur un ordinateur personnel. En un clin d'oeil vous calculez, simulez, modifiez des séries de chiffres pour ensuite représenter le résultat sous forme graphique.

MS Excel **3 600 TTC**
MS Works **1 840 TTC**
Ms File (Fichier) **2 100 TTC**
Tableur Multiplan (Tableur) **2 790 TTC**
4^e Dimension (Base de données relationnelle) **5 500 TTC**
Gestion Simil (Gestion compta/facturation/stock) **6 700 TTC**
Compta Simil (Comptabilité Générale) **4 800 TTC**
ABC 2035 (Compta Prof Libéral) **2 750 TTC**
Compta Maestria (Compta Générale) **4 400 TTC**
Facturation Ténora (Facturation avec Compta /Maestria) **2 900 TTC**



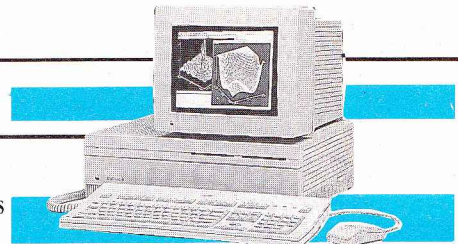
Microprocesseur 16/32 bits MC 68000 à 7.8 MHz.
1 Mo de mémoire vive, extensible extensible à 4 Mo.
256 Ko de mémoire morte.
1 ou 2 lecteur(s) de disquette 3 1/2 d'une capacité de 800 Ko.
1 disque dur intégré SCSI, d'une capacité de 20 Mo.

1 écran intégré noir et blanc de 9", 512 x 342. 2 interfaces série intégrées RS 422.
Interfaces SCSI 7 périphériques.
ADB 16 périphériques.
1 connecteur interne d'extension, 96 broches.
Clavier détachable avec bloc numérique. 81 touches.

Macintosh II

Macintosh II est le plus rapide de la famille Macintosh, et sans doute de la plupart des ordinateurs personnels. Il a des capacités de mémoire et de stockage jusqu'alors réservées à l'informatique "lourde". Sous le capot : 6 slots d'extension, microprocesseurs multipliant par 4 l'exécution des logiciels et calculant 200 fois plus vite. Ecran couleur, mémoire de 2 à 8 Mo, disque dur interne jusqu'à 80 Mo, environnement DOS et UNIX.

Avec le Macintosh II, vous rentrez dans le monde des grands ordinateurs.



Microprocesseur 32 bits MC 68020 à 16 MHz.
1 co-processeur arithmétique 68881.
1 mégaoctets de mémoire vive, extensible à 8 Mo.
256 Ko de mémoire morte.
2 lecteurs de disquette 3 1/2 intégrés d'une capacité de 800 Ko.
1 disque dur SCSI intégré de 20, 40 ou 80 Mo.
1 écran séparé 12" N & B : très haute résolution.

1 écran séparé 13" couleur : très haute résolution.
Carte vidéo 640 480, résolution de 4 bits points extensible à 8 bits par point. 2 interfaces série intégrées RS 422.
6 connecteurs d'extension. 96 broches.
Norme Nu-Bus.
Interfaces SCSI 7 périphériques.
ADB 16 périphériques.
Clavier détachable avec bloc numérique - 105 touches.

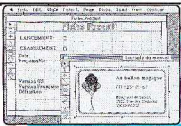
Macintosh...

ACCE-Informatique, 4 bis, rue de Chateaudun
75009 Paris - Tél. 48 78 38 01

Le premier magasin entièrement dédié au Macintosh. Dans un cadre professionnel et sympathique, nous vous ferons découvrir tous les secrets du monde Macintosh. ACCE-Informatique, concessionnaire agréé APPLE Macintosh est lauréat du prix d'excellence Apple pour la qualité de ses services et le professionnalisme de son équipe.

Concessionnaire agréé

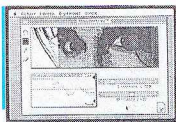
Apple



Dessin/ Traitement de Texte/ Mise en Page

Mac Draw (Dessin) 990 TTC
Mac Draft (Dessin et Rotation Image) 3 390 TTC
Mac Paint (Dessin en Bitmap) 560 TTC
MS Word 3.01 2 760 TTC

Writer Plus (Traitement de Texte et Mise en Page) 2 750 TTC
Mac Write (Traitement de Texte) 990 TTC
Write Now (Traitement de Texte et Mise en Page V.U.S.) 1 700 TTC
Page Maker (Logiciel de Micro Edition) 5 990 TTC
Ready Set Go 3.0 (Logiciel de Micro Edition) 4 800 TTC
XPress (Logiciel de Micro Edition V.U.S.) 6 900 TTC
Just Texte (Textes et Images en Postscript) 2 700 TTC
Laser Paint (Textes et Images en Postscript) 3 800 TTC



Utilitaires

Nouveau Mac Plus/SE/Mac II (Numeriseur sonore) 2 500 TTC
SuperLaserSpool (Spooler Laser Writer) 750 TTC
SuperSpool (Pour I/W II N.V Mac SE Mac II) 730 TTC
DiskTop 3.0 (Gestionnaire Disque Dur) 590 TTC
Smart Alarm (DA/Agenda électronique) 400 TTC
Sentinel (Verrouillage documents) 990 TTC
Print Works(Spooler I/WII & Impression Couleur) 750 TTC
Mac Zap (Utilitaires de Copie) V5.0 690 TTC
Disk Fit (N.V Sauvegarde Disque Dur) 690 TTC
HD Backup (N.V Sauvegarde HD) 590 TTC

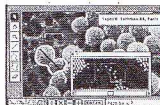
Tous nos prix s'entendent en francs TTC

Disk Express (Fragmentation HD) 595 TTC
HD Util (installeur logiciel sur HD) 795 TTC
Copy II (Version 7.0)
Compatible SE/Mac II) 450 TTC
Super Glue (Imprimer Sans Logiciel) 890 TTC



Langages

Light Speed "C" (Compilateur C) 1 200 TTC
Light Speed Pascal (Compilateur Pascal) 990 TTC
MS Basic (Compilateur basic) 1 990 TTC
Z basic 800 TTC
Object logo (pour mac) 1 200 TTC
V.I.P. 1 300 TTC
Turbo pascal (Compilateur Pascal) 990 TTC



Jeux/ Import U.S.A.

Mac Golf (Simulation Golf) 620 TTC
Sub Battle (Bataille navale) 350 TTC
Shadow Gate (Jeu d'aventure avec Souris) 595 TTC
MAC Bridge 300 TTC
Ferrari (Course de voiture) 595 TTC
Puppy Love 440 TTC
Kids Time (Educatif enfants 1^{er} âge) 400 TTC
Déjà vu (jeu d'aventure avec souris) 595 TTC

Accessoires/Consommables

Pochette Souris (Mac SE) 95 TTC
Pochette disquette (Pour Mac SE) 95 TTC
Housse Mac Plus, Mac 128/512, Mac SE 120 TTC
Housse Image Writer 120 TTC
Housse Lecteur 400/800 75 TTC
Kit Nettoyage souris 195 TTC
Kit Nettoyage Lecteur 250 TTC
Filtre Ecran Polaroid 395 TTC
Support Imprimante 250 TTC

Support Tourant Mac Plus/Mac 128/512/Mac SE 395 TTC
Turbo Mouse Mac 128/512/Plus 990 TTC
Sac de transport Mac +/SE (USA) 750 TTC
Clip Apple Talk (connectique réseau Apple) par 10 100 TTC
Tapis souris 995 TTC
Coffret insonorisant (Image II) 995 TTC
Kit de sécurité Mac 295 TTC
Ventilateur externe (Mac +) 990 TTC
Mac Tilt 595 TTC

DISQUETTES 3 1/2 SONY

Par 10 Par 100
Sony SF/SD 185 165
Sony DF/DD 185 165
Nashua DF/DD 175 155
Neutres DF/DD 109 99

Rayon librairie, revues US.

CHATELAIN pour SAP

TOUS NOS PRIX SONT TTC



I N F O R M A T I Q U E

48 78 38 01

BON DE COMMANDE

(sauf pour produit Apple)

MS 12/87

| DESIGNATION | NOMBRE | PRIX |
|---------------------------|--------|------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| FORFAIT PORT jusqu'à 3 kg | | 40 F |
| | TOTAL | |

Nom, Prénom
Adresse
Ville
Code Postal

COUPON REPONSE

Demande de documentation
 Je suis intéressé par.....

ACCE INFORMATIQUE
L'ESPACE CONSEIL MACINTOSH
4 bis, rue de Chateaudun - 75009 Paris
Ouvert du lundi au samedi de 10 h à 19 h
Mo Cadet ou N.D. de Lorette



ROBOTIQUE & FUTUR:

LES ROBOTS NOUVEAUX SONT ARRIVÉS

Sortis, dans les années 20, de l'imagination fertile du romancier tchèque Karel Capek, les robots, « ces ouvriers artificiels », n'ont cessé depuis le début de ce siècle, de hanter l'imagination des auteurs de fictions littéraires ou cinématographiques. Plus sérieusement et depuis plusieurs années, les robots sont des outils précieux œuvrant dans les domaines industriels et scientifiques. Ils font ainsi l'objet de recherches intensives sur l'Hexagone, en attendant des projets à l'échelle de l'Europe.

Si l'on en croit la très officielle définition donnée par le petit Robert, le robot est un « mécanisme automatique à commande électromagnétique, pouvant se substituer à l'homme pour effectuer certaines opérations et capable d'en modifier lui-même le cycle... » De l'âge des pionniers dans les années cinquante à aujourd'hui, le petit monde de la robotique a considérablement évolué et s'acheminé vers une troisième génération. Rendus plus « intelligents », donc plus autonomes, ces robots annoncent déjà des bouleversements profonds au sein des entreprises.

Comme le souligne une étude menée au sein du laboratoire d'automatique et d'analyse des systèmes (LAAS) du CNRS, le développement de la robotique mobile a posé le délicat problème de la vision tridimensionnelle, autrement dit la perception des images à trois niveaux au lieu de deux, le tout reposant sur l'emploi de nouveaux capteurs.

En effet, l'un des points spécifiques de cette mobilité est la complexité de l'espace dans lequel ils sont amenés à évoluer. Les auteurs constataient ainsi que l'univers d'un robot est le plus souvent vaste, variable et relativement peu structuré...

Pour se mouvoir, l'engin a donc besoin de « comprendre » la nature exacte du « terrain ». Comme l'explique un responsable de LAAS, l'activité scientifique au sein du projet Robots mobiles est centrée autour du principe de « l'autonomie décision-

nelle ». En fait, au cours des années écoulées, se sont surtout posées les problématiques dites de la « fusion multisensorielle » et de « structure de contrôle ». La première repose sur le fait que le robot mobile est un système multicapteurs. C'est ainsi que pour modéliser l'environnement, il s'agit d'intégrer de façon consistante les données et les représentations issues des capteurs, compte tenu évidemment des différentes incertitudes, erreurs de données, etc. Précision importante : cette fusion doit nécessairement s'opérer au niveau numérique et symbolique. Le second concept affirme que la conception d'une structure de contrôle d'un robot mobile est un problème scientifique important et de fond.

L'ensemble de ces travaux trouve ainsi dans le robot mobile « Hilare » un terrain d'expérimentation qualifié par ses concepteurs « d'idéal ». A noter que ce système, conçu pour se déplacer en intérieur ou aux alentours immédiats d'un bâtiment, est, pour l'heure, incapable de franchir des obstacles même minimes. C'est ainsi que les travaux menés ces dernières années sur Hilare ont abouti à l'élaboration de trois types de modèles : géométrique, topologique et sémantique.

Depuis 1984, les robots mobiles ont non seulement retenu l'attention d'organismes publics de financement – civils et militaires – et d'aide à la recherche, mais également de certains industriels intéressés par les possibilités futures de ces procédés. L'ensemble de ces travaux a donné lieu à un

certain nombre de contrats, concernant notamment IBM pour la conception et la réalisation d'un robot mobile de service d'atelier, le contrat européen Esprit 1 560 Skids sur la fusion multisensorielle, ou encore dans le cadre d'Eureka, deux réalisations concernant l'une, AMR, des robots mobiles d'intervention pour la Sécurité civile, la seconde baptisée Prometheus est une aide intelligente à la conduite automobile. A signaler également des activités de recherche et développement comme le projet « contrôle d'exécution pour robots mobiles » du Gip-Promip regroupant l'ensemble de l'Etablissement régional Midi-Pyrénées, le CNRS, l'ONERA, l'UPS, l'IMP, ainsi que la société Midirobots.

Comme le note le LAAS, les thèmes majeurs à souligner dans le développement de ces travaux concernent principalement la perception et la fusion multisensorielle.

C'est ainsi que le groupe Robotique et Intelligence artificielle (RIA) a élaboré un programme de recherches centré sur la problématique de cette fameuse troisième génération. A noter également qu'à côté des « mobiles » comme Hilare, s'est développée la cellule flexible d'assemblage (CFA) considérée comme le prototype de la robotique de travail à poste fixe. C'est dans ce cadre, né du programme ARA en 1981, qu'ont été étudiés des thèmes comme la commande adaptative et asservie à la tâche, la génération automatique de trajectoires, le contrôle d'exécution, etc.

Arnault Lafaille, directeur adjoint de l'AFRI : Un marché porteur mais encore restreint...

« Il faut développer au maximum les capacités de décision et augmenter l'autonomie des robots », assure-t-il, notant au passage un projet de système mobile destiné à une chaîne hôtelière, pour le nettoyage et l'entretien des locaux... Pour lui, cependant, « l'impact social devrait être minime... ». Et l'ouverture du Marché Européen en 1992 ? « Les programmes Esprit et Euréka devraient renforcer les liens existants entre les différents pays de la CEE. » Reste évidemment à résoudre le délicat problème des normes... « Cette interrogation devrait trouver une réponse dès le 1^{er} trimestre 1988, avec une définition d'un certain nombre de règles communes concernant les robots... ».

A propos du programme Euréka, divers projets européens – sept au total et deux par pays – concernant la robotique ont été sélectionnés et regroupés sous le label FAMOS. C'est ainsi que la France devrait présenter les travaux des sociétés Merlin Gerin (contacteurs) et Philips pour la production de machines à laver.

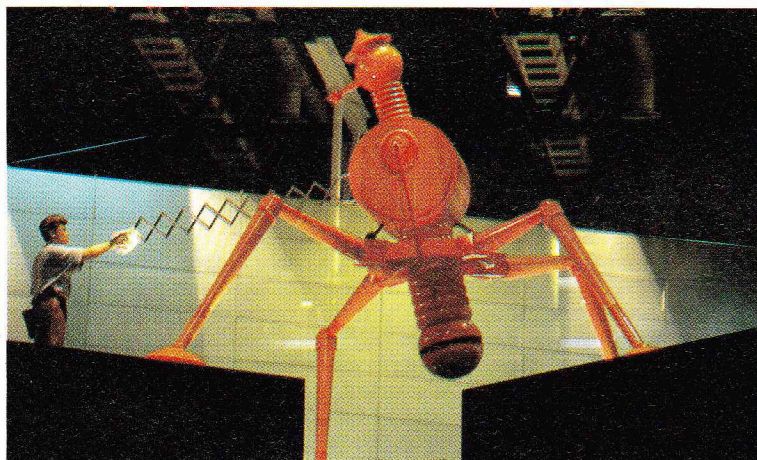
« Pour l'heure, il existe une cinquantaine d'entreprises nationales fabricantes de systèmes robotiques, s'adressant pour l'essentiel aux grandes unités... »

Ainsi, souligne-t-il encore, même si le thème de robotique est porteur, il reste néanmoins et restera pour longtemps un marché restreint.

La France dans le peloton de tête européen et mondial...

Pour l'Association française de robotique industrielle – 300 adhérents –, la France se situe actuellement au deuxième rang européen et au quatrième rang mondial, en nombre de robots installés. C'est ainsi qu'en 1986 le parc global de systèmes en service était de 5 270 robots de classes C et D (programmables et « intelligents ») dont 1 120 robots industriels (C et D) et 900 manipulateurs automatiques à séquences variables. Avec un taux de croissance annuel de 27 %, la robotique industrielle française a exporté quelque 180 robots et 64 % de la production était nationale.

L'AFRI classe ainsi les robots industriels en diverses catégories : A pour les engins de manipulation motorisés, commandés par l'homme et ayant au moins quatre degrés comme les appareils de déplacements de charge. Classés également A, les télémanipulateurs où la commande s'effectue à distance avec l'aide de leviers/boutons pour la manutention en ambiance dangereuse, forge, nucléaire et subaquatique. La seconde catégorie – B – concerne le manipulateur automatique, engin non asservi, à cycle automatique, fonctionnant en séquence fixe (machine à peindre, chargement/téléchargement de presse, chargeurs d'outils, etc.) ou en séquences variables (chargement/déchargement de machine-outil, assemblage, etc.). Classé en catégorie C, le robot programmable se définit comme un système automatique de trois axes et plus, dont – au moins – deux sont programma-



Dr Livingsteel.

Distribution des robots installés par applications
Tableau comparatif 1985-1986 (robots de classe C et D)

| | FRANCE | | R.F.A. | | U.K. | | ITALIE | | SUEDE | | BELGIQUE | | ESPAGNE | |
|----------------------------|--------|-------|--------|--------|-------|-------|--------|-------|-------|-------|----------|-------|---------|-----|
| | 85 | 86 | 85 | 86 | 85 | 86 | 85 | 86 | 85 | 86 | 85 | 86 | 85 | 86 |
| ROBOTS DE PROCESS | | | | | | | | | | | | | | |
| Soudage par points | 910 | 1 050 | 2 548 | 3 100 | 511 | 534 | | | | | 561 | | 336 | 350 |
| Soudage à l'arc | 476 | 605 | 1 781 | 2 300 | 411 | 454 | | | 674 | | 84 | | 102 | 146 |
| Projection | 248 | 287 | 775 | 1 100 | 193 | 210 | | | | | 26 | | 47 | 50 |
| Parachèvement, usinage | 105 | 127 | 318 | 90 | 279 | 288 | | | | | 7 | | 3 | 4 |
| Collage | 24 | 63 | | | 43 | 68 | | | | | | | 21 | 29 |
| ROBOTS MANIPULATION | | | | | | | | | | | | | | |
| Charg./Décharg. M.O. | 1 430 | 1 956 | 1 236 | 1 790 | 952 | 1 172 | | | | | 84 | | 112 | 142 |
| Manutention-Palettisation | | | 1 179 | 1 750 | 130 | 145 | N | N | 834 | | 21 | N | | |
| Assemblage | 452 | 620 | 753 | 1 650 | 294 | 348 | | | | | 5 | | 34 | 64 |
| Inspection, test | 37 | 54 | | | 56 | 62 | C | C | | | | C | 6 | 6 |
| EDUCATION-RECHERCHE | 250 | 282 | 210 | 310 | 165 | 211 | | | | | 109 | | 11 | 21 |
| DIVERS | 218 | 226 | | 310 | 174 | 191 | | | 538 | | 78 | | 21 | 42 |
| TOTAL | 4 150 | 5 270 | 8 800 | 12 400 | 3 208 | 3 683 | 3 808 | 5 000 | 2 046 | 2 383 | 975 | 1 050 | 693 | 854 |

Origine : Associations européennes de robotique

THÈME DU MOIS

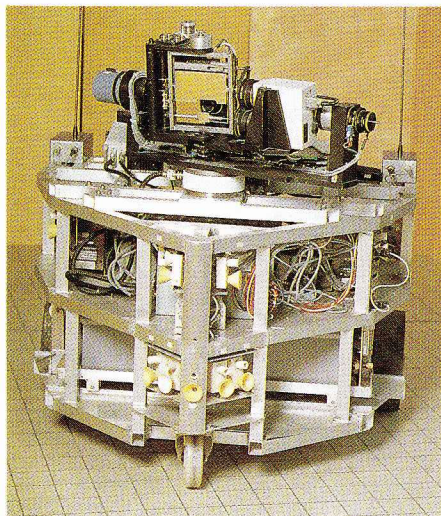


Photo © CNRS.

Robot Mobile Hilare.

bles par apprentissage ou par langage symbolique. Pour cette catégorie, il faut distinguer les manipulateurs automatiques comportant moins de cinq axes programmables et les manipulateurs automatiques cinq axes et plus.

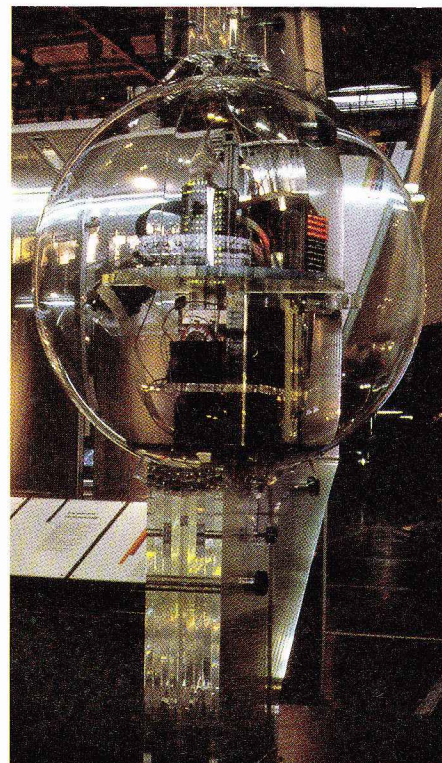
Entre ainsi dans cette optique le robot vertical, horizontal, portique, scara à l'exclusion toutefois des machines-outils CN, DNC, centre d'usinage, chariots autogui-

dés, tables traçantes, etc. Enfin la catégorie D, représentant les robots « intelligents », autrement dit les manipulateurs automatiques programmables, capables d'analyser les modifications de l'environnement et de réagir selon la situation : manipulations avec reconnaissance de forme et assemblage.

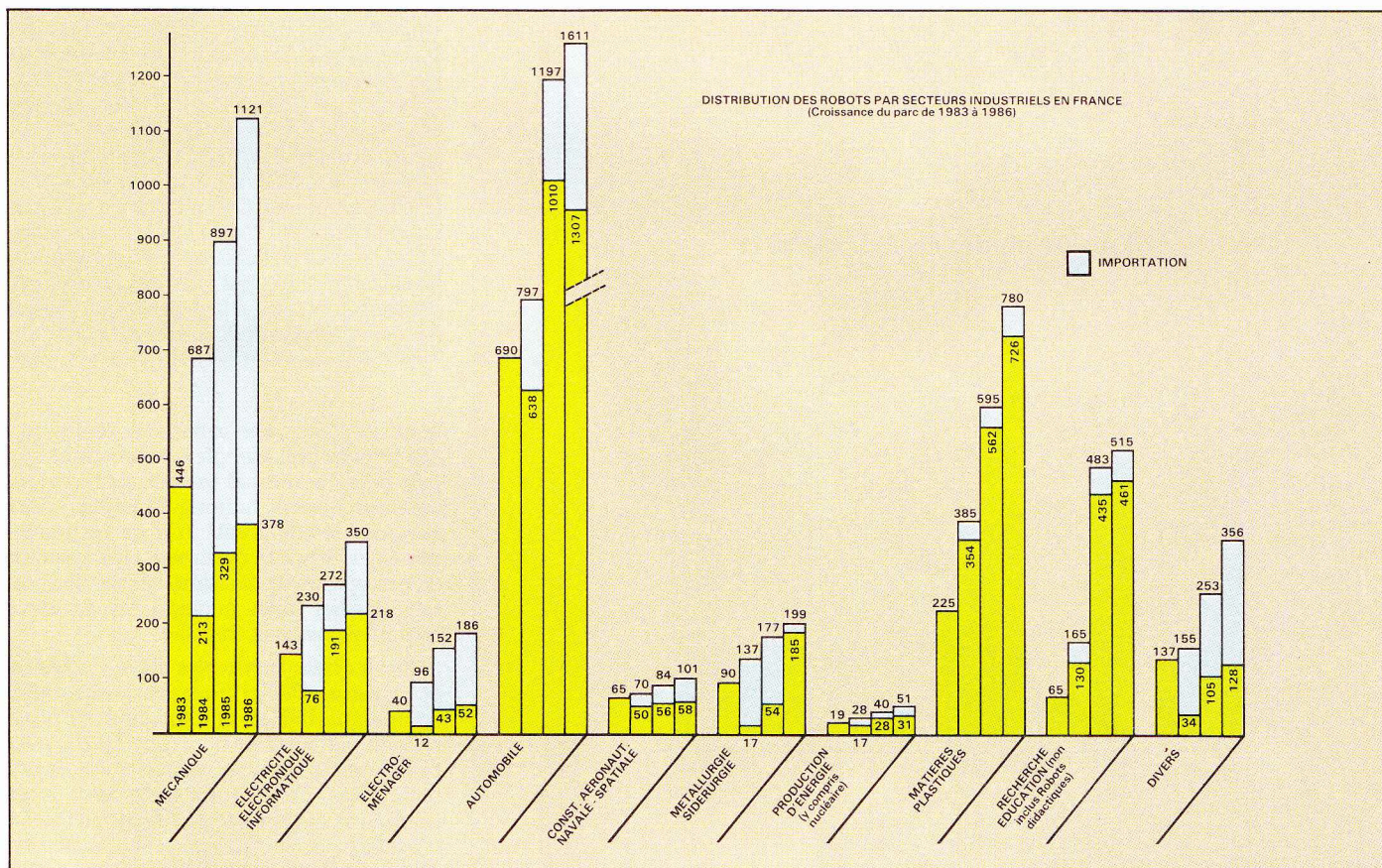
Des liens entre Recherche et Industrie

Tout comme le CNRS, l'Institut national de recherche en informatique et en automatique (INRIA) a mené d'importants travaux sur la robotique du futur. Ainsi, comme l'a démontré '87 ICAR, conférence internationale organisée cette année du 13 au 15 octobre à Versailles avec le concours du JIFRA (Japan Industrial Robot Association), sous le patronage de l'AFRI, « la Robotique de troisième génération » – thème de cette manifestation – est bien la préoccupation numéro un des chercheurs et des industriels concernés.

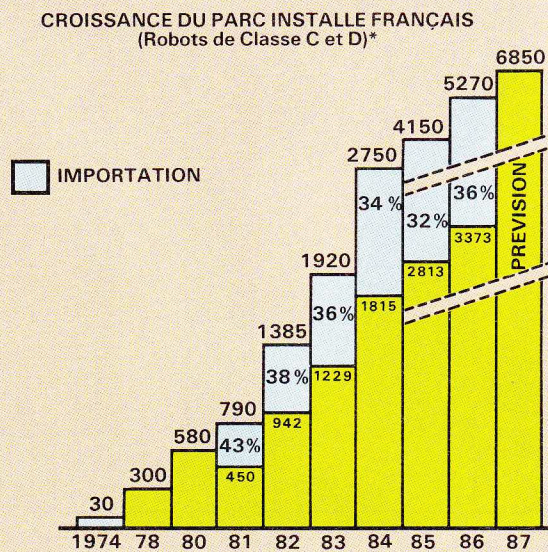
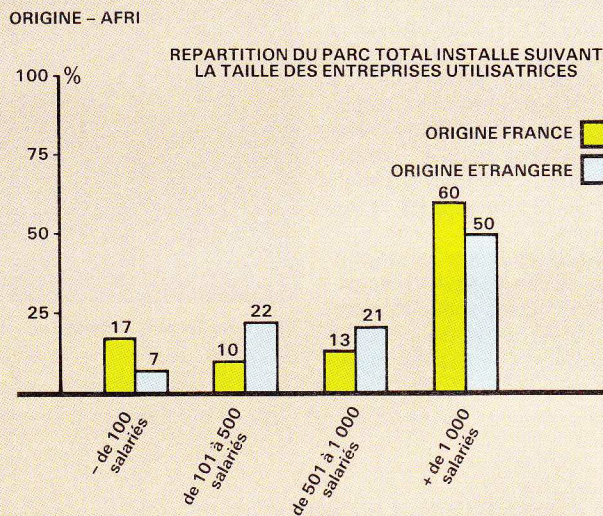
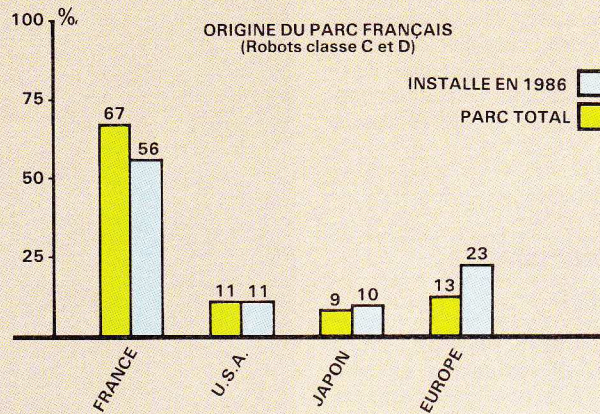
Comme devaient le souligner les responsables de ce colloque, les objectifs poursuivis par ce type de systèmes sont des « performances accrues en flexibilité, en vitesse et en précision d'exécution ». On peut égale-



Musée des Sciences et techniques : le bon robot.



Distribution des robots par secteurs industriels en France (croissance du parc de 1983 à 1986).



* C - Robot programmable
D - Robot dit « intelligent »

Georges Giralt, directeur de recherche au CNRS, responsable du LAAS de Toulouse: un impact social important

« Nous avons axé toute notre action sur la robotique de troisième génération », explique ce dernier notant que la mobilité des robots couvre des domaines aussi divers que la maintenance, la lutte contre l'incendie, les services de protection civile, voire même les domaines nucléaire, agricole ou maritime.

« Dans le cadre d'Euréka, divers pays dont la France, l'Italie et l'Espagne travaillent déjà sur des projets de robots mobiles... »

Une fois n'est pas coutume, reconnaît volontiers Georges Giralt, la France se situe en matière de recherche sur la robotique parmi les premiers au rang mondial. Preuve de cette « santé éclatante », l'échange de chercheurs entre la France... et le Japon, en attendant les États-Unis, leader incontesté dans ce domaine.

Concernant les conséquences sociales liées à l'introduction massive de la robotique au sein des entreprises françaises et européennes, Georges Giralt est catégorique: « L'impact risque d'être énorme... »

Et l'avenir? « Nous y sommes déjà, puisque le CNRS vient de signer avec la Chine une convention destinée à aider la construction et la mise en place, à Pékin, d'un institut de robotique. » Ce pays immense, au potentiel culturel important, devrait donc être « le grand enjeu économique et industriel de demain... »

ment y ajouter une capacité à répondre à des besoins plus complexes (robotique spatiale ou sous-marine) tout en améliorant la communication homme-machine. C'est ainsi que les différents axes de recherches liés à ces thèmes concernent des fonctions de perception, de décision, de planification, sans oublier bien entendu les fonctions communications et contrôle.

Sur les liens inévitables entre Recherche et Industrie, Olivier Fougenas, responsable scientifique INRIA et animateur du projet « vision par ordinateur et robotique », note, par exemple, qu'une collaboration importante vient d'être développée avec la société Noesis, chargée de l'industrialisation d'un logiciel - Visilog - de traitement d'analyse d'images, développé à l'INRIA et à l'École des mines de Fontainebleau.

THÈME DU MOIS

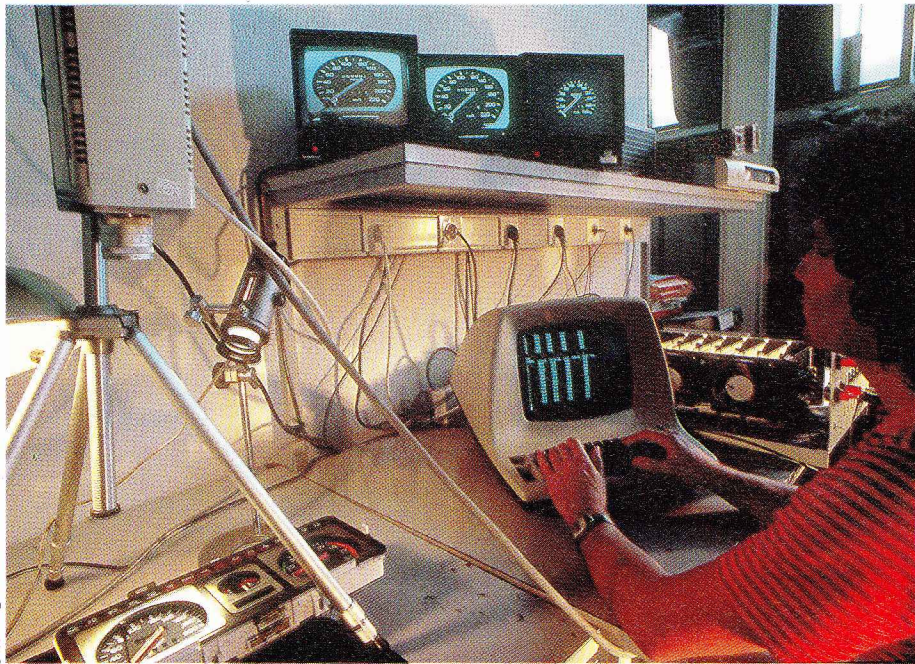


Photo © CNRS.

Reconnaissance des formes.

Par ailleurs, à l'intérieur du projet Esprit (P 940), l'INRIA a poursuivi sa collaboration avec différentes sociétés françaises (Matra Espace, ITMI et Noesis) ou étrangères comme Eltag en Italie et GEC en Grande-Bretagne. Pour cette année, Olivier Faugenas annonce déjà une collaboration avec EDF pour la réparation de centrales nucléaires. C'est ainsi qu'un premier contrat concernant le robot ISIS installé à Chinon vient d'être signé avec la DER-IMA. D'autres projets sont en cours afin d'adapter le système actuel aux installations PWR. Sur ce chapitre, à signaler également « les liens importants noués avec Robosoft, pour l'industrialisation et la commercialisation de travaux menés au sein du laboratoire ».

A la pointe du combat...

Depuis 1979, date de naissance du projet, la cité des Sciences et de l'Industrie de La Villette a fait couler beaucoup d'encre en suscitant un intérêt certain de la part du public. Doté à sa création d'un budget d'investissement de 4,450 milliards de francs dont 60 % pour le seul bâtiment, cette structure – devenue dès 1985 établissement public – emploie quelque 900 personnes pour un budget d'exploitation estimé en 1986 à 600 millions de francs, dont 545 millions de subventions de l'Etat et 55 millions de recettes.

« Il faut démystifier les robots auprès du public et donner aux gens les moyens d'avoir un jugement personnel », explique Jean-Marc Vermorel, animateur de La Villette, persuadé de l'utilité d'une telle institution.

Recevant à la fois le grand public, les élè-

ves des classes techniques et les enseignants, le secteur robotique de La Villette, malgré sa jeune existence – quelques mois à peine –, bénéficie déjà d'une certaine notoriété... en attendant la consécration.

C'est ainsi qu'à l'entrée même de l'espace « Zoo des robots » installé au sein de l'exposition permanente Explora, s'élève une sculpture intitulée « le Bon Robot », œuvre originale de Gilles Roussi, qui a pour fonction de « faire passer le public de l'état de méfiance envers les robots à un sentiment de familiarité... » Issue de technologies de pointe, son « intelligence » fonctionne grâce à un microprocesseur réalisé avec le système de développement HP 64000 élaboré par Hewlett-Packard France.

Comme l'explique avec humour son créateur, « il parle de sa voix synthétique de robot, ne s'arrête jamais et soliloque pour se lancer parfois dans des réflexions à peine audibles. Capricieux, il peut même cesser tout dialogue avec le visiteur et sommeiller dans l'attente d'un climat plus favorable... »

En dehors même de l'anecdote, « le Bon Robot » schématise ainsi l'effort des responsables de la Cité des Sciences pour rendre sympathiques et familiers ces systèmes à un large public. Ni Goldorak ni Guerre des Etoiles, le monde des robots peut donc aider à une meilleure compréhension de ce que sera la physionomie de l'entreprise et du monde de demain.

Comme le note avec optimisme Jean-Marc Vermorel, la robotique bénéficie, au sein du parc de La Villette, d'un avantage lié à la nature même de ces technologies : secteur évoluant sans cesse, ce département de la Cité des Sciences ne peut être qu'à la pointe du combat pour le progrès.

Yves Offer

CODE MSI 3615
TOUTE LA MICRO-INFORMATIQUE
DANS VOTRE MINITEL

VARISOLVER

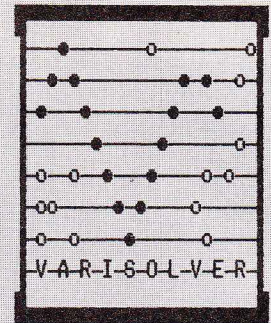
La Solution !

Le solveur "TEMPS REEL" pour Compatible PC et CPC Amstrad

Electricité, équations, mécanique, Back-solving, Finances, simulation en temps réel, systèmes non linéaires, études de fonctions, sciences et techniques au bout des doigts. A l'aide de variateurs le modèle de votre choix évolue sous vos yeux, la courbe se construit à l'écran, les valeurs sélectionnées sont lues et écrites sur le disque et sur le port série, les résultats défilent sur l'imprimante. Du plus simple au plus complexe les modèles sont stockés sur le disque et constituent une bibliothèque accessible et évolutive en fonction de vos besoins.

Variateurs + Solveur = VARISOLVER

le Laboratoire Numérique



Version disponible sur :

- Compatible PC standard CGA 490 F
- CPC AMSTRAD 464,664 6128 380 F
- APPLE II //e, //c nous consulter
- Prix TTC + frais de port 15 F
- Documentation sur demande

- injecteurs
- boucle auto
- clavier <->
- port série
- 4 fichiers disques

MODE CALCUL
+-----+
_ VARIABLES
_ . . .
+-----+
MODELE

- écran
- graphique
- imprimante
- port série
- 4 fichiers disques

EQUASIM Les Chapelles Bourbon 77610 Fontenay Trésigny Tél (1) 64 25 48 31

SERVICE-LECTEURS N° 279

SYSTÈME DE DÉVELOPPEMENT POUR INTEL* SUR IBM-PC/XT/AT*

Tout ce dont vous avez besoin pour créer des applications en temps réel pour circuit INTEL* sur IBM PC/XT/AT* :

- Système d'exploitation ISIS avec transfert de fichiers depuis le MDS au PC.
- Compilateurs/Assembleur/Éditeur de liens PL/M C Pascal sous Dos.
- Système d'émulation en temps réel **HITEX**
 - pour processeur 8051/52 - 8085 - 8086/88 - 80186/188
 - mémoire d'émulation de 256 kB
 - mémoire symbole sans limite
 - « HLL-débugger » pour PL/M, Pascal, C, ...
 - manipulation aisée par fenêtrage et système de menu.
- Programmeur universel pour PAL, EPROM, composants monocircuit.



Tout ce dont vous avez besoin pour créer des applications en temps réel pour circuit INTEL* sur IBM-PC/XT/AT.



COMPUTER ACCESS SYSTEMS

Mini Parc Bâtiment 7 - 6, av. des Andes - ZA de Courtaboeuf - 91952 LES ULIS Cedex - Tél. : 69.07.85.64

JE DÉSIRE : RECEVOIR UNE DOCUMENTATION
 ASSISTER A UNE DÉMONSTRATION CHEZ LE REVENDEUR LE PLUS PROCHE

SOCIÉTÉ :
NOM :
ADRESSE : TÉL :

MS 12/87

SERVICE-LECTEURS N° 280



PARIS 8^{eme}
 Vente par correspondance
 34, rue de Turin
 75008 PARIS
 16 (1) 42 93 47 32
 Métro : Place de Clichy, Liège

PARIS 12^{eme}
 60, cours de Vincennes
 75012 PARIS
 Métro : Porte de Vincennes

PARIS 15^{eme}
 près de la Porte de Versailles
 44, rue Cronstadt
 75015 PARIS
 Métro : Convention

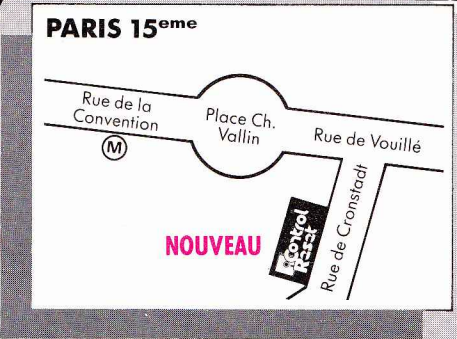
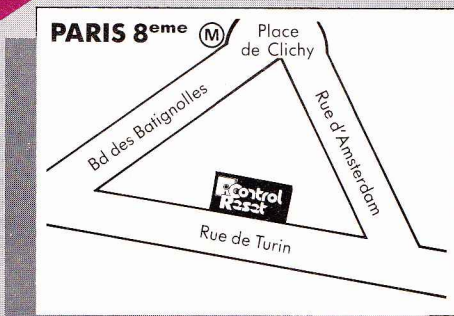
Conditions de vente
 Pour éviter les frais du contre-remboursement, nous vous conseillons de régler vos commandes intégralement (y compris frais de port).
 Forfait de port 40 F jusqu'à 5 K, au-delà nous consulter. Nos prix sont TTC.
 *marques déposées, photos non contractuelles.

Nos boutiques sont ouvertes du Mardi au Samedi de 10 h à 19 h sans interruption. Nos produits sont garantis 1 an.

LE PÈRE NOËL

ouvre le 30 Novembre 1987

2 nouveaux points de vente
 au cœur de PARIS



PARIS 15^{eme}
 près de la Porte de Versailles
 44, rue Cronstadt
 75015 PARIS
 Métro : Convention

PARIS 12^{eme}
 60, cours de Vincennes
 75012 PARIS
 Métro : Porte de Vincennes

recherchons
 techniciens de maintenance
 bon niveau : BTS + 5/6 ans
 ☎ (1) 42 93 47 32

★ **EN CADEAU** 1 Souris Graphique aux 5 premiers clients***

★ **EN CADEAU** 10 Disquettes Goldstar aux 15 clients suivants***

★ **JUSQU'AU 30 DÉC.** : Disque Dur 20 Mo TANDON* **1990^F ttc**

★ **EXCEPTIONNEL** : Business card TANDON* 20 Mo **2980^F ttc**

DES CENTAINES D'AUTRES CADEAUX

***pour un minimum d'achat de 400 F ttc

ROBOTIQUE MUSICALE, LA TRIPLE FONCTION: IA, ROBOTIQUE, INDUSTRIE

A Tokyo, les étudiants de l'université de Waseda développent chaque année des projets de robotique musicale. Au-delà de l'aptitude d'un robot à lire une partition musicale, et à jouer du piano, en accompagnant un chanteur qui improvise, c'est tout un secteur industriel qui est concerné : celui de la robotique intelligente.

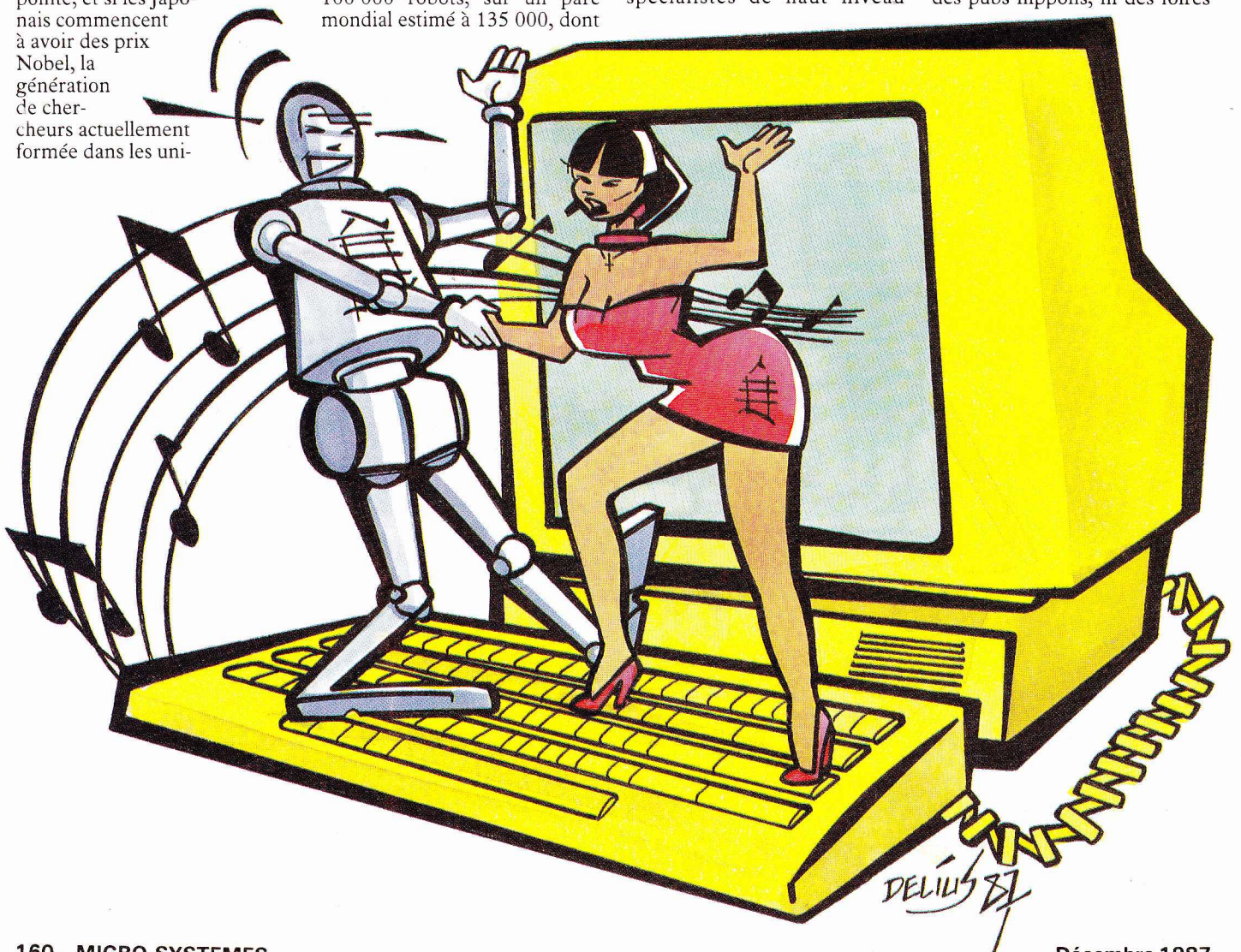
La cinquième génération d'intelligence artificielle japonaise vise la domination d'un marché mondial. Comme Jean-Jacques Servan-Schreiber l'analyse dans *le Défi mondial 86* (coll. Poche), l'esprit samouraï s'exprime désormais dans la conquête des marchés de pointe, et si les Japonais commencent à avoir des prix Nobel, la génération de chercheurs actuellement formée dans les uni-

versités nippones ne fera qu'accentuer cette tendance. La liaison industrie-université, les équipements d'avant-garde, les meilleurs professeurs, rien n'est laissé au hasard pour développer les secteurs de la robotique et de l'IA. Dans cette course à la rentabilité, le Japon était déjà, en 1984, premier en nombre de robots installés : 100 000 robots, sur un parc mondial estimé à 135 000, dont

6 000 en Allemagne, 2 000 seulement en France, et 8 000 à peine aux Etats-Unis. Début 1986, le chiffre atteignait 150 000 pour le seul Japon. Les premières usines sans ouvrier de Toyota, à Aïchi, y atteignaient leur plein rendement, commençant à créer des problèmes de reconversion, sauf... en robotique, où la demande en spécialistes de haut niveau

continuera longtemps à dépasser l'offre.

A Waseda, ce sont des équipes pluridisciplinaires qui forment les étudiants à la conception de robots sophistiqués. Si la série des Wabots semble destinée à l'accompagnement musical et à l'improvisation, le but recherché n'est certainement pas de réchauffer l'ambiance des pubs nippons, ni des foires



THÈME DU MOIS

internationales, même si les meilleurs programmes se permettent d'accompagner, au pied levé, des chanteurs qui n'arrivent pas à décrocher le robot musical. A travers la multiplicité des techniques mises en œuvre, ce sont les principes qui sont visés. De même que les équipes de Carnegie-Mellon attaquent les « décisions en univers incertain » par le biais du jeu d'échecs, de même ici, le domaine musical sert de point de départ à des études sur la modélisation des futurs possibles, à partir d'analyses rapides sur l'instant présent. Si l'on y connaît la musique, plus encore ce sont les algorithmes sophistiqués qui y sont développés, afin de gérer l'incertitude.

A l'origine, le projet Wabot était destiné à coordonner quatre départements distincts de l'université de Waseda.

Le Pr Sadamu Otheru dirigeait l'aspect vision et systèmes associés. Le système parole synthétique et reconnaissance vocale était représenté par le Pr Katshuhiko Shirai. La partie mouvement et structure des membres était confiée au Pr Ichiro Kato. Enfin, la partie purement musicale, avec ses aspects de prospective, était prise en charge par le Pr Seinosuke Narita. Le projet recevait à juste titre le nom de Wabot, d'une contraction de Waseda et de... robot.

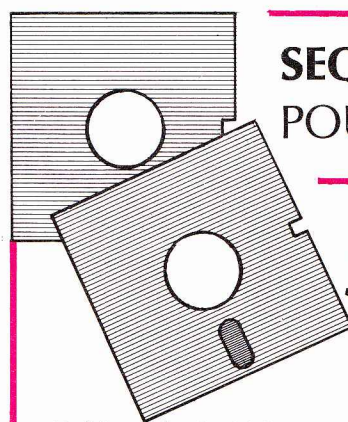
Parmi les performances que la série Wabot a progressivement développées, figure l'aptitude à lire une partition musicale codée de façon classique, et placée au dernier moment devant les yeux du robot. Celui-ci l'étudie alors quelques instants, par le biais d'une caméra vidéo, et d'algorithmes de reconnaissances de formes. Le robot, étant conçu de façon anthropomorphique, bouge et se déplace. Mais dans un premier temps, la lecture en mouvement posait des problèmes relativement ardues. C'est pourquoi les premiers modèles de Wabot ne permettent pas de prendre en compte les mouvements éventuels du robot en cours de lecture. Les premiers algorithmes se concentrent sur les problèmes posés par les imperfections de l'éclairage, qui n'est que rarement uniforme. Des paramètres vont contrôler les irrégularités, et ajuster la vision

en fonction d'un ensemble de cadres découpant la partition en 64 par 64 sous-ensembles. La partition peut être lue à un mètre de distance. Le robot prépare alors la séquence des gestes qui suivront. Au moment de jouer, grâce à ses deux bras munis de chacun dix doigts, Wabot frappe les touches du piano avec délicatesse, tout en continuant à suivre la partition « dans sa tête », puisque celle-ci est seulement mémorisée. De même, Wabot sera capable d'utiliser ses pieds pour actionner les pédales de son piano. Pour cette première étape, aujourd'hui dépassée, le système exigeait cependant une définition de 2 000 x 3 000 pixels.

Parmi les morceaux qu'il est capable de déchiffrer, on trouve aussi bien du folklore que du classique, et notamment des œuvres baroques. A ce niveau, c'est l'aspect robot anthropomorphique capable d'accomplir des tâches physiques rapides et délicates qui attire l'attention. Un soin tout particulier a été apporté à l'étude des jointures, de façon à permettre une souplesse, et un coulé des gestes. Dès les versions de 1984, Wabot était capable de frapper les touches au rythme de quinze fois par seconde. Peut-être lui manque-t-il aujourd'hui un peu de « feeling » lorsqu'il joue du Bach. J'aimerais cependant savoir s'il est capable de transposer un concerto de Mozart en blues, en signe de nostalgie face aux difficultés japonaises aux Etats Unis...

Le Pr Ichiro Kato est particulièrement fier des sept degrés de liberté que ses étudiants ont réussi à donner aux bras de Wabot, qui définissent les mouvements de l'épaule, du coude, et du poignet. Chaque bras est contrôlé par un ordinateur indépendant, et par un ensemble de seize microprocesseurs. Par ailleurs, un des problèmes rencontrés a été l'harmonisation du programme de partition avec la gestion des gestes du pied, destinés à gérer la qualité du son, et l'expressivité du morceau. A partir de là, tout un ensemble de projets ont été définis par Sumitomo Electric, Yamaha, et Nippon Electric, projets qui, eux, n'ont plus rien à voir avec la musique...

Jacques de Schryver



SEQUENTIEL INDEXE POUR QUICK BASIC*

595 f. ttc

- Fichiers de 64.000 enregistrements
- 1 à 255 champs par enregistrement
- 1 à 5.000 caractères par enregistrement
- 12 clés triées en temps réel par fichier (B-Tree)
- Clé reproductible ou non reproductible
- Gestion automatique des suppressions
- Recherche par clé ou portion de clé
- Gestion des erreurs.

Outre le séquentiel indexé les **outils de développement** comprennent:

- Scroll ascendant et descendant
- Choix d'un fichier dans un répertoire
- Puissant éditeur de zones (pour nombres, dates et chaînes)
- Gestion des menus et des questions à choix multiples.

Utilisation très simple des outils, au moyen d'instructions :

CALL <Procédure [(paramètres)]>

- Livré avec un manuel en français, des exemples de programmes et tous les utilitaires d'initialisation et de régénération des fichiers et des index.
- Pas de royalties sur les applications développées.
- Assistance téléphonique.

SOMMA

France

3, rue Ruhmkorff
75017 PARIS

Tél. : (1) 45 72 17 38 +
Télex : 642 255

BON DE COMMANDE

MS 12/87

Nom : _____

Société : _____

Adresse : _____

Ville : _____

- Outils de développement 595 F TTC pour Quick Basic 2.01 pour Quick Basic 3.0
- Quick Basic 1000 F TTC version 2.01 (français) version 3.0 (anglais)
- Supplément pour outils réseau + 200 F TTC _____
- TOTAL par chèque joint
- Je désire recevoir une documentation

*Quick Basic est une marque déposée Microsoft.

APPLE IIGS

Le nouvel APPLE IIGS "graphique/son"

- Microprocesseur 65C816, 16 Bits à Emulation du 65C02
- 256K de Ram, extensible à 8 Mo
- Clavier détachable Azerty avec Pavé Numérique
- Souris/Graphisme Hte Résolution 600 x 200 en couleurs et 320 x 200 en 16 couleurs parmi 4096
- Son 16 voies/Synthèse Vocale/Horloge Temps Réel
- Compatible avec la majorité des Logiciels Ile/Ic

Configuration DUO

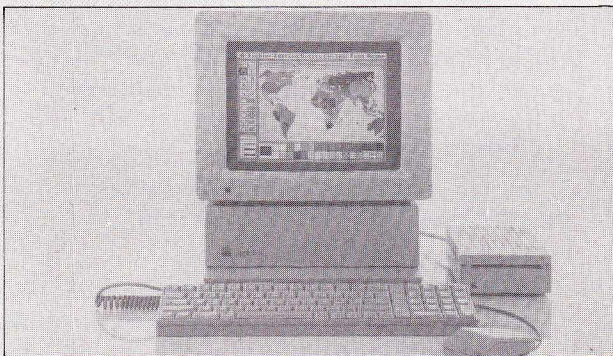
- 1 Apple IIGS 512k Ram
- 2 lecteurs 3,5 - 800k
- 1 Moniteur Couleur RVB Hte Déf.
- Logiciels GS Write + GS Paint

15 900 F TTC

Configuration Disque Dur

- 1 Apple IIGS 512k Ram
- 1 lecteurs 800k
- 1 Moniteur Couleur Hte Déf.
- 1 Disque dur 20 Mo + carte SCSI
- Logiciels GS Write GS Paint

19 900 F TTC



PRIX EXCEPTIONNEL DE L'ENSEMBLE

LOGICIELS

| | | |
|--|----------|-------|
| AppleWorks 1.4 (T.T/Gest. Fichiers/Tableurs) | 2 190,00 | F TTC |
| Arts Parts Volume 1 ou 2 (Librairie d'images) | 350,00 | F TTC |
| Bard's Tale I (Jeu d'aventures) | 450,00 | F TTC |
| Caverne Cobra (Jeu d'Arcade) | 450,00 | F TTC |
| De Lux Paint II (Graphique et animation) | 860,00 | F TTC |
| Draw Plus (Metteur en Page) U.S.A. | 990,00 | F TTC |
| Famous Course Disk (Parcours Golf), | 320,00 | F TTC |
| Fantavision animation graphique | 750,00 | F TTC |
| GS/Paint (Graphique) | 990,00 | F TTC |
| GS/Write (Traitement de texte) | 990,00 | F TTC |
| Grapholies (Librairie d'images) | 245,00 | F TTC |
| GraphicWriter (Graphique et traitement de texte) | 1 750,00 | F TTC |
| GS/File VF (Base de données) | 1 790,00 | F TTC |
| Hacker II (Jeu d'aventure) | 390,00 | F TTC |
| Kid'Stime II (Logiciel d'éducation musical) | 425,00 | F TTC |
| MultiScribe | 1 100,00 | F TTC |
| Memsoft Progr. (Logiciel de développement) | 3 440,00 | F TTC |
| Music Construction Set (Composition musicale) | 490,00 | F TTC |
| Music Studio V.F (Composition musicale) | 711,00 | F TTC |
| Music Studio Version U.S.A. | 650,00 | F TTC |
| Mean 18 (Golf) | 425,00 | F TTC |
| Notes.N.Files (Fichiers et Texte) | 1 650,00 | F TTC |
| Orca/Macro Assembleur | 1 290,00 | F TTC |
| PaintWorks Plus (Graphique) U.S.A. | 990,00 | F TTC |
| Reader Rabbit 5 à 7 ans (Educatif) | 550,00 | F TTC |
| PrintShop | 750,00 | F TTC |
| Sea Stricke (Jeu d'Arcade) | 450,00 | F TTC |
| Senior 2.0 (Gestion générale) | 6 580,00 | F TTC |
| Shangai (Jeu du Mah Jong) | 295,00 | F TTC |
| Silent Service (Bataille navale sous-marine) | 425,00 | F TTC |
| Tass Time (Jeu d'aventures) | 295,00 | F TTC |
| Thexder (Jeu d'arcades) | 590,00 | F TTC |
| Tower of Myraglen (Jeu d'aventure) | 450,00 | F TTC |
| TML/Pascal APW | 1 290,00 | F TTC |
| TML/Pascal Version S.Alone (Souris) | 1 600,00 | F TTC |
| Unimate (Utilitaire) 5 1/4 en 3,5 | 595,00 | F TTC |
| VS/Com (Logiciel de communication) | 990,00 | F TTC |
| VS/File (Gestion de base de données) | 1 100,00 | F TTC |
| VS/Draw (Metteur en Page) | 990,00 | F TTC |
| V.I.P Professionnel Version U.S.A | 2 860,00 | F TTC |
| Visualizer (Grapheur Couleur) | 1 100,00 | F TTC |
| Word Perfect (Trait. de Texte avec 115 000 mots) | 2 250,00 | F TTC |
| 816/Paint (Education Graphique) | 595,00 | F TTC |

CARTES ET PÉRIPHÉRIQUES, IIGS

| | | |
|--|-----------|-------|
| Apple Tell Carte Modem | 4 150,00 | F TTC |
| Carte CR8C Carte de programmation d'automate | | NC |
| Carte Speedisk 1 Méga à mémoire permanente | 5 500,00 | F TTC |
| D.G.A 1 Carte de digitalisation Vidéo | 5 300,00 | F TTC |
| GS/Ram Plus 2 Méga Extensible à 8 Méga | | NC |
| Kurta Tablette à digitaliser Graphique | 5 500,00 | F TTC |
| Méga-Core 20 Méga Disque Dur Interne | 15 300,00 | F TTC |

| | | |
|---|----------|-------|
| MégaStore 20 Méga SCSI Disque Dur | 6 900,00 | F TTC |
| MIDI Carte Synthétiseur | | NC |
| Penman Traceur autonome | 5 690,00 | F TTC |
| SCSI Carte SCSI pour disque Dur | 1 200,00 | F TTC |
| Supersonic Digitalizer Carte digitalisation Son | 750,00 | F TTC |
| Supersonic Stéréo Carte Stéréo | 750,00 | F TTC |
| Z80 Carte avec CP/M 5-1 + utilitaires | 1 550,00 | F TTC |
| Z80 Carte sans CP/M | 990,00 | F TTC |
| Carte Extension Mémoire 256 K | 1 150,00 | F TTC |
| Extension 256 K Ram pour Cte Extension | | PROMO |
| Carte Extension 1 Mo à 8 Mo | 420,00 | F TTC |
| Carte Apple Talk pour I-III | 1 400,00 | F TTC |
| Carte Serial-Grappier (Interface série + Recopie Ecran) | 1 450,00 | F TTC |
| Carte AD/DA (2 Voies/12 Bits) | 4 100,00 | F TTC |
| Câble Périel IIGS | 250,00 | F TTC |
| Câble Adaptateur lecteur 5" 1/4 | 150,00 | F TTC |
| Lecteur 3" 1/2 Apple | 2 300,00 | F TTC |
| Lecteur 5" 1/4 Apple | 1 900,00 | F TTC |
| Disque Dur 20 Mo SCSI (Mécanique Seagate) | | |
| Prix Club | 5 900,00 | F TTC |

APPLE IIC

Le nouvel APPLE IIC est maintenant livré chez Microshop avec une mémoire étendue à 1Mo pour le prix de 384K!!!



Configuration UNO

- Garantie Totale 1 an
- 1 Apple IIC 1 Mo
- 1 Moniteur Apple Vert + Support
- 1 Souris
- 1 Joystick
- 1 Boîte de Disquettes Logiciels

OPTIONS : Lecteur Supplémentaire 3"1/2 ou 5"1/4

APPLE IIE

Pour tout achat d'une configuration Apple IIGS, Microshop vous reprend votre unité centrale II + IIE pour : 1 500 F



OPERATION REPRISE IIE/IIC

Cette opération étant soumise à certaines conditions consultez-nous dès maintenant.

LOGICIELS

| | | |
|--|----------|-------|
| Tous ces logiciels se font en version 3"1/2 ou 5"1/4 | | |
| Apple Works 1.4 | 2 190,00 | F TTC |
| Gestion II (Gestion Club,Association) | 1 190,00 | F TTC |
| Epistole (Traitement de Texte Souris) | 1 180,00 | F TTC |
| Version Calc (Tableur Souris) | 1 180,00 | F TTC |
| Version Tel II (Communication Souris) | 1 180,00 | F TTC |
| Graph Works (Graphiques Apple Works) | 995,00 | F TTC |
| Apple Works 1.4 | 2 190,00 | F TTC |
| Apple Logo II | 950,00 | F TTC |
| Pascal 1.3 | 1 790,00 | F TTC |
| Turbo Pascal 3.0 | 790,00 | F TTC |
| Instant Pascal | 1 100,00 | F TTC |
| Turbo Pascal Tool Kit | 990,00 | F TTC |
| Turbo Tutor | 350,00 | F TTC |
| Mouse Desk (Bureau Souris) | 295,00 | F TTC |
| Extasie (Dessins Graphiques/Souris) | 695,00 | F TTC |
| Epistole (128 k/Souris) Traitement de texte | | |
| Souris | 1 180,00 | F TTC |
| Version Calc (Tableur) | 1 180,00 | F TTC |
| Easy Puss (Gestion de Fichiers Souris) | 1 390,00 | F TTC |
| Procode (Assembleur) | 950,00 | F TTC |
| Version Liste (Gestion de Listes) | 695,00 | F TTC |
| Print Shop (Utilitaire Graphique) | 550,00 | F TTC |
| Dazzle Draw (Graphique) | 550,00 | F TTC |
| Fantavision (Animation Graphique) | 695,00 | F TTC |
| Flight Simulator (Simulation Aviation) | 530,00 | F TTC |
| Winter Games (Jeu Olympiques) | 350,00 | F TTC |
| Summer Games (Jeu Olympiques) | 350,00 | F TTC |
| Karateka (Jeu) | 395,00 | F TTC |
| Ultima IV (Jeu Aventures) | 695,00 | F TTC |
| Gato (Jeu sous-Marin) | 495,00 | F TTC |
| Pin Ball Construction Set (Flipper) | 440,00 | F TTC |
| Silent Service | 420,00 | F TTC |
| F15 Strick (Simulation aviation) | 450,00 | F TTC |

MONITEURS

| | | |
|--|----------|-------|
| Moniteur Goldstar (12" Vert ou Ambre/22 Méga) | 990,00 | F TTC |
| Moniteur Philips (14" Couleur Hte Définition + Périel) | 2 900,00 | F TTC |

UN MONOCHIP COMPATIBLE 68000 LE SCC 68070 DE RTC

Fabriqué en technologie C-MOS, le SCC 68070 regroupe en un seul boîtier une unité centrale compatible 68000 et un ensemble de périphériques couramment utilisés dans les applications à hautes performances, où les coûts d'études et de fabrication restent prédominants.

Le processeur du SCC 68070 a une architecture compatible avec celle du 68000 (jeux d'instructions, de registres...) de façon à assurer une compatibilité logicielle totale. Mais il existe quelques différences, comme le traitement des erreurs de bus (BUSERROR) et d'adressage qui s'apparente à celui du 68010. D'autre part, le compteur de programme (PC) est sur 32 bits, ceci permettant d'adresser l'ensemble des périphériques internes dans une autre page que l'espace proposé par le 68000. Les périphériques intégrés n'introduisent pas de nouvelle instruction ; ils sont en effet adressables comme des positions mémoires (registres équivalents connus comme ceux des 68681, 68440, 68230...) directement sur la puce en mode superviseur, et situés en dehors de l'espace mémoire externe de 16 méga-octets. Ces périphériques sont :

- une unité de gestion mémoire (MMU) orientée « segment » avec huit descripteurs internes (idéal pour les systèmes multitâches temps réel) de 2 Mo chacun ou avec 128 segments, dont 120 externes de 128 Ko chacun ;
- deux canaux d'accès direct mémoire (DMA) totalement indépendants, capables de faire des transferts jusqu'à 1,6 méga-mots de 16 bits par seconde entre périphériques et mémoires pour les deux canaux (simultanément) et entre mémoires pour le canal 2 (mode paquet jusqu'à 128 Ko) ;
- un canal de communication asynchrone (UART) incluant un générateur de fréquences séparé pour l'émetteur et pour le récepteur (16 vitesses différentes de 75 à 19 200 bauds) ;
- trois fonctions compteur/temporisateur dont l'une (TO) est un compteur temps réel sur 16 bits à rechargement automatique, qui assure la commutation des tâches

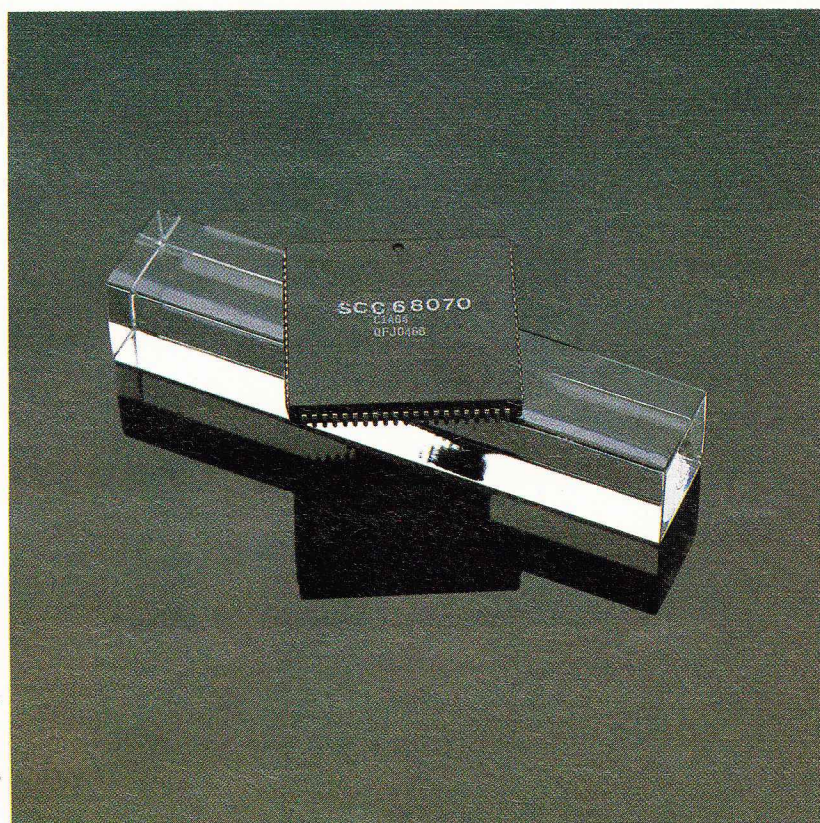


Photo J.-M. Aragon

dans les systèmes multi-utilisateurs. Les deux autres compteurs 16 bits (T1 et T2) peuvent, associés à deux broches du boîtier, fonctionner en comptage d'événements, en mode capture ou en décomptage de temps ;

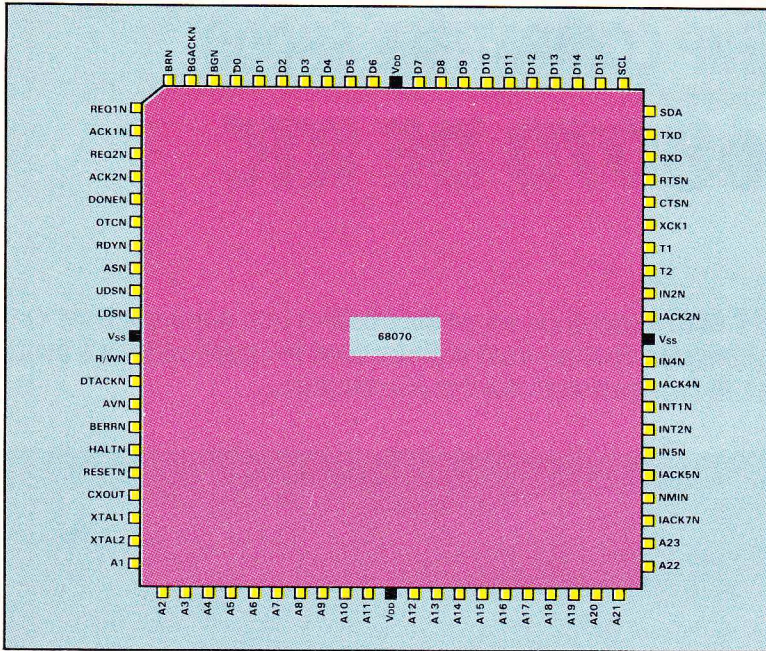
- une interface au bus 12C permettant des transferts jusqu'à 100 K-bits par seconde sur un bus bifilaire intégrant l'arbitrage multi-maître. De nombreux circuits sont disponibles, qui comportent tous l'interface 12C : microcontrôleurs N-MOS et C-MOS, RAM C-MOS, EEPROM, circuits d'E/S analogique et digital, circuits spécialisés en téléphonie, audio et vidéo. Il faut noter que sur le SCC 68070 le clavier

et la souris sont connectés sur le bus 12C, laissant libre la ligne de communication asynchrone pour interfacer un modem (y compris les signaux RTS et CTS) ou un autre ordinateur ;

- une horloge intégrée (oscillateur à 20 MHz, horloge distribuée de 10 MHz de façon interne et externe) et une entrée pour l'UART à 3,6864 MHz permettant le fonctionnement aux vitesses standard. Il est possible aussi de n'utiliser qu'un seul quartz à 19,6608 MHz servant toutes les fonctions ;
- une logique d'interruption intégrant la circuiterie nécessaire à la génération et à l'acquittement des requêtes. Outre les sept autovec-

RTC - SCC 68070





Brochage du SCC 68070.

teurs internes (57 à 63) programmables sur tous les niveaux et servant les périphériques internes, quatre niveaux d'interruption externe (2, 4, 5 et 7) sont accessibles de manière entièrement décodée (IACKN séparés). Deux lignes supplémentaires d'interruption, programmables sur les niveaux 1 à 7, sont également disponibles en mode autovectorisé (nécessaire aux applications avec le bus 12C lorsqu'elles génèrent des interruptions).

Présenté dans un boîtier 84 broches PLCC pour le montage en surface, le SCC 68070 ne consomme que 10 mA sous 5 V à 10 MHz.

Un circuit associé, le SCC 66470-VSC, intègre la logique nécessaire au séquençage du RESET, un décodage mémoire pour des E/S, des EPROM, et pour le contrôle de mémoire dynamique intégré et un contrôleur d'écran orienté bit ainsi qu'une interface coprocesseur.

J.-L. Baudoux

Pour plus d'informations cerchez 79

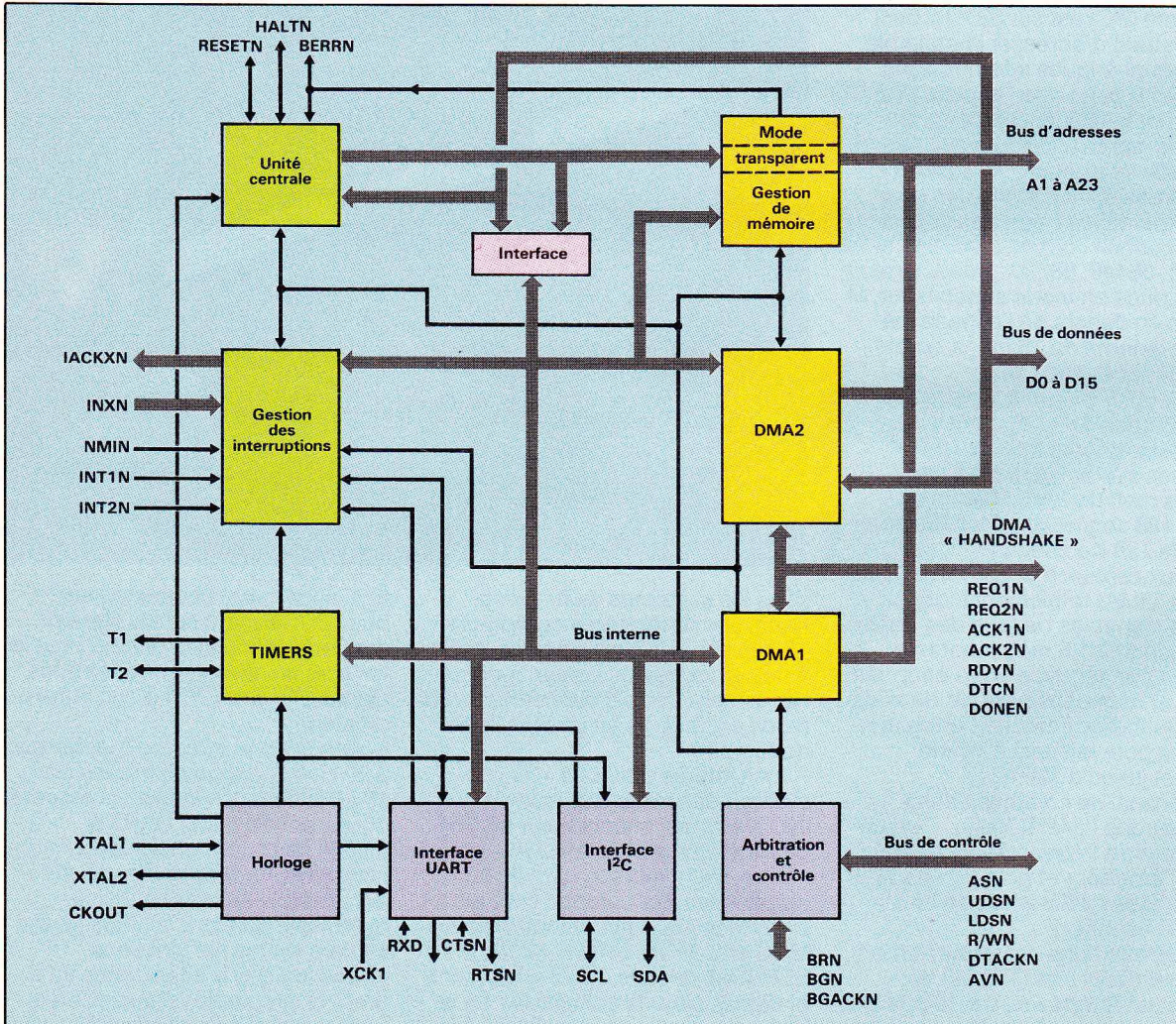


Schéma bloc du SCC 68070.



LA PROGRAMMATION SANS PANNE (4)

100 MILLIARDS D'INSTRUCTIONS

Quel ne fut mon étonnement lorsque je reçus les cartouches magnétiques commandées à X..., un des fabricants d'ordinateurs les plus connus. J'avais commandé une boîte, on m'en a livré deux et facturé trois !

La secrétaire se confondit en excuses et m'expliqua : c'est le nouveau programme de gestion des stocks mis en place récemment qui est à l'origine de cette erreur, mais tout rentrera dans l'ordre bientôt.

Tout ceci n'est finalement pas très gênant pour les clients de X..., mais, pour son directeur financier, le choc est plutôt désagréable : les frais à engager afin de remettre les choses en place sont de l'ordre de plusieurs centaines de milliers de francs ! Il faut d'abord déterminer l'ampleur des dégâts, faire retourner les pièces envoyées en trop, livrer celles « oubliées » par l'ordinateur, corriger le ou les programmes « fautifs », sans oublier les lettres d'excuses aux clients avec les rectificatifs comptables. Sans compter le préjudice moral, le temps énorme perdu... Bref, cette faute de programme, on aurait pu s'en passer...

Beaucoup d'argent qui flotte

Peut-on évaluer les sommes d'argent perdues dans des situations pareilles ? Sans doute des milliards.

La Terre comporte vingt instructions par habitant dont seize écrites en Cobol. Et de la façon dont le sujet est abordé, une partie de ce patrimoine constitue plus un boulet que l'on traîne qu'un sujet de fierté. Ce dernier article sur la PSP se termine néanmoins sur une note optimiste, après avoir dévoilé un côté de l'informatique aussi vrai qu'insoupçonné.

Mais plus angoissante encore est la question suivante : Peut-on être sûr que ce genre de situation ne se renouvellera pas ? Quelles garanties le logiciel offre-t-il ?

Pour le moment, il n'existe aucune règle rationnelle permettant de juger les risques de panne logiciel. Ce n'est pas le cas du hardware : on peut connaître le MTBF* d'un circuit intégré dès les premières heures de test. Ainsi, le constructeur de matériel électronique peut offrir une garantie globale de fonctionnement. Quant au logiciel, c'est plutôt la corde raide : à la moindre secousse, c'est la chute libre.

Ceci est malheureusement vrai même au niveau des Systèmes d'Exploitation des grands ordinateurs. Non pas que ceux-ci ont des « bugs », mais beaucoup d'entre eux ont à peine changé depuis plus de quinze ans, alors que le matériel sur lequel ils tournent a évolué à la vitesse grand V. Ceci est dû probablement à la prudence des constructeurs, qui

savent qu'un changement de structure des logiciels de base est extrêmement coûteux. Pour le client aussi, d'ailleurs...

Cette remarque est encore plus vraie pour les Systèmes d'Exploitation des micro-ordinateurs : chacun se rappelle la malheureuse expérience du MP/M qui était censé succéder au célèbre CP/M.

Les pistes sont brouillées

Comment alors se débrouillent les SSII, chargées de produire tous les jours du logiciel ? Tout le monde connaît les différentes phases, classiques dirions-nous, par lesquelles passe un logiciel d'application :

- analyse de l'existant et faisabilité,
- analyse fonctionnelle,
- analyse organique,
- programmation,
- test,
- livraison partielle puis finale.

Lorsque le logiciel est de taille modeste, cela se passe

plus ou moins bien. Mais dans le cas d'applications plus importantes, la livraison finale traîne souvent. Et ceci est généralement dû à un manque dans l'analyse initiale. La responsabilité n'est parfois pas évidente à cerner : l'utilisateur est accusé de ne pas avoir fourni tous les renseignements suffisants, et la SSII est, elle, accusée de ne pas avoir approfondi son étude et de vouloir brûler les étapes. Mais l'utilisateur n'est pas un surhomme, et la SSII ne peut pas tout deviner dans les délais.

Goliath

Certains informaticiens ont voulu approfondir au maximum la phase d'analyse : il faut étudier dans le détail toutes les données susceptibles d'avoir une importance. Des tonnes de papier décrivant les schémas « logiques », « physiques » puis « conceptuels » des données avec leurs corrélations sont à produire. Ces papiers doivent être examinés par l'utilisateur pour approbation.

A notre avis, cette façon de procéder a toutes les chances d'échouer : elle est extrêmement longue à mettre en œuvre et, pendant ce temps, l'environnement même de l'étude risque de changer. Ainsi, on se retrouverait dans le cercle vicieux de vouloir figer quelque chose qui a déjà bougé ! Ensuite, elle est aussi extrêmement coûteuse parce qu'elle demande beaucoup d'efforts à beaucoup de monde. Enfin, elle a le principal défaut de ne pas se mettre au niveau de l'utilisateur qui, fi-

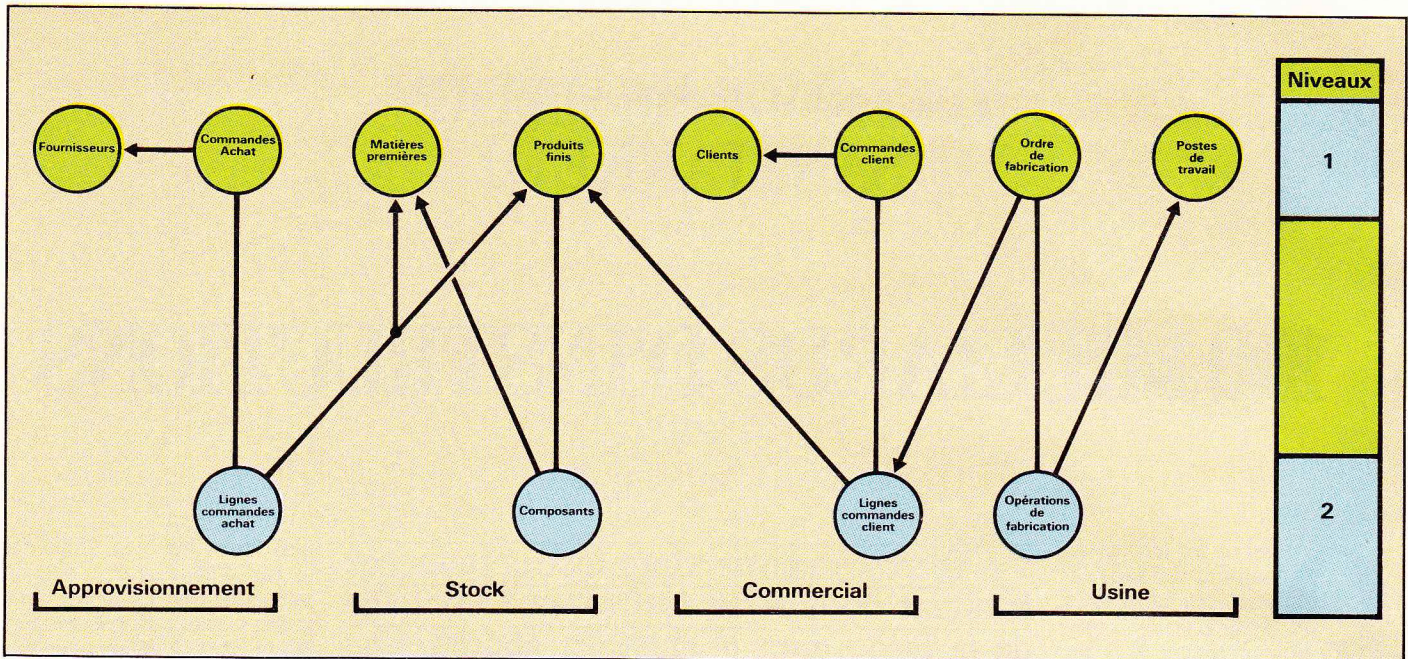


Fig. 1. - Graphe d'Objets. Cas simplifié Commerce & Industrie.

nalement, est assez loin de la technique informatique.

... et David

L'autre réaction serait de devancer l'utilisateur. En d'autres termes :

- faire une analyse minimale des besoins ;
- développer rapidement des programmes fiables ;
- si besoin est, corriger la trajectoire avec la même rapidité et la même fiabilité.

Ainsi, peu de temps après que l'utilisateur a formulé ses requis, il dispose d'un ensemble de programmes qu'il pourra évaluer. Il est encore temps de rattraper un éventuel oubli sans que l'impact sur les coûts soit important.

Toutefois, cela suppose que l'informaticien dispose d'outils lui permettant de développer rapidement et de manière fiable. Or, depuis peu, ces outils existent sur le marché. Nous en avons mentionné quelques-uns dans notre précédent article** : il s'agit des « outils experts », permettant de produire des programmes comme s'il s'agissait de Systèmes Expert. Le programmeur introduit l'algorithme sous forme de Base de Connaissances, et l'outil génère le programme correspondant.

Cette façon de procéder est

rapide, donc peu coûteuse, et surtout elle rend peu probable l'impact d'un éventuel changement d'environnement. De plus, le fait de coder l'algorithme sous forme de Base de Connaissances permet de rapprocher l'utilisateur de la technique informatique : contrairement à ce que l'on pourrait penser, un utilisateur comprend mieux des Règles de Connaissances se rapportant à son travail qu'un organigramme achevé.

Objets trouvés

La solution « David » mérite d'être développée en détails : en effet, c'est très beau de devancer l'utilisateur, mais comment ? L'informaticien est-il censé pouvoir comprendre tous les méandres de la logique des utilisateurs, à quelque métier qu'ils appartiennent ?

Avant de répondre à cette question, nous rappelons au lecteur les bases de la Démarche Structuraliste de la Programmation Sans Panne (PSP), décrite dans un précédent article*** : il s'agit de rechercher les objets à traiter, puis de déterminer les traitements subis par ces objets, et sous quelles conditions. La PSP définit les Objets comme étant les substantifs de la description d'une situation.

A titre d'exemple, reportons-nous au Graphe d'Objets dessiné à la figure 1. Les objets sont représentés par des ronds et sont liés entre eux soit par des flèches (préexistence), soit par des traits verticaux (décomposition). Ainsi, un Fournisseur préexiste à une Commande d'Achat, qui se décompose elle-même en Lignes.

Nous prétendons qu'un Graphe d'Objets peut servir de guide tout le long de la vie d'une application informatique. De plus, il peut être déterminé rapidement et au départ. En effet, il suffit d'un minimum d'expérience pour isoler les notions qui reviennent souvent dans la bouche du futur utilisateur. En outre, l'informaticien peut détecter assez tôt les « trous » ou les oublis de l'utilisateur rien qu'en examinant le graphe et en lui appliquant les règles élémentaires de cohérence. Enfin, les Objets préexistent à leurs traitements, ce qui facilite leur recherche.

Mais une utilité plus essentielle à notre avis est que tout utilisateur peut comprendre le graphe sans trop d'efforts. Ceci contribue à le rapprocher de la technique informatique sans qu'il ait une formation d'informaticien. Les éventuels malentendus sont ainsi réduits au minimum, de même que les coûts

qu'ils engendrent fatalement pour tout le monde.

Une fois le Graphe d'Objets construit, les traitements subis par ces Objets correspondent aux fonctions attendues de l'ordinateur. Celles-ci sont classées par Objet ou groupe d'Objets. Il faut retenir qu'un éventuel oubli au niveau du Graphe d'Objets aurait pour conséquence une revue de presque tous les programmes.

Ainsi, la réponse à la question ci-dessus se traduit par la négative : l'informaticien n'est pas supposé comprendre tous les méandres de la logique de tout le monde (heureusement). En revanche, il dispose d'un outil de dialogue simple permettant de fixer les idées une fois pour toutes. En ce sens, le Graphe d'Objets est au logiciel ce que le plan d'architecte est à la construction de bâtiments.

Ajoutons à cela les « outils experts » apparus récemment sur le marché, et voilà la fronde de « David » mise en place...

25 milliards de dollars

Cette façon d'aborder la production de logiciel permet de diminuer de manière importante la maintenance. Celle-ci est estimée à 25 milliards de

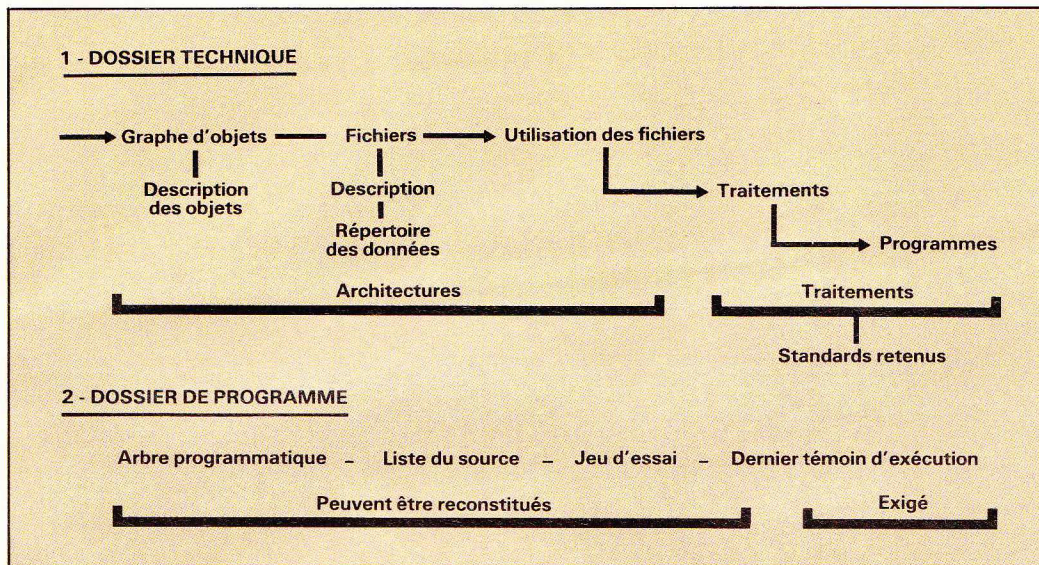


Fig. 2. — La documentation rationnelle.

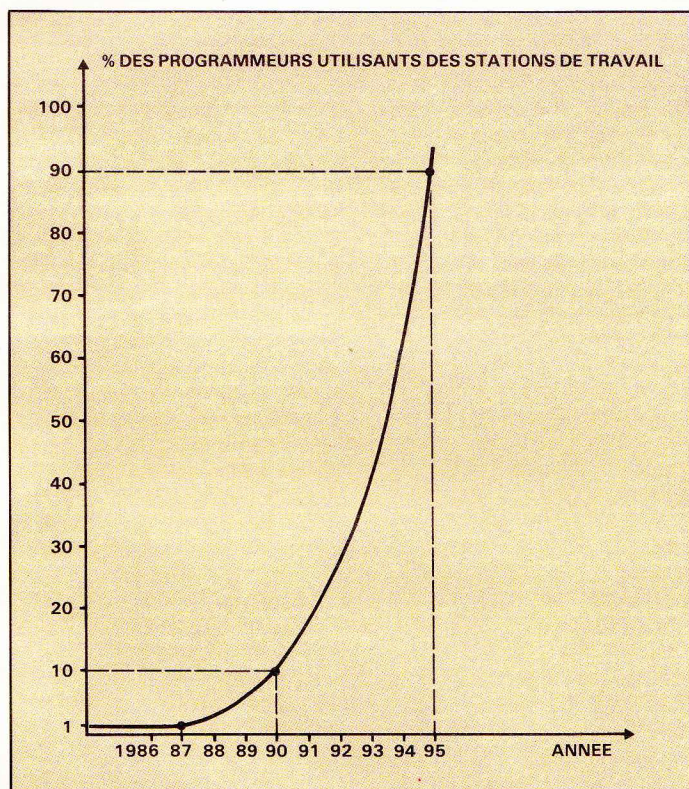


Fig. 3. — Pénétration des stations de travail (chiffres tirés de L'informatique Professionnelle », mars 1987).

dollars (soit 150 milliards de francs) pour les logiciels installés dans le monde. De quoi exciter plus d'un financier !

Nous voyons que les enjeux industriels de méthodes telles que la PSP sont loin d'être négligeables. Car il s'agit de reprendre partiellement ou totalement les programmes concernés par cette mainte-

nance. Inutile de dire que les résultats de cette reprise doivent être garantis, sinon...

Comment aborder ce marché ? Quelles en sont les caractéristiques essentielles ?

Sans prétendre le décrire exhaustivement, signalons les points suivants.

- L'utilisateur privilégiera la qualité à la sophistication.

- La maintenance doit être sinon nulle du moins maîtrisée totalement.
- Le logiciel doit prouver être évolutif.
- L'utilisateur exigera des moyens clairs de contrôler l'exécution des programmes.
- Il sera beaucoup plus près du technique.
- En revanche, il est prêt à payer le prix qu'il faut pour s'en sortir.

En un mot, ce marché mettra, en face de l'informaticien, un utilisateur aigri et échaudé par ses expériences antérieures, mais plus averti. Il sera très exigeant car il ne veut pas retomber dans les mêmes erreurs.

Pour le convaincre, il faut être plus qu'un programmeur astucieux : il faut des garanties palpables, des méthodes claires et des outils.

L'eau de roche

La clarté d'une méthode se mesure à sa facilité d'utilisation. Cette simplicité doit être apparente, tant du point de vue du technicien que de celui de l'utilisateur. Dans le cas de la PSP, cette clarté transparaît au niveau de la documentation du logiciel.

En effet, le problème de la documentation du logiciel remonte aux premiers temps de l'exploitation des ordinateurs. Il provient de la stricte ressemblance apparente entre un ordinateur programmé et un qui ne

est pas... Ajoutez à cela des programmeurs-astucieux-météores et laissant des traces difficilement déchiffrables par des non-initiés.

On a ainsi eu tendance soit à tout commenter dans le moindre détail, soit à se fier à des programmes autodocumentés, soit à un mélange savant des deux. Ce fut (et c'est encore souvent) le cauchemar des programmeurs : documenter avant, pendant ou après le développement du logiciel ? Et, ensuite, documenter quoi ? A quel niveau de détail ?

Une réponse possible à toutes ces questions est donnée par la PSP. Elle s'appuie sur les principes d'efficacité et d'utilité de la documentation. Celle-ci doit être :

- nécessaire et suffisante ;
- simple à mettre à jour ;
- à enchaînement logique.

Trois dossiers sont utilisés en PSP : le Dossier Technique, le Dossier de Programme et le Dossier d'Implémentation.

Le Dossier Technique part de la définition des objets et de leur représentation sous forme de graphe (voir fig. 1). Ensuite, on découpe les objets en fichiers. Puis on associe les traitements par fichier (ou groupe de fichiers), et ces traitements sont découpés en programmes. Enfin vient la description des standards de programmation, constitués par les règles que les programmeurs doivent tous suivre. Un répertoire (ou dictionnaire) des données couronne le tout. Il faudrait noter que toute description doit être succincte afin que l'exploitation du dossier soit efficace.

Cette description « linéaire » du logiciel, partant du global (objets) et aboutissant au détail (donnée) est une garantie pour la bonne compréhension du travail à faire, non seulement pour le technicien mais aussi pour l'utilisateur. Ce document servira de référence pour tout le monde.

Le Dossier de Programme sert de document de travail au programmeur. Pour chaque programme, quatre éléments au plus :

- la liste du source ;
- l'Arbre Programmatique ;
- le jeu d'essai ;
- le dernier témoin d'exécution .

DÉVELOPPEMENT

cellence ? Non, car si c'était le cas, seules les Rolls seraient des voitures de qualité ! Il serait plus juste de dire que la qualité se définit par rapport à des niveaux, et qu'il existe, pour chaque niveau, des points de repère permettant d'apprécier un produit. Ainsi, dans le cas d'automobiles, nous observons chez chaque constructeur des gammes de produits obéissant à des normes précises. Dans le cas où l'on voudrait comparer la qualité de deux voitures, il faudrait d'abord s'assurer qu'elles appartiennent toutes deux à une même catégorie.

Mais revenons à la qualité logicielle. Comment la cerner ? Comment définir les niveaux de qualité ? La réponse à ces questions deviendra évidente dès que l'on aura maîtrisé la notion de test d'un programme. Et là, les avis sont très mitigés. La célèbre réflexion du non moins célèbre Dijkstra disant qu'il était irréaliste de tester un programme reste ancrée dans les mémoires. L'idée, derrière cette réflexion, était que, pour tester tous les cas de figure d'un programme aussi simple que celui qui fait une multiplication, il fallait des dizaines d'années... Alors que serait-ce pour un programme moyen de 200 lignes !

Pourtant, en Programmation Sans Panne, il existe un moyen simple et efficace de tester un programme, même complexe. Non pas tous les cas de figure, mais toutes les structures du programme (voir fig. 4 et les deux premiers articles de cette série).

On peut facilement ainsi constituer un banc de test par programme en écrivant une procédure qui exécute les différentes étapes du test et qui compare les résultats obtenus avec les résultats attendus. Si ce banc de test peut être automatisé, comme le conseille vivement la PSP, on obtiendra un véritable système de contrôle de la qualité des programmes, analogue au contrôle que l'on trouve dans certaines industries japonaises prônant le « zéro défaut ». En effet, chaque fois que le programme subit une modification, le fait de le passer à la moulinette du banc de test assure qu'il fait toujours ce qu'il faisait avant. Sa qualité ne peut pas se dégrader.

Nous prétendons que ce type

de banc de test permet de cerner la qualité logicielle, car il constitue un outil de mesure des résultats attendus d'un programme. Par ailleurs, plus ce banc est sophistiqué, plus ses résultats sont fins, et plus le niveau de qualité est supérieur.

Cette mesure de la qualité s'insère naturellement dans les différents changements apportés par les stations de travail. Espérons que ces changements se feront sans heurt, comme cela s'est parfois produit avec les mutations successives du matériel informatique et l'introduction de nouvelles technologies.

Le choc logiciel

Nous assistons déjà à une standardisation des logiciels de base sur les micros et mini-ordinateurs : MS-DOS est quasiment partout sur les micros, Unix a beaucoup de succès et est annoncé sur pas mal de minis. Nous attendons encore l'effet du nouveau OS/2 d'IBM. Quoi de plus naturel que le logiciel suive ? Les normes de programmation, tant souhaitées à l'heure actuelle, seront peut-être obligatoires demain. Alors, pourquoi ne pas les définir et les essayer dès maintenant ?

On sait que les performances d'un système informatique proviennent essentiellement du logiciel. Nous ne comprenons pas alors que le matériel devance autant le logiciel en ce qui concerne la normalisation. Est-ce parce qu'il n'existe pas de leader, dans ce domaine, semblable à IBM ? Ou bien le contexte économique actuel ne l'encourage pas ? Si c'est le cas, un changement de ce contexte est peut-être pour bientôt, et nous espérons qu'au moment du choc les quelques milliards d'instructions n'auront pas en face d'eux des personnes désarmées.

J. Maalouf

(*) Moyenne de Temps de Bon fonctionnement (en anglais Mean Time Between Failures) : temps moyen passé entre deux pannes.

(**) Intelligence Artificielle et Démarche PSP. *Micro-Systèmes*, octobre 1987.

(***) La Programmation Sans Panne : Principes Fondamentaux. *Micro-Systèmes*, juillet-août 87.



POWER LAB D'ONDYNE.

**A PARTIR
DE 2995F**

LES PETITS DERNIERS PREMIERS PARTOUT.

Les Power Lab, une nouvelle série d'alimentation de secours mise au point par France Onduleurs Ondyne, ne sont petits que par leurs prix : à partir de 2 995 F HT. Partout ailleurs ils sont les premiers. Premiers car ils protègent efficacement la micro-informatique et plus spécialement les PC, XT, AT et compatibles contre les aléas du secteur (parasites et pannes de courant jusqu'à 30 mn). Premiers car ils proposent un large choix de 200, 400, 800 et 1200 VA. Premiers car ils satisfont parfaitement les utilisateurs de la micro (plusieurs milliers).

Et pour les configurations plus élaborées, France Onduleurs Ondyne vous conseille sa gamme d'onduleurs dont la puissance s'échelonne de 300 VA à 30 KVA.



L'ALIMENTATION DE SECOURS DE VOTRE ORDINATEUR.

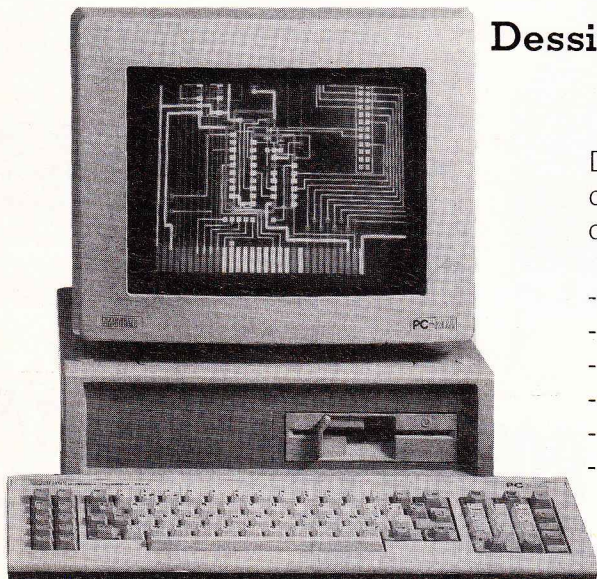
FRANCE ONDULEURS ONDYNE
8, Rue de la Mare
91630 AVRAINVILLE
Tél. 60.82.06.54 Téléx 690 804

SERVICE-LECTEURS N° 295

1 350 F
TTC

DACIM

DAO pour circuits imprimés



Dessinez **VITE** et **BIEN** vos circuits imprimés

DACIM est déjà utilisé par de nombreuses sociétés d'électronique pour gagner du temps et réduire les coûts d'étude des circuits imprimés.

- librairie de composants extensible
- sortie sur imprimante ou table traçante
- déplacement et effacement des composants
- sortie des documents à l'échelle 1 ou 2
- utilisation très facile et rapide
- fonctionne sur compatible PC et AT

Demander documentation à SIDENA
117 rue de la Croix Nivert
75015 PARIS - Tél. : 45.33.86.23

SERVICE-LECTEURS N° 296

Asian Sources Computer

SYSTÈMES INFORMATIQUES

64, avenue de la Marne
92600 ASNIÈRES

Tél. : **47.91.38.03**

Tél. : **47.93.76.05** Télex : 649541

AT COMPATIBLE 286-386

PC/XT COMPATIBLE

IMPRIMANTES LASER

MONITEURS EGA **PRIX PROMO : 3750F** TTC

CARTES D'EXTENSION

TRAITEMENT DE TEXTE

LOGICIELS - FORMATION

SYSTEMES PAO - SCANNER

CARTES TELETEXTE / TELEX

**VENTES - LOCATION
CRÉDIT-BAIL**

BON DE COMMANDE

DOCUMENTATION

MS 12/87

Nom : Fonction : Tél. :

Société : Adresse :

Télex : Code Postal : Matériel :

IBM, PC/XT/AT sont des marques déposées IBM corp.

SERVICE-LECTEURS N° 297

C++: LE NOUVEAU LANGAGE OBJET

Certains privilégiés en ont déjà une version préliminaire sur Macintosh après lui avoir fait traverser l'Atlantique (la version définitive sera annoncée, semble-t-il, début 1988). On connaissait déjà Object Pascal fonctionnant sous Macapp avec accès à la ROM du Macintosh. L'idée serait d'offrir ce type d'environnement pour C++ et de couvrir ainsi une grande partie des développeurs. Il est probable que ce langage aura une importance décisive pour la consécration d'Apple dans le monde industriel : si on prend en compte son ouverture au monde d'Unix avec le Mac II, véritable petite station de travail, l'intégration de C++ semble naturelle et confirme l'aptitude professionnelle de la lignée Macintosh.

Si son auteur est Bjarne Stroustrup (AT & T Bell Labs), son nom est une trouvaille d'un dénommé Rick Mascitti, qui date de l'été 1983. Pourtant, dès 1980, les versions précédentes utilisent la terminologie de « C avec des classes ». Le terme « C++ » marque l'extension de C symbolisée par son opérateur d'incrément « ++ ». L'introduction du livre de Bjarne Stroustrup, véritable bible sur C++, commente avec humour cette appellation : « Le nom légèrement plus court de C+ est une erreur de syntaxe ; il a par ailleurs été utilisé pour un autre langage. Les connaisseurs de la sémantique de C trouvent C++ moins bon que ++C. Le langage ne se nomme pas D du fait que c'est une extension de C dont le but n'est pas de résoudre des problèmes existants par de nouvelles caractéristiques. Pour une autre interprétation du nom C++, voir l'appendice du livre d'Orwell. » [1] (« 1984 »). A Porigine, il s'agissait pour son auteur de

Adeptes de C, fans des objets et les autres, à vos claviers : C++ arrive. Extension du langage C au concept d'objet, C++ est d'ores et déjà distribué en France par la société Axis depuis août 1987 et il est probable de le voir apparaître dans le catalogue de différents constructeurs.

programmer des simulations dirigées par les événements (event-driven), pour lesquelles Simula 67 aurait été le langage idéal s'il ne posait pas des problèmes d'efficacité. Le but était avant tout de rendre plus aisée l'écriture de bons programmes. « Il n'y a jamais eu, précise-t-on dans l'introduction du livre, de « projet C++ », de document de spécification ni de « comité pour la conception de C++ » : l'idée en est venue de manière naturelle pour répondre à un besoin, l'évolution s'est faite ensuite selon les besoins exprimés à travers des discussions entre l'auteur et ses amis ou collègues ». [1].

Pourquoi C++ ?

L'un des buts initiaux de C était de remplacer l'assembleur pour la plupart des tâches systèmes. Il fallait conserver cet acquis avec C++. La grande différence entre C et C++ est l'importance des types et des structures. C est expressif et permissif, C++ est encore plus expressif mais contrôle le type des objets. Connaissant le type, le compilateur peut gérer correctement les expressions pour lesquelles le programmeur aurait eu sans cela la charge d'écrire péniblement toutes les opérations. La connaissance du type permet également la détection d'erreurs qui, sinon, subsisteraient jusqu'aux tests.

Si l'accent est mis sur la structuration avec C++, c'est qu'avec la taille croissante des programmes C sont apparus des problèmes causés par une mauvaise structure initiale. Il fallait offrir le moyen de structurer de gros programmes de façon rationnelle pour qu'il ne soit plus inabordable pour un programmeur de faire face à 25 000 lignes de codes. Il existe des programmes encore plus

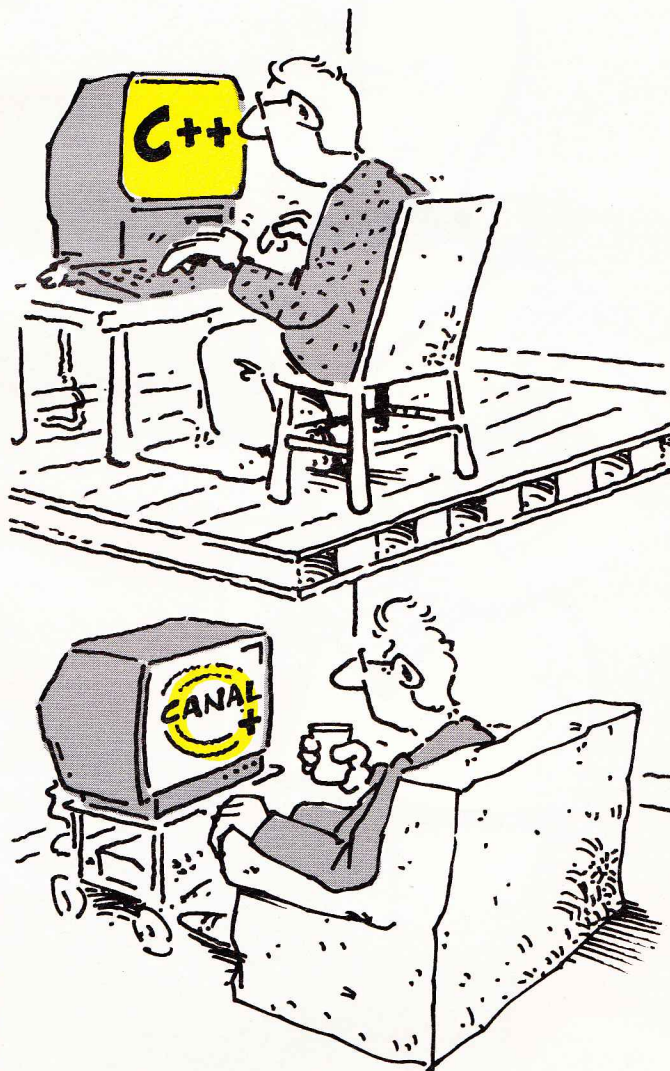


Illustration Colibr-Thibert

INTELLIGENCE ARTIFICIELLE

importants, mais ceux qui tournent se décomposent pour la plupart en différentes parties assez indépendantes les unes des autres et de taille bien inférieure à 25 000 lignes. Cependant, la difficulté d'écrire et de maintenir un programme ne dépend pas forcément du nombre de lignes mais plutôt de la complexité de l'application.

Bjarne Stroustrup s'est attaché dans son livre à décrire des techniques pour concevoir des outils et des types à caractère général, des bibliothèques, etc. On pourrait penser qu'un programme manipulant des types plus élaborés génère plus de texte source. Avec C++, ce n'est pas le cas : le programme est légèrement plus court que son équivalent en C.

C est souvent critiqué sur sa

La relation entre les types de base peut être écrite de la manière suivante:

$$1 \quad \text{taille-de(char)} \leq \text{taille-de(short)} \leq \text{taille-de(int)} \leq \text{taille-de(long)} \\ \text{taille-de float} \leq \text{taille-de(double)}$$

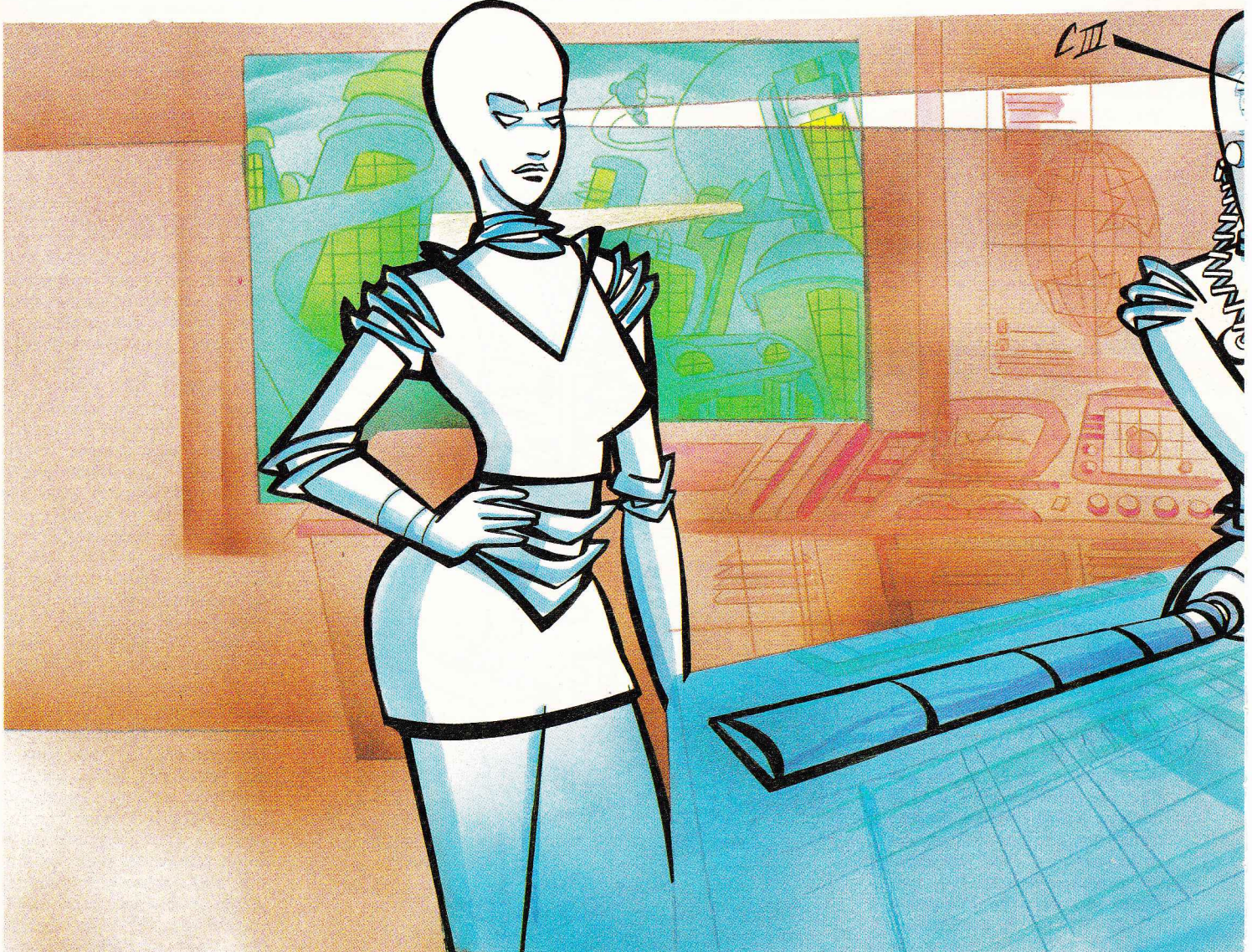
Fig. 1. - Les types de base.

permissivité et notamment sur l'absence relative de vérification sur les types (type checking). Comme l'extension générée par C++ induit justement un enrichissement des types, l'apport est évident. C'est peut-être un inconvénient pour les « rois de la bidouille », qui tiraient parti de cette situation. L'existence d'un mécanisme d'« operator overloading » (« surcharge ou redéfinition d'opérateur »), qui

permet de définir des fonctions sous un même nom mais avec des paramètres différents, apporte une compensation à cet « inconvénient ». Il est clair que l'opération « afficher » ne s'exécute pas de la même façon selon que l'on veut afficher une fenêtre ou un cercle, qui sont deux objets de type différent. On parle aussi de routines « polymorphiques » (ce terme signifie « existant sous plusieurs formes ») et s'applique à des

constructions d'un langage appartenant à des types différents).

C++ étend le concept de structure de C (*struct*) (fig. 3), accroît l'efficacité et facilite la maintenance en favorisant la transparence. Les temps de développement se trouvent raccourcis par la simplicité des mécanismes de programmation qui reposent sur l'indépendance réelle des types se référant chacun d'eux à un objet




```

class set {
    struct setmem {
        int mem;
        setmem* next;
        setmem(int mem, setmem* n) {mem=m; next=n}
    };
public:
    set () { first=0 }
    insert (int m) {first = new setmem (m,first);}
    // ....
};
    
```

Fig. 2. - Il est possible d'emboîter des classes.

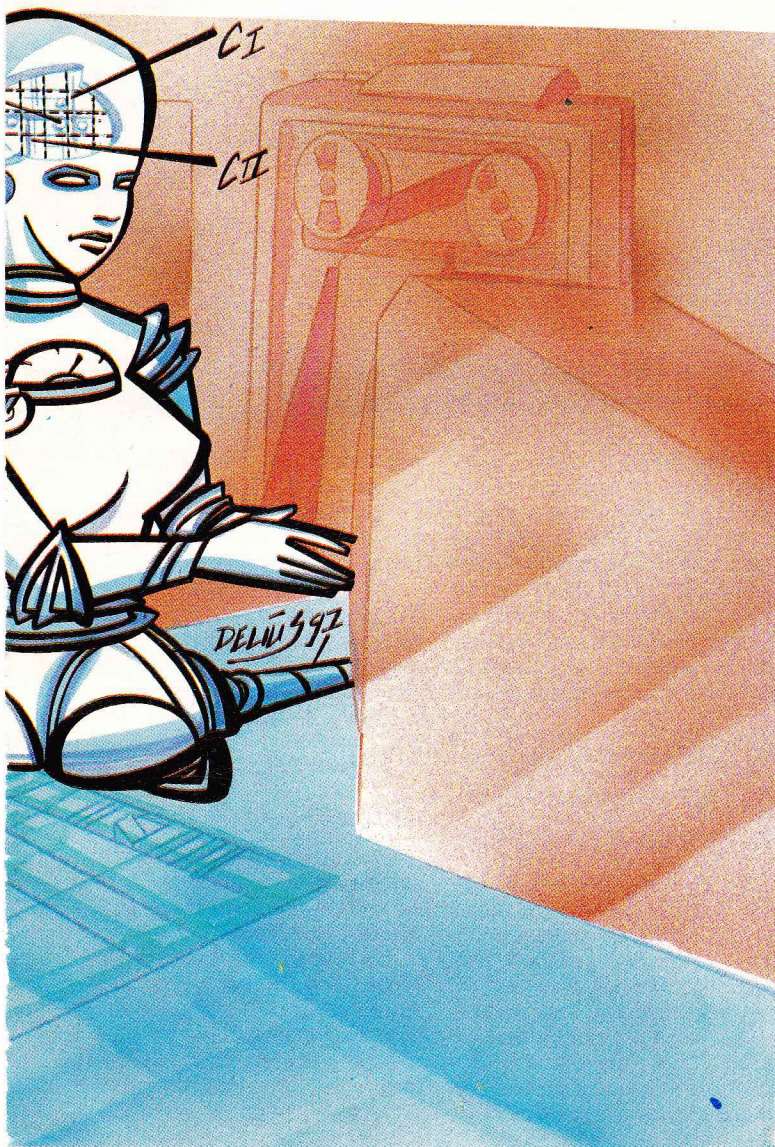
réel et comportant leurs propres opérations. On trouve là un aboutissement de la programmation par modules, qui

est à la base de la programmation structurée. L'évolution d'un système à l'autre s'explique de manière très simple

quand on conçoit un module sous la forme d'un objet. Ce dernier devient quelque chose dont on peut parler, sur lequel on peut opérer, il est « réifié ». Sa maintenance, sa modification, son développement s'en trouvent extrêmement facilités et accélérés. Il est possible de

changer, de façon transparente par rapport au reste du système, ses « réactions » (ou opérations décrites par les fonctions membres). C'est ainsi que des applications en C deviennent réalisables : « J'avais essayé d'écrire un « window manager » en C et j'ai vite abandonné devant la difficulté, alors qu'avec C++ j'ai pu le concevoir assez rapidement » témoigne Jean-Christophe Collet de chez Axis Digital. On croirait presque entendre une publicité pour une nouvelle lessive qui lave « plus blanc que blanc ». Et pourtant, c'est ce type de langage que tiennent habituellement les nouveaux adeptes des objets : l'essayer (ou le programmer), c'est l'adopter !

L'interfaçage avec des bibliothèques de programmes C existants ne pose aucun problème : C++ est lui-même écrit en C et se présente sous la forme d'un préprocesseur qui transforme le source en code C à destination



Par définition, *struct* définit simplement une classe dont les membres sont publics :

```

struct s { ...
est équivalent à
class s { public: ...
    
```

Inversement, il est possible avec C d'implémenter un concept proche de celui de classe en incluant des fonctions dans un *struct* (elles deviennent des fonctions membres). Mais cela n'implique pas que ce soit le seul moyen d'accéder aux objets de la classe, le principe d'encapsulation n'est pas respecté :

```

Struct date {
    int jour, mois, année;

    // déclaration des fonctions
    void modifier (int, int, int);
    void lire (int*, int*, int*);
    void suivant ();
    void afficher ();
};
    
```

struct définit une classe dont tous les membres sont publics et son utilisation se justifie quand l'encapsulation des données n'est pas nécessaire. Elle peut définir une union c'est à dire une structure où tous les membres ont la même adresse ce qui peut être utile pour économiser de l'espace quand on sait qu'un seul membre de la structure sera utilisé à un instant donné.

Le label "public" permet de séparer la partie cachée de la classe de celle qui assure son interface avec l'extérieur :

```

class date {
    int jour, mois, année;

    // déclaration des fonctions
public:
    void modifier (int, int, int);
    void lire (int*, int*, int*);
    void suivant ();
    void afficher ();
};
    
```

Fig. 3. - La notion de classe étend *struct*.

Encadré 1

Le point de vue de B. Stroustrup sur les concepts de type abstrait de donnée (T.A.D.) et de programmation par objet (P.P.O.)

En juin dernier, Bjarne Stroustrup, l'auteur même de C++, a consacré son intervention, lors de la conférence ECOOP'87, à essayer de clarifier les notions de « type abstrait » et celle de « programmation objet ». « Ces termes sont devenus très communs, dit-il. Malheureusement peu de gens sont d'accord sur leur sens » [2]. On met souvent en parallèle les notions de « type abstrait » et d'« objet » (une classe définit un type pour toutes ses instances), à tel point que beaucoup confondent volontairement type et classe. Des langages comme Ada, Clu, C++ donnent la possibilité de définir des types qui se manipulent plus ou moins de la même façon que des types de base. Cela donne lieu à ce que l'on appelle souvent les « types abstraits de données ». B. Stroustrup exhibe la notion de type abstrait sous la forme du résultat d'une évolution dans les techniques de programmation, se refusant à tomber dans l'ornière d'une nouvelle définition. (Voir cependant l'encadré 2 qui donne des précisions supplémentaires sur la notion de type, et plus précisément de type abstrait.)

C++ est, selon lui, l'un des rares langages qui intègre à la fois ces deux concepts (P.P.O. et T.A.D.), outre les possibilités de programmation classiques. Comprendre le concept de type abstrait, c'est d'abord savoir ce qu'est la programmation procédurale et l'encapsulation de données (data hiding). Programmer avec des modules conduit à la centralisation de toutes les données d'un certain type sous le contrôle d'un gestionnaire de type. Le concept de module avec l'encapsulation de données autorise la programmation par type abstrait mais ne la « supporte » pas : « Un langage ne supporte pas une technique si l'écriture d'un programme faisant appel à cette technique nécessite un effort ou une habileté exceptionnelle. On peut, par exemple, écrire des programmes structurés en Fortran, des programmes typés en C et utiliser des types abstraits en Modula-2, mais c'est d'une grande difficulté car ces langages ne supportent pas ces techniques. » Bjarne Stroustrup établit en effet une distinction fondamentale entre des concepts, des principes ou techniques énoncés abstraitement et la manière dont ils sont réalisés, implémentés ou intégrés à un langage. En d'autres termes, il distingue le concept et la façon dont il est « supporté » ou, plus généralement, l'abstraction et la réalité.

Le concept de classe est le moyen de définir de nouveaux types qui puissent être utilisés aussi simplement que les types de base. Le concept d'héritage associé à celui de type abstrait (ou classe) assure la transition vers la programmation objet en offrant un mécanisme de généralisation (en montant dans la hiérarchie des classes) ou d'affinage (en descendant) : il faut pouvoir faire la différence entre une forme générale, qui peut être dessinée, et la forme « cercle », qui a un diamètre et une fonction dessin qui lui sont spécifiques. « La programmation par objet est une programmation qui utilise l'héritage. Les types abstraits induisent une programmation qui utilise des types définis par l'utilisateur » [2].

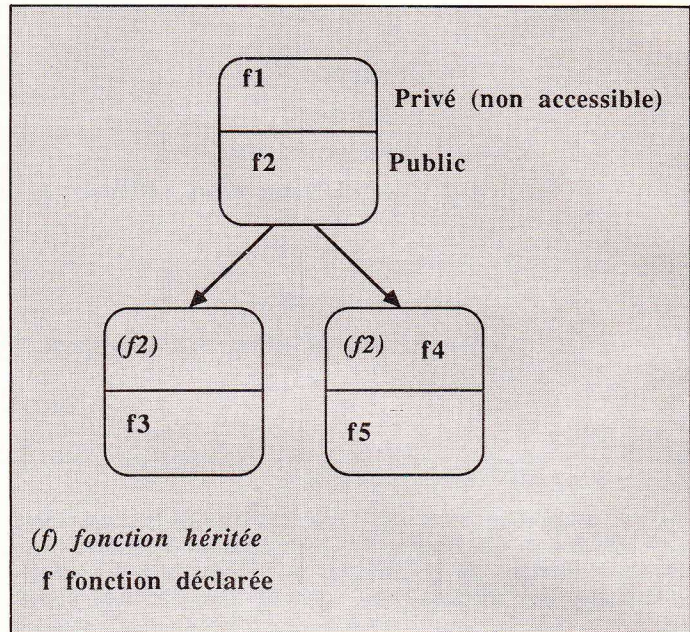


Fig. 4. — La dérivation de classe.

```
class entreprise : public employé {
    .....
};

class entreprise : employé {
    .....
};
```

Dans le premier cas, un membre public de la classe de base est également public dans la classe dérivée alors que dans le second il devient privé.

Fig. 5. — Déclaration d'une classe dérivée et visibilité.

du compilateur C. Il y a cependant deux limitations : d'une part, C++ a été conçu pour des machines 32 bits et s'adapte donc plus difficilement à des architectures 16/32 bits ; d'autre part, il faut tenir compte des bibliothèques liées à une machine donnée dont les routines sont souvent écrites en assembleur.

Structure d'un programme C++ et types de bases

Un programme C++ se décompose de manière classique en une série de fichiers sources qui contiennent chacun une partie déclarative pour les types, les fonctions, les variables et les constantes. De même qu'en C, un nom doit être déclaré en *external* s'il réfère la même chose mais dans deux fichiers différents. On retrouve

également les fichiers *header* qui servent à séparer de telles déclarations des fichiers sources où elles sont référencées par le biais d'un *include*.

Par exemple, la racine carrée (SORT) peut être dans un fichier header MATH.H regroupant les fonctions mathématiques. Si un programme veut l'utiliser, il faut faire un *include* :

```
* include <MATH.H>
...
x = sqrt(4);
```

L'objectif de C++ est d'étendre la puissance du typage tout en offrant et conservant certains types de base :

- les types fondamentaux (*char, short, int, long* pour des entiers et *float, double* pour des nombres flottants) ;
- les types dérivés : ces opérateurs créent de nouveaux types à partir des types de base (fig. 1).

Encadré 2

QU'EST-CE QU'UN TYPE ABSTRAIT DE DONNÉE ?

« La révolution des types abstraits de donnée a commencé au début des années 1970. Ce concept est l'un des plus importants dans ce domaine, et il en est issu peu à peu une définition que la plupart des gens acceptent maintenant :

Un type de donnée est un ensemble de valeurs sur lesquelles est défini un ensemble d'opérations. »

« L'idée majeure qui sous-tend les types abstraits de donnée est de séparer l'utilisation d'un type de sa représentation et de son implémentation : l'utilisation ne devrait en effet dépendre que de l'ensemble d'opérations et de valeur définissant un type. » [3].

Le concept de donnée abstraite étend celui de procédure ou de module qui l'a précédée. L'utilisation d'un type abstrait se définit par une syntaxe et une sémantique, alors que son implémentation est décrite par une représentation et des algorithmes (fig. 8). Les définitions les plus précises du concept de type abstrait ont en commun la notion de « signature » « qui décrit la syntaxe de ce type (nom des opérations, type de leurs arguments), mais elle ne définit pas les propriétés des

opérations du type. C'est par ce dernier aspect que diffèrent les méthodes de définition des types de données » [4]. Ces propriétés sont souvent décrites par des axiomes (formules logiques), comme ont choisi de le faire les auteurs de ce livre (fig. 9). Il s'agit de donner une signification (sémantique) aux composants de la signature : sortes et opérations. « La définition d'un type abstrait de données est alors composée d'une signature et d'un ensemble d'axiomes. Les axiomes sont accompagnés de la définition d'un certain nombre de variables. Ce type de définition s'appelle une définition algébrique ou axiomatique d'un type abstrait. Pour abrégé, on parle souvent de types abstraits algébriques » [4].

En pratique : type abstrait et programmation

Quand on décrit un algorithme, on ne tient pas compte de la représentation interne des données, qui sont alors considérées de manière abstraite uniquement. Quand on manipule des réels sur un ordinateur, il n'est pas nécessaire de connaître leur représentation interne mais seulement la manière de les utiliser

et les opérations que l'on peut faire dessus (+, -...). « Tout informaticien utilise donc le concept de type abstrait, parfois comme M. Jourdain qui faisait de la prose sans le savoir. » [4].

On exprime habituellement un type de donnée directement dans le langage de programmation utilisé en l'associant à des procédures ou des fonctions correspondant aux opérations de ce type. « A partir de là, si on pratique une « bonne » programmation, qui consiste à ne plus manipuler les objets du type que par les opérations du type, on programme en fait avec des types abstraits » (procédure ascendante). Mais si on est familiarisé avec le concept de type abstrait, on peut concevoir l'algorithme directement à partir de la définition des types (procédure descendante). En d'autres termes, on raisonne sur des abstractions sans s'embarrasser des contingences de la programmation : celle-ci consiste ensuite à concrétiser ces types abstraits en les exprimant dans le langage. « La conception est plus simple, puisqu'on n'a pas à prendre en compte des détails de programmation » [4].

conçu de telle façon qu'il est très lié aux objets, instructions, types, opérateurs manipulés par la machine : les nombres, les caractères et les adresses. Excepté pour les opérateurs gérant le « free store » (new et delete), les expressions et instructions C++ ne nécessitent pas de support run time ni de sous-routines (voir encadré 3 pour une définition du « free store »). Chez Axis Digital, on résume en quelques points, sur lesquels nous reviendrons, les possibilités nouvelles qu'offre C++ par rapport à C.

– Le concept de classe permet la restriction d'accès à certaines données, une initialisation garantie, des conversions de type non prédéfinis, l'allocation de type dynamique grâce aux fonctions virtuelles. Le nom d'une classe est un nom de type.

– Les types des arguments d'une fonction peuvent être spécifiés (les conversions éventuelles sont effectuées).

– Les noms de fonctions peuvent être « surchargés » (réutilisés plusieurs fois : technique de l'« overloading »).

– Les opérateurs peuvent être redéfinis.

– Les fonctions peuvent être « inline » (déclaration et définition regroupées).

– Une allocation dynamique de la mémoire est possible grâce aux opérateurs « new » et « delete ».

Il s'ensuit quelques contraintes : les nouveaux mots clés (class, friend...) ne peuvent être utilisés comme identificateurs, de même que les noms de classe ne peuvent être utilisés comme noms de variables. « f() ; » est la déclaration d'une fonction sans arguments.

L'encapsulation des données (Data hiding)

C autorisait d'ores et déjà avec *struct* l'implémentation d'un concept avec des opérations associées sous la forme de fonctions (fig. 4). Il fallait ajouter à cela l'interdiction de l'accès aux objets d'un certain type autrement que par les fonctions qui le manipulent. Autrement dit, il fallait introduire ce que

Le plus de C++ : le concept de classe

L'apport essentiel de C++ consiste à pouvoir créer de nouveaux types, qui ne diffèrent des types de base que par la façon dont on les crée. Le fondement de cette nouvelle fonctionnalité est le concept de classe introduit par Simula au début des années 1970. Une classe permet de définir un type pour les objets qui lui sont rattachés par un lien d'instanciation (on parle alors d'instan-

ces). Cette définition comprend, outre des données statiques, les opérations qu'il est possible d'appliquer aux instances. Ces opérations sont désignées sous le terme de *fonctions membres* dans C++. L'implémentation d'une classe peut être modifiée sans que les utilisateurs soient concernés. Il suffit que la partie publique de la classe et la déclaration des fonctions membres (donc l'interface avec l'extérieur) demeurent inchangées (fig. 2). Il est possible d'« emboîter » des classes, mais cette pratique est déconseillée dès que l'on a affaire à des classes complexes, qui

doivent alors être déclarées séparément (fig. 3). Pour chaque objet, le « moule » ou « squelette » qui détermine la structure des données de toute instance prend des valeurs qui lui sont propres. Il est cependant parfois élégant d'implémenter certains types en partageant des données entre tous les objets. On déclare alors la donnée membre au sein de la classe en *static*. Elle peut être accédée de l'extérieur si sa déclaration fait partie de la zone publique. L'emploi de *static* réduit considérablement le recours à des variables globales.

Le sous-ensemble C est


```

Si on reprend l'exemple de la fig. 4, on peut déclarer des fonctions d'initialisation
différentes:

date (int, int, int);    // jour mois année
date (char*);           // chaîne de caractères

date (int);             // jour avec le mois et l'année courante
date ();                // la date courante (option par défaut)
    
```

Fig. 6. — Initialisations multiples d'une classe.

```

class ostream (
// ...
public:
    ostream& operator<< (char*);
    ostream& operator<< (int i);
    ostream& operator<< (return *this<<long(i));
    ostream& operator<< (long);
    ostream& operator<< (double);
};
    
```

Fig. 7. — Déclaration de l'opérateur « << » dans la classe Ostream pour chaque type de base.

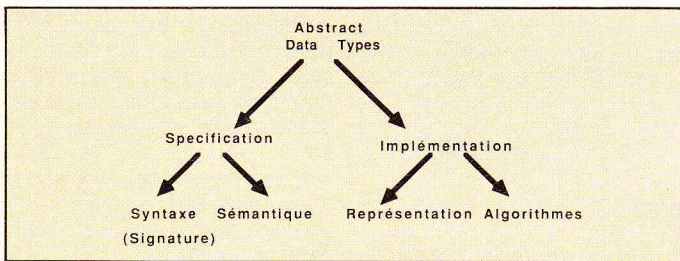


Fig. 8. — Les différentes composantes d'un type abstrait de donnée.

On appelle l'encapsulation des données pour pouvoir véritablement parler de Type Abstrait de Donnée (T.A.D.) et de classe (voir encadrés 1 et 2 sur la notion de T.A.D.). Il y a différents avantages à restreindre l'accès à une structure de données. Tout d'abord, cela permet de localiser l'erreur si l'une des données de la structure prend une valeur fautive. Par ailleurs, apprendre le maniement d'un type se réduit à connaître ses fonctions membres. De façon plus générale, tout changement dans le comportement d'un type peut et doit être effectué en modifiant ses membres. Du point de vue syntaxique, l'encapsulation est définie en séparant la « partie » « publique » de la partie « privée » d'une classe par le label *public* (fig. 3). La partie privée ne pouvant être utilisée que par les fonctions membres, la partie publique constituant l'interface avec les objets de la classe. « La protection des données privées repose sur la limitation imposée par les noms des fonctions membres. De ce fait, on peut la contourner par des manipulations d'adresses et des conversions de types mais c'est bien sûr une tricherie... » [1].

La dérivation de classes

L'une des grandes caractéristiques d'un langage orienté objet est de dériver des classes à partir d'autres en permettant l'héritage, de propriétés entre ces classes. Avec cette notion d'héritage, de nouvelles classes sont définies à partir d'autres existantes en réutilisant ainsi une programmation et une compilation déjà effectuées. Une *classe dérivée* hérite d'une *classe de base* et peut, elle-même, être la classe de base d'autres classes. On voit ainsi se constituer une hiérarchie de classes. Une classe dérivée ne déclare que les données et les méthodes (opérations) qui lui sont propres. Autrement dit, les fonctions membres de la classe de base appartiennent également à la classe dérivée. Si elles sont définies en « public », elles deviennent « privées » pour la classe dérivée et donc accessibles pour les niveaux inférieurs à travers les membres publics de la classe dérivée (fig. 4). (Notons que la classe dérivée n'a pas accès à la partie privée de la classe de base, car ce serait la négation même du

concept d'encapsulation, même si cela reste possible avec les fonctions friend décrites plus loin.) Par ailleurs, avec le mécanisme de dérivation, il devient simple de définir une interface commune à plusieurs classes de façon à ce que les objets de ces classes puissent être manipulés de manière identique par d'autres parties du programme. Cela implique notamment que le type d'un objet puisse ne pas être connu à la compilation pour être utilisé de manière adaptée aux contextes d'utilisation à l'exécution (fig. 5). La notion de *fonction virtuelle* est introduite pour résoudre de telles dépendances à l'exécution. Le compilateur sait alors qu'une telle fonction peut être redéfinie dans toutes les classes dérivées (elle garde cependant le type qui a été déclaré dans la classe de base).

Les fonctions

Autres réminiscences de C, les fonctions sont des sous-parties de programme qui peuvent être appelées d'un point ou un autre soit de l'intérieur du programme où elles sont définies, soit de l'extérieur par le biais des déclarations en « externe ». Le compilateur peut différencier deux fonctions de même nom si elles ont des arguments différents (ex. : élever à la puissance).

Les fonctions *constructor* portent le même nom que la classe à laquelle elles appartiennent. Elles « construisent » les valeurs d'un type donné, leur rôle est de créer et d'initialiser les membres privés de la classe. Quand une classe a un constructeur, tous ses objets seront initialisés. Il est possible et souvent agréable de pouvoir initialiser l'objet d'une classe de différentes façons (fig. 6). Tant que les arguments des fonctions d'initialisation diffèrent suffisamment, le compilateur peut sélectionner le bon.

Les fonctions *destructor* assurent la fonction inverse. Il est en effet parfois utile de détruire les objets d'un type donné. En particulier, un certain nombre de classes utilisent de la mémoire faisant partie du *free store* que l'on alloue par un constructeur et désalloue par un destructeur.

Les fonctions *inline* ont été conçues afin de résoudre le problème posé par la manipulation des appels de petites fonctions qui peuvent affecter lourdement l'efficacité des programmes. C'est un aspect de la programmation par classes qui s'explique par le nombre et la petite taille des opérations générées pour chaque type. C'est aussi le résultat des nouvelles techniques de programmation à base de modules courts et indépendants : « Ce qui était une convention devient un standard reconnu par le compilateur » [1]. Les fonctions courtes peuvent être déclarées en *inline* pour optimiser le temps d'exécution : la génération de code est alors faite dès la déclaration. Toute fonction membre dont la définition fait partie de la déclaration de la classe est « *inline* » implicitement.

Les fonctions *friend* (« amies ») permettent d'accéder à la partie privée d'une classe sans être pour autant des fonctions membres. Cette facilité se justifie quand, par exemple, une fonction fait appel à deux classes différentes : si on veut multiplier un objet vecteur et un objet matrice, la fonction *multiplier* sera déclarée en *friend* dans chacune des deux classes. Ce moyen augmente l'efficacité de la fonction en évitant de passer par des fonctions membres qui pourraient alourdir l'accès inutilement (en l'oc-

LA NOTION DE « FREE STORE » DANS C++

Si rien de particulier n'est spécifié, la « vie » d'un objet se limite au temps pendant lequel son bloc est accédé, ou à la durée d'exécution du programme si son nom est déclaré en global. Dans un cas, on parle d'objet dynamique, et dans l'autre d'objet statique. Il est parfois utile de le maintenir jusqu'à ce que l'on n'en ait plus besoin. On utilise pour cela l'opérateur *new* (et inversement *delete* pour le détruire) : on dit alors que les objets alloués par *new* sont dans le *Free Store* (appelé habituellement *heap*).

La signature d'un type abstrait est la donnée d'une part des noms de ses ensembles de valeurs (appelés sortes), d'autre part des noms des opérations qu'on leur applique, en précisant à quels ensembles appartiennent leurs arguments:

| | |
|-------------|--------------------|
| sorte R,S,T | |
| opérations | |
| o | : R x T ---> S |
| p | : R x T x S ---> R |
| q | : R ---> T |
| v | : R ---> T |

Un type bien connu, celui des booléens:

| | |
|---------------|---------------------------------|
| sorte Booléen | |
| opérations | |
| vrai | : --> Booléen |
| faux | : --> Booléen |
| ¬ | : Booléen --> Booléen |
| ∧ | : Booléen x Booléen --> Booléen |
| ∨ | : Booléen x Booléen --> Booléen |

Il est clair que ces descriptions n'ont pas de sens en soi, il faut leur adjoindre une sémantique qui donne une signification à R,S,T, o,p,q,v.

Exemple d'axiome :

$\text{borneinf}(s) \leq i \leq \text{bornesup}(s) \Rightarrow \text{ième}(\text{changer-ième } s,i,e) = e$
où s, i, e sont des variables de sorte Séquence, Entier, Élément.

Cet axiome exprime que si i est compris entre les bornes d'une séquence s, alors quand on construit une nouvelle séquence en donnant au ième élément la valeur e et qu'on veut ensuite accéder au ième élément de cette nouvelle séquence, on obtient e.

Fig. 9. - Types abstraits de données : signature et axiome.

currence en contrôlant par exemple les indices). Une telle fonction ne peut elle-même être déclarée en membre dans deux fonctions à la fois, ce qui explique le recours à la notion de friend.

Self-référence

La « self-référence » est un mécanisme de base en programmation-objet : elle permet de désigner le receveur du der-

nier message envoyé (avec C++, on parle plutôt d'appels de fonctions). Celui-ci peut alors se « parler à lui-même » en émettant un message qu'il se destine par le biais de « self ». En C++, dans une fonction membre, on peut référer directement à l'objet pour lequel la fonction membre est appelée (fig. 10). L'équivalent de « self » est « this », qui permet de pointer sur cet objet. This est utilisé principalement pour

écrire des fonctions membres qui manipulent directement des pointeurs.

Les entrées-sorties

L'implémentation pour C++ de la librairie de gestion des E/S diffère totalement de la librairie classique du C qui ne traite que des types prédéfinis classiques. Avec la notion de

stream, C++ offre des possibilités nouvelles d'input/output (I/O) qui prennent également en compte les types définis par l'utilisateur. Dans un tel contexte, il est difficile de proposer un système d'I/O qui contente chaque utilisateur. Aussi a-t-on préféré fournir un outil simple et souple, qui permette des alternatives adaptées aux applications. La bibliothèque <stream.h> contient les outils d'interface utilisateur et se base exclusivement sur des processus de conversions d'objets typés en séquences de caractères. « Il existe d'autres modèles d'I/O mais celui-ci est fondamental dans un système Unix [1]. La tâche du programmeur est alors de définir une correspondance entre un objet typé et une chaîne de caractères non typée. On ne fait plus appel à des fonctions au sens de C mais à des opérateurs qui pourront être redéfinis selon le type (technique de l'overloading).

En ce qui concerne les « sorties », C++ distingue les types de base et les types utilisateur. La définition d'une opération en sortie utilise la classe *ostream* (pour Output Stream) qui se trouve dans le fichier header standard auquel l'utilisateur n'a pas accès. Il n'est en effet pas nécessaire de la changer ni d'avoir accès à ses données privées, même lorsqu'il s'agit d'un type utilisateur. Cela permet par ailleurs de modifier l'implémentation de la classe sans affecter son utilisation. Dans le cas des types de base, l'opération « << » est définie pour chacun des types dans la partie publique de la classe *ostream* (fig. 7). Dans le cas des types utilisateur, l'opérateur « << » peut être redéfini pour chaque nouveau type à l'aide de « *operator* ».

Pour les entrées, on trouve de manière symétrique la classe *istream* à laquelle est associé l'opérateur « >> ».

Les entrées-sorties sont basées de façon globale sur les streams. Trois solutions sont possibles :

- soit continuer à utiliser les I/O de C ;
- soit développer ses propres I/O à partir de la classe *stream* (cette alternative montre les possibilités d'extensions indui-

Le "paradigme" de programmation d'origine (et celui qui est encore le plus couramment utilisé) est le suivant:

Définissez vos procédures;
utilisez le meilleur algorithme possible.

Avec l'encapsulation le paradigme devient:

Définissez vos modules;
Partitionnez le programme de sorte que les données
soient cachées au sein des modules.

L'introduction des types abstraits de données le transforme en:

Définissez vos types;
Rattachez un ensemble d'opérations à chaque type.

L'héritage permet d'aboutir à la programmation-objet:

Définissez vos classes;
Rattachez un ensemble d'opérations à chaque classes;
Explicitez ce qui est partageable en ayant recours à l'héritage;

Fig. 10.
Evolution des
mécanismes de
programmation
(d'après
B. Stroustrup
[1]).

tes par les concepts de classes et d'héritage entre autres);
- soit utiliser les I/O de C++, qui sont en fait la plupart du temps largement suffisantes et aussi efficaces qu'en C.

Comment ne pas programmer en C avec C++ ?

Le grand inconvénient de C++ est sans doute qu'il ne prévoit pas de moyen d'empêcher les programmeurs chevronnés de continuer à programmer en C avec C++! Il appartient à une nouvelle génération de langages dont font partie Lisp et, d'une manière

générale tous les langages orientés objet. Rappelons les premières lignes d'introduction du manuel de LE-Lisp sur Macintosh: « LE-Lisp est un langage de manipulation symbolique. Le comprendre, c'est s'ouvrir au monde de l'Intelligence Artificielle, découvrir les chemins qui ont conduit à l'élaboration des systèmes experts et des analyseurs de langage naturel. Au début, LE-Lisp est un langage déroutant. Sa syntaxe fait un usage démesuré de parenthèses, qui rebute le débutant comme le programmeur chevronné spécialiste de Fortran, Basic ou Pascal » [5]. On peut d'une certaine manière effectuer un parallèle avec la programmation objet. Programmer en termes d'objets,

de classes, d'envoi de messages, d'opérations locales à un objet, c'est s'imposer une nouvelle vision de la programmation qui peut amener l'adepte des langages classiques à une remise en cause proche du traumatisme! Les automatismes et les mécanismes de transformation du réel en termes informatiques sont totalement bouleversés. Les informaticiens ont pris l'habitude de faire compliqué, d'où la difficulté de faire simple maintenant. Et pourtant rien de plus évident et de plus naturel que le concept d'objet: à un « objet » réel quelconque (fenêtre, icône, aile d'avion...) correspond un « objet » informatique et ses opérations associées sous forme de bouts de

code (réduire, déplacer, afficher, calculer...). On peut espérer qu'avec C++ la programmation objet se « démocratise » et surtout s'industrialise pour sortir du cercle des privilégiés en accédant au monde d'Unix, grâce à la compatibilité avec les programmes en C existants. De cette façon, C++ entérinerait la maturité atteinte par les langages objets, qui pourraient passer des mains des chercheurs à ceux des industriels.

G. Houbart

Pour en savoir plus :

[1] « The C++ programming language », par Bjarne Stroustrup, Addison-Wesley 1986.

[2] Actes de la conférence ECOOP'87, numéro spécial de la revue Bigre (n° 54) (s'adresser à l'AF-CET ou à l'INRIA).

[3] « An introduction to Data Types », par J. Craig Cleaveland, Addison-Wesley 1986.

[4] « Types de données et Algorithmes », par M.-C. Gaudel, M. Soria et C. Froidevaux, édité par l'INRIA, 1986.

[5]

AXIS DIGITAL

Axis-Digital
135, rue d'Aguessau
92100 Boulogne
Tél. : (1) 46.03.37.75

Axis-Digital a pris une licence auprès de AT & T Bell Laboratories (USA) pour commercialiser C++ en France en assurant son portage sur différentes machines sous Unix.

Machines et tarifs : Axis distingue les machines du groupe 1 telles que le PC/AT sous microport système 5/AT pour lequel la licence après portage vaut 7 500 F, et les machines du groupe 2 telles que :

- le DLX de Thomson ou le MicroVax 1 (licence : 12 000 F);
- Le Vax 750 ou 8500 (licence : 20 000 F).

Prestations : Axis assure de façon générale le portage à la demande de C++ sur toute machine (à moins que le constructeur l'ait déjà entrepris). En particulier sur Macintosh, l'annonce sera peut-être faite directement par Apple (avec qui Axis est d'ailleurs en contact).

(4) CONCEPTION D'UN SYSTEME EXPERT

Pour qu'un système expert parvienne rapidement à trouver la solution à un problème donné, il faut qu'il dispose d'une importante base de connaissances : plus les connaissances et métaconnaissances sont nombreuses, mieux les faits décrivant la situation peuvent être pris en compte, et plus rapide et appropriée est la résolution. La constitution de la base de connaissances est donc d'une importance capitale. C'est à cette tâche que nous nous sommes attelés, le domaine d'étude étant celui de la contraception féminine. Les trois phases interdépendantes de ce travail sont le recueil des connaissances, le codage des connaissances et éventuellement la représentation de l'incertitude.

La première étape dans la réalisation de la base de connaissances est le recueil d'informations auprès d'expert de grande expérience, c'est le rôle du cognitif. Le cognitif se situe donc entre l'expert et le programmeur.

L'équivalent de ce terme, déposé par la société « Cognitech », est : ingénieur de la connaissance.

Le rôle du cognitif

Le métier de cognitif consiste à recueillir des connaissances auprès d'experts, et à les coder de manière utilisable dans une base de données, par un moteur d'inférence. Toute la difficulté de ce métier tient à comprendre comment l'expert raisonne et à identifier sa stratégie pour la représenter dans un système expert.

■ **Les qualités du cognitif**

Il possède une grande ouverture d'esprit, et ses connaissances sont suffisamment diverses et étendues, pour lui permettre d'aborder, sans *a priori*, un do-

maine neuf pour lui. Sa curiosité intellectuelle doit être hors du commun. Une humilité certaine est nécessaire : le cognitif est toujours conscient de son ignorance, son rôle n'est pas d'en savoir plus que l'expert, mais de toujours tenter d'enrichir la base de connaissances par un dialogue fructueux. Cela nécessite donc des qualités indéniables en matière de communication. Le cognitif est demandeur d'informations auprès d'experts généralement plus âgés que lui, et de grande expérience ; l'habileté du cognitif est donc telle qu'il puisse convaincre l'expert de livrer des connaissances qui sont le fruit d'années de travail. Il sait donc diriger les entretiens de manière directive, mais non contraignante.

Il ne rationalise pas à tout prix, ne cherche pas à rapprocher le raisonnement de l'expert de schémas connus, mais à interroger celui-ci de manière à compléter les éléments manquants du raisonnement ; il ne cherche en aucun cas à se substituer à l'expert, même s'il croit avoir perçu quelle est sa démarche intellectuelle. Il cherche plutôt à révéler les raisonnements sous-jacents, et parfois inconscients, de ce dernier. Il tente donc, au cours des entre-

tiens, de découvrir les articulations du raisonnement.

Si le cognitif doit pouvoir obtenir les informations clés qui lui manquent, il doit aussi savoir faire face à un grand flot d'informations, parfois inorganisées. Il lui faut très rapidement faire le tri entre les informations pertinentes et les autres.

Si on ajoute qu'il lui faut aussi posséder une formation informatique classique qui lui permettra de choisir un mode de représentation des connaissances adapté aux caractéristiques techniques du système expert, on conclura que le cognitif est un homme rare, d'autant plus qu'il n'existe pas à ce jour de formation.

L'acquisition des informations, le recueil des connaissances se font d'abord au cours d'entretiens. Ceux-ci peuvent être menés de façon variable.

■ **Méthodes d'entretien**

Il n'existe pas de règle en la matière, mais ouvrir une série d'entretiens par des discussions très libres semble être indispensable. Les premiers entretiens permettent de cerner le domaine sur lequel va porter l'échange. Il est important que l'expert, comme le cognitif,

trace les limites du thème sur lequel porteront les discussions. Le cognitif met en place, au cours de ces premiers entretiens, une méthode de recueil des connaissances adaptée à la personnalité de son interlocuteur et choisit un mode de représentation. Les experts que nous avons rencontrés se sont spontanément exprimés sous la forme de règles conditionnelles, nous leur avons donc demandé par la suite d'exprimer systématiquement toutes les connaissances sous cette forme ; c'était le mode de représentation le plus approprié.

Nous avons recueilli, au cours des premiers entretiens, une masse d'informations inorganisées ; les experts n'ont pas essayé, et nous ne leur avons pas demandé, d'exposer de façon construite tous les aspects du sujet.

Dans un deuxième temps, les connaissances recueillies ont été soumises aux experts, sous la forme de règles de production. Ils les ont critiquées et complétées. Ce feedback aurait pu ne pas être nécessaire, dans la mesure où les connaissances étaient souvent livrées sous cette forme, mais ce contrôle est, en fait, très important : il permet à l'expert de revenir sur

la forme et sur le sens, et d'ajouter éventuellement des compléments. Si la méthode de représentation est en elle-même porteuse de contraintes, elle ne doit pas pour autant limiter l'expression de l'expert : celui-ci doit rester libre d'en choisir la formalisation.

Comme la nécessité d'affecter des coefficients de pondération aux règles est apparue très vite, les experts ont spontanément attribué ces coefficients sous la forme de pourcentages.

La phase la plus complexe de cette étude a été de déterminer des règles de niveau supérieur, ou métarègles qui permettent au système de choisir les grandes orientations de sa recherche, devant une situation donnée. Ces connaissances, non élémentaires, sont celles qui sont le moins apparentes dans le discours des experts. Il s'agit alors, pour le cognicien, de déduire, des connaissances détaillées qui lui sont transmises, le raisonnement qui les sous-tend.

Un autre écueil à éviter est le fait d'omettre d'explicitier les connaissances évidentes et non exprimées par l'expert. Le cognicien devra percevoir ces manques et les faire formuler par l'expert. La tâche est d'autant plus difficile que les connaissances non explicitées peuvent également être évidentes pour le cognicien. Le risque est grand d'omettre ces connaissances dans la base, où elles sont aussi indispensables au fonctionnement du système que n'importe quelle connaissance très élaborée.

Ceci nous amène à décrire plus précisément ce qui reste à faire, une fois les informations recueillies, pour obtenir une base de connaissances viable. Loin d'être terminé, le travail du cognicien entre dans une phase consacrée à la mise en forme des connaissances.

Devant une base de faits conséquente, il se trouve d'abord face à la difficulté d'éliminer les incohérences et les contradictions ; la situation est alors paradoxale, la base de règles est, théoriquement, constituée de « granules » de connaissances indépendants, dont il ne doit pas, à l'avance, préjuger de l'utilisation. Cependant, il lui faut vérifier, dans un même

temps, que des conflits insolubles entre les règles ne peuvent surgir dans aucun cas de figure.

La mise en forme des connaissances est donc un travail extrêmement long et délicat qui demande beaucoup de précaution, le cognicien ne doit ni omettre de connaissances sous-jacentes au discours de l'expert, ni le trahir.

Le recueil et la constitution de la base sont d'une durée variable selon l'étendue du domaine et le degré de précision auxquels désirent parvenir les concepteurs du système ; une base de connaissances n'atteint jamais l'état fini, elle peut toujours être améliorée et complétée afin que les problèmes auxquels est confronté le système soient traités dans toute leur complexité.

La représentation des connaissances

Le choix de la méthode de représentation est déterminant : elle doit être adaptée au domaine et au raisonnement des experts travaillant sur ce sujet.

Différentes techniques sont utilisées dans le domaine de l'IA. La plus familière est celle des règles, dites de production, qui sera présentée plus longuement ci-dessous, mais la technique des frames et celle des réseaux sémantiques ne sont pas sans intérêt pour le sujet qui nous concerne.

Les réseaux sémantiques

Ils sont ainsi nommés car ils se représentent sous la forme de réseaux dont les nœuds figurent des concepts, et les arcs reliant ces nœuds, les relations entre les concepts.

Prenons un exemple : Jean possède un chien nommé Médor, noir, et de race épagneul. L'ensemble des épagneuls est un sous-ensemble des chiens. Jean, Médor, noir, épagneul, et chien sont des concepts ; possède, est, est élément de, est sous-ensemble de, sont des relations entre ces concepts (fig. 1).

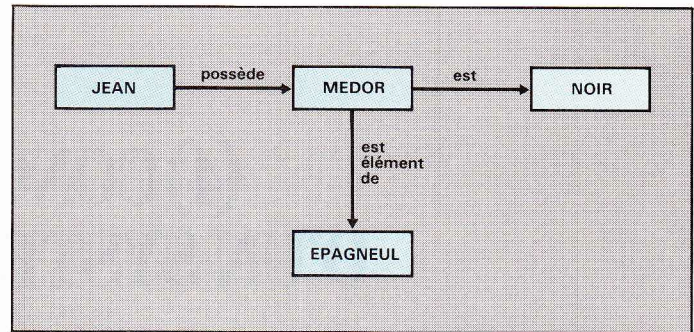


Fig. 1. — Relation et concept définissent un réseau sémantique.

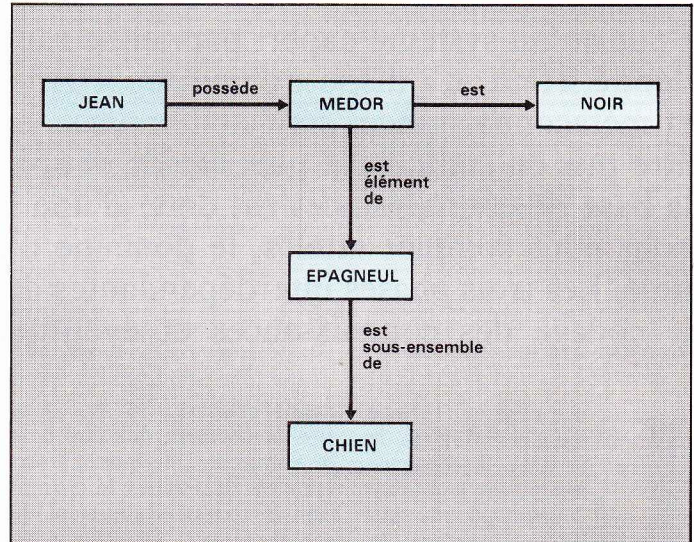


Fig. 2. — Déduction d'une conclusion grâce à un réseau sémantique.

Cette représentation suggère le mode d'inférence utilisable sur un tel réseau : il suffit de suivre le lien « possède », à partir de Jean, les liens « est élément de » et « est un sous-ensemble de », pour déduire que Jean possède un chien (fig. 2).

L'héritage des propriétés est une autre forme fondamentale de déduction sur un réseau sémantique. Par exemple : Médor est un épagneul, il « hérite » donc de toutes les propriétés de l'épagneul ; comme l'épagneul est un sous-ensemble des chiens, Médor « hérite » donc de toutes les propriétés des chiens. Il faut traiter cette notion avec de grandes précautions : l'héritage des propriétés n'est possible qu'à travers certaines relations : les relations « est un élément de » ou « est un sous-ensemble de ». C'est ainsi qu'il est ridicule d'affirmer que Jean est noir, alors qu'il est pourtant en relation avec ce concept, à travers les relations « possède » et « est ».

Les réseaux sémantiques

permettent également de représenter des concepts beaucoup plus généraux et abstraits.

Il semblerait que la faiblesse de ce type de représentation réside dans la difficulté de représenter la notion de conjonction des hypothèses, nécessaire aux systèmes experts, mais qu'il convient particulièrement au système de gestion de dialogue en langage naturel (ex. : Schollar, système d'EIAO).

Les frames ou cadres

Les frames ne se distinguent pas fondamentalement des réseaux sémantiques, ils en constituent une formalisation plus précise et plus rigoureuse.

Un frame est une structure qui permet de décrire un concept selon un modèle ou prototype. Sa définition comporte la référence au prototype, un identifiant unique, et un ensemble de valeurs.

Ainsi, le prototype « personne » (fig. 3) comprend toutes les propriétés de toute en-

INTELLIGENCE ARTIFICIELLE

| | |
|----------------------|----------------------|
| personne | |
| nom : | chaîne de caractères |
| prénom : | chaîne de caractères |
| sexe : | élément de [F, M] |
| année de naissance : | entier |

Fig. 3. — Définition d'un cadre définissant un concept.

| | |
|----------------------|--------|
| personne : 1 | |
| nom : | Dupond |
| prénom : | Jean |
| sexe : | M |
| année de naissance : | 1947 |

Fig. 4. — Occurrence d'un cadre (ou frame) : ici personne 1.

| | |
|----------------------|----------------------|
| personne | |
| nom : | chaîne de caractères |
| prénom : | chaîne de caractères |
| sexe : | élément de [F, M] |
| année de naissance : | entier |
| conjoint : | personne |

Fig. 5. — Enrichissement du cadre défini figure 3.

| | |
|----------------------|--------------|
| personne : 1 | |
| nom : | Dupond |
| prénom : | Jean |
| sexe : | M |
| année de naissance : | 1947 |
| conjoint : | personne : 2 |

Fig. 6. — Occurrence du nouveau cadre.

Script Cinéma

Acteurs : client, ouvreuse...
Objets : caisse, ticket, place, film...
Événements :
Le client entre dans le cinéma.
Le client va à la caisse.
Le client paie le ticket.
L'ouvreuse déchire le ticket.
L'ouvreuse choisit une place.
Le client regarde le film.
Le client quitte le cinéma.

Fig. 7.

tité de type « personne » : son nom, son sexe, son âge, sa taille. Le frame « personne 1 » (fig. 4) est une occurrence de type « personne », et d'identifiant 1 ; ses caractéristiques spécifiques sont les valeurs attribuées aux propriétés nom, sexe, âge, taille du prototype associé. En cela, rien ne distingue le frame de l'occurrence d'un tableau de données.

La représentation des relations par les frames est réalisée par certaines propriétés, qui sont en fait des références à d'autres frames, de même type ou de type différent. Par exemple, enrichissons le prototype « personne » en lui ajoutant la

propriété « conjoint » (fig. 5). Pour le frame « personne 1 », de type « personne », la propriété « conjoint » prend la valeur « personne 2 » (fig. 6). Pour en connaître les caractéristiques, il suffit de chercher le frame « personne 2 », et d'en lire les valeurs.

Les propriétés attachées aux frames sont donc de deux types :

- les propriétés terminales prennent des valeurs qui sont directement utilisables, chaînes de caractères, entiers, ou des mots reconnus par le système ;
- les propriétés de relations permettent d'établir des liens entre les frames. Les valeurs prises par ces propriétés sont alors des références à d'autres frames, et non des valeurs directement utilisables par le système.

Le modèle peut se compliquer encore par la possibilité de typer les propriétés des prototypes, par des références à d'autres frames dits de définition. Ainsi, la propriété « nom », de type chaîne de caractères, pourrait être typée par

un frame « mot » de propriété « nature » et de valeur « chaîne de caractère », et dont le prototype serait la frame « propriété ». Le réseau ainsi constitué permet non seulement de rendre compte des entités et des interrelations, mais aussi du codage même des propriétés sous forme de chaînes de caractères, d'entiers ou de réels...

■ L'attachement procédural

L'utilisation de ce type de représentation par un système informatique nécessite que l'on puisse créer des frames et en valoriser les propriétés de façon dynamique. Pour cela, on fait correspondre à certaines propriétés des procédures qui permettent de leur affecter, dynamiquement, des valeurs. On peut également leur associer des procédures à activer lorsque des valeurs sont présentes. Elles permettent alors d'effectuer des contrôles et des comparaisons. On peut, par exemple, prévoir dans le frame « personne » le contrôle de la date de naissance qui doit être inférieure à la date du jour. Ceci constitue l'attachement procédural. Cette dernière caractéristique fait des frames beaucoup plus qu'une base de données relationnelle, puisque l'intégration, dans le système, d'appels à des procédures d'affectation et de contrôle de valeur devient possible. Cette utilisation conjointe des modes procéduraux et déclaratifs n'est pas sans danger pour la modularité de la structure de la base de connaissances. Nous reviendrons sur ce problème.

Les scripts ou schémas

Ils décrivent des situations stéréotypées sous la forme d'une succession d'actions et comprennent aussi une liste des acteurs et des objets nécessaires à la description de la situation (ex. : fig. 7). Un script peut faire référence à d'autres scripts. Ils sont utilisés essentiellement pour traiter des situations simples, décrites en langage naturel.

Les règles de production

Ce mode de représentation des connaissances est utilisé par

de nombreux systèmes experts. Les règles de production sont de la forme :
SI (prémises) ALORS (conclusions).

Le membre gauche décrit les conditions dans lesquelles peuvent se déduire les conclusions du membre droit de la règle. Des opérateurs logiques de conjonction (et) ou de négation (non) peuvent relier des prémisses entre elles.

Dans le cas le plus simple, prémisses et conclusions sont des faits de la base, cette représentation constitue alors une application concrète de la théorie mathématique de la logique des propositions, mais si l'on y ajoute l'utilisation de la notion de variable et des quantificateurs mathématiques « il existe » et « quel que soit », les règles de production s'utilisent selon la logique des prédicats, ou logique du premier ordre, beaucoup plus riche. Ainsi, en logique des propositions, nous ne pouvons écrire de règle plus générale que :

Si infarctus alors surveiller tension artérielle
en logique des prédicats, on pourra écrire :
Si problème cardiaque alors surveiller tension artérielle
où « problème cardiaque » est une variable pouvant prendre différentes valeurs, dont « infarctus » par exemple.

L'intérêt des règles de production est de permettre de représenter les informations indépendamment de la manière de les utiliser. Ce mode de représentation, de type déclaratif, fait des règles des « granules » de connaissance, indépendants, et donc théoriquement utilisables par différents interprètes.

Choix d'un mode de représentation

Le concepteur de système cherche à trouver la meilleure adéquation possible entre le mode de représentation et la typologie du domaine. Dans notre système expert, le choix de la méthode de représentation s'est fait selon d'autres critères. Nous nous en sommes remis à l'intuition des experts médecins qui, lors des entretiens, expri-

INTELLIGENCE ARTIFICIELLE

maient naturellement leur expérience sous la forme « si... alors ». Il semble que cette formalisation convienne à la nature des connaissances du domaine médical concerné. A cet argument s'ajoutait celui de la viabilité d'un autre système expert, célèbre celui-là : Mycin. Le choix s'est donc porté sur les règles de production. La modularité, la neutralité, la souplesse d'utilisation de ce mode de représentation, sont des qualités qui le rendent adapté à l'usage qui devait être fait des connaissances. De plus, la littérature, prolixe sur le sujet, se révéla être une aide précieuse.

Choisir différents modes de représentation pour différents types de connaissances paraît être une règle judicieuse ; mais dans le cas présent, le choix unique des règles de production se justifie par leur facilité de manipulation et la possibilité de coder différents types de connaissances sous cette forme.

Typologie de la base de règles

Comme décrit ci-dessus, l'élément de base du système de production est la règle de production dont les prémisses et la conclusion sont des faits de la base.

ex. : si diabète alors contre-indication pilule normodosée et contre-indication minipilule

Mais pour optimiser le fonctionnement du système expert, il est nécessaire de déterminer des métarègles. Celles-ci sont de deux types : les métaconnaissances et les règles stratégiques.

Les métaconnaissances

Une métaconnaissance est une connaissance sur des connaissances de niveau inférieur. Ici, les connaissances de niveau inférieur sont des connaissances élémentaires, de type médical. Les métaconnaissances apportent des informations sur la façon de les utiliser. Elles sont nécessaires, par exemple, pour choisir dans une situation donnée entre toutes les règles actives, qui ne sont pas toutes intéressantes au même degré. La métaconnaissance

permet d'explicitier une stratégie de choix. D'autres méthodes sont utilisables :

– Choisir la première règle applicable revient à laisser le choix au hasard. A chaque modification du système, la première règle pourra changer de rang.

– Choisir la règle utilisée dans le délai le plus court est une possibilité, mais ce critère ne se justifie généralement pas.

– Choisir la règle qui cite le fait le plus récemment obtenu ou attribuer un ordre d'importance aux faits et choisir la règle qui cite le fait le plus important sont d'autres stratégies.

Mais l'inconvénient majeur des méthodes citées ci-dessus est qu'elles ne permettent pas d'explicitier sous la forme de connaissances le critère de choix des règles à appliquer ; or, cette stratégie est elle-même une connaissance d'expertise : elle exprime le raisonnement de l'expert, la représentation sous forme de métaconnaissance se justifie de ce fait amplement. De plus, elle permet de travailler sur un système de production qui reste modifiable sans que le mode de raisonnement n'en soit affecté. Les métarègles peuvent s'exprimer sous la forme de règles de production. La stratégie de choix n'est plus alors interne au moteur d'inférence, mais devient une connaissance d'expertise à part entière.

Sous-systèmes

L'utilisation de métaconnaissances dans ce système expert est liée à la notion de sous-expertise ou expertise locale. Les métaconnaissances sont de la forme : si infarctus alors envisager les règles relatives aux problèmes cardiaques

Ces règles forment un sous-ensemble de règles indépendantes des autres en ce sens qu'elles forment un sous-système dont les entrées sont des faits issus d'autres règles, et les sorties, des faits utilisables par d'autres règles ; mais les faits intermédiaires, obtenus à l'intérieur du sous-système, ne sont utilisés que dans le sous-système. L'intérêt de constituer ainsi des sous-ensembles de connaissances est grand : les

sous-systèmes peuvent travailler en parallèle et s'échanger des résultats par l'intermédiaire de la base de connaissances, mais les modes d'obtention restent mutuellement inconnus des différents sous-systèmes, on aboutit ainsi à la création d'un véritable système multi-expert.

Les métaconnaissances rendent prioritaires un ensemble de règles, et non une règle élémentaire, les choix de stratégies deviennent ainsi plus clairs et plus efficaces. Un système de production trop volumineux, composé de centaines de règles, devient plus manipulable si les règles sont classifiées dans des sous-systèmes.

Ce mode de fonctionnement, outre une segmentation de la base, permet de gagner un temps précieux par l'utilisation en parallèle des sous-systèmes.

Les règles stratégiques

Nommées également procédurales, ces règles permettent d'exécuter, dans un contexte donné, un algorithme, composé d'une suite d'instructions. Il s'agit ici d'une séquence de règles à appliquer dans un ordre donné.

L'utilisation de règles procédurales dans un système expert est controversée ; la base de connaissances, doit être, en effet, composée de connaissances totalement indépendantes entre elles et sans lien avec les méthodes d'utilisation. Les problèmes ne sont pas posés et décrits à l'avance par le concepteur du système expert, qui doit donc pouvoir s'adapter à tout problème nouveau, interne au domaine de connaissance. Cette approche déclarative s'oppose à l'approche procédurale qui propose, pour chacun des problèmes posés, une suite d'actions à entreprendre, déterminée à l'avance. Cette approche suppose de connaître tous les cas qui peuvent se présenter et de les décrire précisément. Dans une approche déclarative, le système peut travailler avec des informations incomplètes, et donner un résultat cohérent. L'utilisation du procédural suppose que toutes les informations nécessaires à l'exécution de l'algorithme soient connues.

L'intérêt du système expert est de pouvoir simuler le raisonnement d'un expert, même,

et surtout, s'il est placé dans une situation inédite. Dans certains domaines, il est impossible de décrire à l'avance tous les problèmes, et encore moins tous les algorithmes de résolution ; ceci est vrai, particulièrement, dans les domaines où tous les facteurs agissant ne peuvent être connus. Le domaine médical se prête donc à une approche déclarative. D'autre part, on doit théoriquement pouvoir enlever ou rajouter de nouvelles règles à un système de production, sans que celui-ci n'en soit affecté. Seule une approche déclarative garantit l'indépendance des données.

Cela étant posé, il a pourtant fallu avoir recours à des règles stratégiques ou procédurales. La nécessité en est apparue au cours des entretiens qui ont porté sur l'interrogation du patient par le médecin avant l'examen médical. Que l'interrogatoire soit de type général, clinique ou biologique, le questionnement de l'expert suit un ordre de priorité, constant d'un entretien à l'autre. Une première règle, pour un nouveau patient, impose en début de consultation d'exécuter dans un ordre donné des règles relatives à l'âge, à l'état général, aux antécédents médicaux, au passé obstétrical... L'ordre dans lequel ces recherches doivent être exécutées est donc déterminé par cette première règle de type stratégique. Les connaissances que ces premières règles permettent d'obtenir sont essentielles et prioritaires pour que le système parvienne à un résultat. L'ordre dans lequel elles sont acquises correspond à l'ordre dans lequel l'expert les obtient au cours de l'entretien préalable avec le patient. On peut considérer que ces règles spécifiques ne devront subir que peu de modifications ou de rajouts ; l'utilisation du procédural est donc, ici, sans grande conséquence.

Les règles de définition

Nous avons, jusqu'à présent, défini deux niveaux de connaissances, un de base et un supérieur : celui des métarègles. Le niveau des connaissances de base peut lui-même être subdivisé en deux : celui des connais-

INTELLIGENCE ARTIFICIELLE

sances factuelles ou de définition, et celui d'expertise.

Par exemple, les règles :
si tension min > 9 et tension max < 15 alors hypertension artérielle

si taux de glucide < 1.3 /l de sang à jeun alors mauvais taux de glycémie

sont des règles de définition dont la validité n'est pas mise en doute. Les règles de ce niveau ne font pas progresser le raisonnement, elles expriment des faits admis de tous, qui ne sont pas du domaine de l'expertise. Elles constituent des descriptions de concepts, et ne sont donc pas des connaissances opérationnelles. La limite entre règles de définition et règles d'expertise n'est pas toujours nette selon les domaines dans lesquels elles sont utilisées. Dans le médical, la frontière est floue et cela peut être une raison d'adopter le même mode de représentation pour les deux types de règles.

Règles empiriques et règles rationnelles

Au niveau des règles d'expertise, on peut à nouveau distinguer deux types de connaissances : celles de type empirique et les celles de type rationnel. Les connaissances rationnelles sont des connaissances rigoureuses, démontrées de façon scientifique. Elles peuvent s'expliquer et se prouver, tels les théorèmes mathématiques, les formules de calcul de résistance des matériaux ou la température de fusion des métaux... Les connaissances empiriques sont obtenues, elles, par l'expérience, et qu'aucun argument de type scientifique ne peut infirmer ou confirmer. Ce sont des règles de bon sens admises de tous, ou des connaissances de type savoir-faire, tour de main, acquis par certains experts au cours de leur carrière. Même si les règles de bon sens sont utilisées par un grand nombre de personnes, elles n'en sont pas pour autant plus rationnelles que les règles empiriques des experts, bien au contraire.

Les connaissances de ce type sont plus ou moins nombreuses selon le domaine : si on veut construire un système expert capable de calculer la résistance

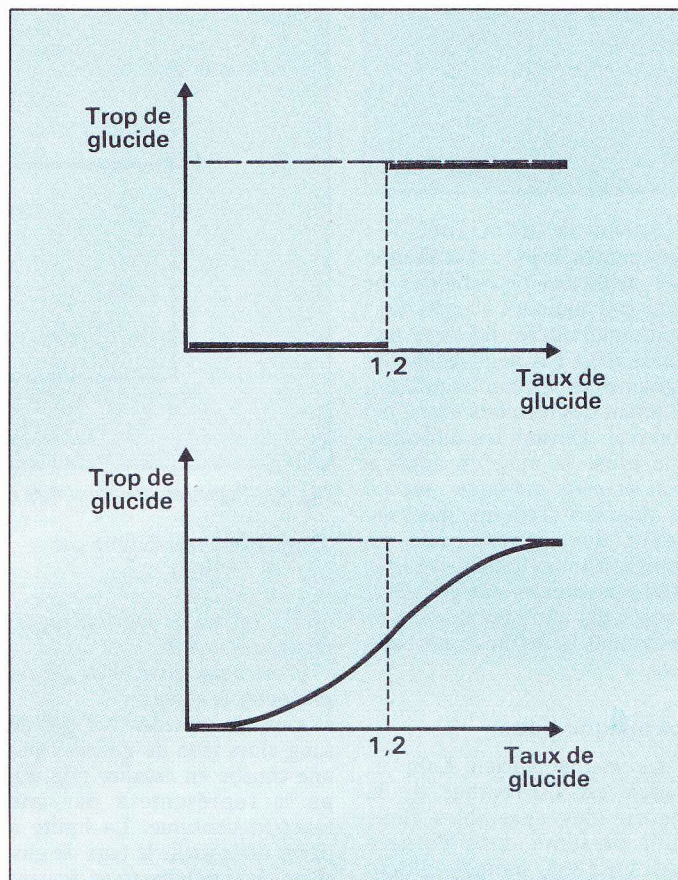


Fig. 8. - Visualisation d'une règle par une courbe en escalier (en haut) ou par une fonction continue (en bas).

d'une coque aux vagues, on fera surtout appel à des règles rationnelles ; au contraire, pour réaliser un système capable de jouer aux échecs rapidement ou de diagnostiquer les infections microbiennes du sang, il faut aussi utiliser des règles empiriques. C'est pour obtenir ce dernier type de règles que l'on fait appel à des experts de grande expérience, qui ont acquis des savoirs non livresques, vérifiés dans des situations variées. Dans les domaines où les règles rationnelles sont peu nombreuses, ce sont la précision et la justesse des connaissances empiriques transmises par les experts qui font qu'un système donne des résultats utilisables lorsqu'il est confronté à des situations nouvelles. En toute logique, le système utilisera en priorité les règles de type rationnel si elles peuvent s'appliquer.

Les métaconnaissances sont, ici, un moyen de favoriser l'application de connaissances rationnelles aux dépens de connaissances empiriques. La

limite n'est pas, ici encore, clairement définie entre le rationnel et l'empirique. La réalisation d'une base de connaissances peut être l'occasion de découvrir les bases théoriques qui fondent certaines règles empiriques.

Le raisonnement des médecins est difficilement formalisable selon la logique classique. Les symptômes utiles au diagnostic sont parfois ambigus, l'information nécessaire à la poursuite du raisonnement n'est pas toujours disponible, et en admettant que tous les signes soient exacts à 100 %, l'incertitude, inséparable du raisonnement médical, n'est pas toujours représentable par la logique classique.

Lors des entretiens que nous avons eus avec les experts, la question s'est posée clairement : comment rendre compte de notions floues telles que la jeunesse, l'obésité ou une alimentation équilibrée ; comment rendre compte des « environ », « à peu près », « généralement » ou « probable-

ment » qui parsemaient le discours de nos interlocuteurs ?

Cette question, tous les informaticiens qui ont cherché à traiter des connaissances humaines, ou à simuler des raisonnements humains, se la sont posée.

Les problèmes liés à la représentation de la connaissance ou des structures de raisonnement ne sont pas neufs, mais les théories classiques fondées sur la logique s'avèrent insuffisantes, lorsqu'il s'agit de traiter du raisonnement médical, par exemple.

Trois approches ont retenu notre attention : les méthodes probabilistes ou statistiques, proches parentes, les méthodes utilisant la logique floue, et enfin, celle utilisée par le système expert Mycin, qui fait intervenir le degré de confiance et de défiance. Ces trois modes de traitement de l'incertitude ne sont pas sans rapports, mais leurs spécificités les rendent plus ou moins appropriés au domaine.

La méthode probabiliste

Cette méthode utilisée par certains systèmes experts, est basée sur l'utilisation du théorème de Bayes. Etant donné un ensemble de diagnostics $D = \{D_i, 0 < i \leq n\}$, la probabilité d'observer un diagnostic D_i pour un patient présentant le symptôme S se calcule de la façon suivante :

$$P(D_i/S) = \frac{P(S/D_i) * P(D_i)}{\sum_{i=1}^n [P(S/D_i) * P(D_i)]}$$

où $P(D_i)$ représente la probabilité d'observer le diagnostic D_i dans une population donnée.

$P(S/D_i)$ représente la probabilité que le signe S apparaisse dans le cas où le diagnostic D_i est fait.

Les calculs de probabilité effectués par les systèmes experts probabilistes se basent sur la connaissance des probabilités d'observer un diagnostic lorsqu'un symptôme est présent. L'application du théorème de Bayes permet de calculer les probabilités modifiées par la connaissance d'un deuxième signe S_2 .

$$P(D_i/S_1 \text{ et } S_2) = \frac{P(S_2/D_1 \text{ et } S_1) * P(D_i/S_1)}{\sum_{j=1}^n [P(S_2/D_j \text{ et } S_1) * P(D_j/S_1)]}$$

Pour établir ces probabilités, il faut non seulement connaître les $P(D_j/S_1)$, mais aussi toutes les probabilités conditionnelles d'observer un symptôme, connaissant un diagnostic et un autre symptôme : $P(S_2/D_j \text{ et } S_1)$. Des statistiques n'existant pas généralement pour tous ces cas de figure, il faut demander aux médecins d'établir ces probabilités en se basant sur leur expérience clinique et leurs connaissances médicales. Ces données sont très difficiles à obtenir puisqu'elles demandent de quantifier des raisonnements intuitifs, et qu'elles ne correspondent pas toujours à une réalité observable.

Dans la mesure où ces données peuvent être obtenues, le système expert qui les prend en compte présente toutes les apparences de la rigueur scientifique : les diagnostics finaux sont affectés d'une probabilité calculée à partir des statistiques ou probabilités données par les experts médicaux. Mais il semblerait que ces systèmes prennent mal en compte les cas extrêmes, que les maladies rares soient difficilement détectées. De plus, le système est incapable d'expliquer les résultats de ses calculs, d'expliquer sa démarche à un médecin non spécialiste en probabilités.

L'application du théorème de Bayes suppose également que tous les signes soient indépendants et cela n'est pas toujours exact. La manipulation de probabilités « contraires » du type :

Si $P(\text{appendicite/douleurs abdominales}) = 0,3$

Alors $P(\text{appendicite/douleurs abdominales}) = 1 - 0,3 = 0,7$

se conçoit dans l'exemple ci-dessus, mais est plus difficilement concevable dans la situation suivante, apparue au cours de nos entretiens :

$P(\text{contre-indication pilule normodosée/tabagisme}) = 0,75$
d'où on en déduirait :

$P(\text{indication pilule normodosée/tabagisme}) = 1 - 0,75 = 0,25$
beaucoup moins justifiable.

Selon le domaine, les contraintes liées à l'utilisation des méthodes probabilistes ne sont pas toujours acceptables ; pourtant l'intérêt de cette méthode n'est pas négligeable, les systèmes expert qui les utilisent donnent des résultats tout à fait corrects. Devant les difficultés que cause la mise en application de cette méthode : recueil de données chiffrées, indépendance des signes, prise en compte des cas limites, d'autres systèmes experts ont été développés, qui n'intègrent que partiellement la méthode probabiliste.

La logique floue

Le mathématicien Lofti A. Zadeh est l'inventeur de la théorie des ensembles flous qu'il présente dans l'article paru en 1965, intitulé « Fuzzy Sets ». Cette théorie apporte une réponse au problème du raisonnement incertain : elle permet de rendre compte de ce qu'une part de l'information contenue dans une base de connaissances est imprécise et incomplète parce qu'elle manque de nuances. Si la combinaison logique de probabilités résolvait, en partie, le problème de l'incertitude du raisonnement, celui de la formulation en langage naturel ne l'était pas. Les nuances du langage naturel, telles que celles exprimées par : « moyennement », « généralement » ou « peu », sont représentables par la logique floue.

■ Les ensembles flous

La théorie des ensembles flous est une généralisation de la théorie des ensembles de Cantor. L'appartenance d'un élément v à un ensemble A n'est pas soit vraie, soit fausse, mais définie par un degré d'appartenance, compris dans l'intervalle $[0,1]$. A chaque sous-ensemble flou A , d'un référentiel E , est associée une application U , appelée fonction

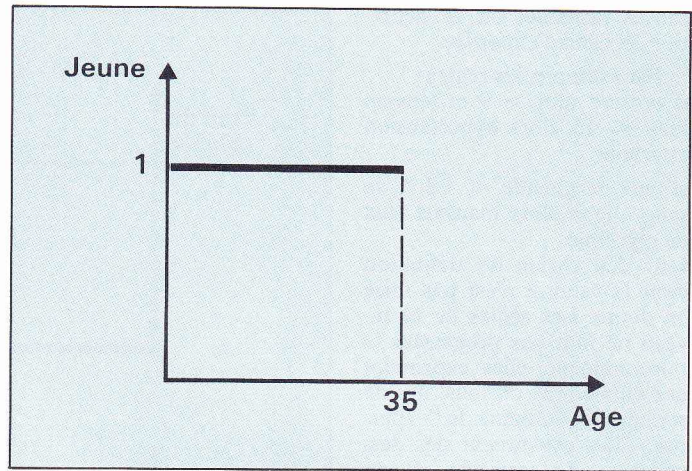


Fig. 9. - Représentation d'une règle en logique classique.

d'appartenance, définie par :

$$U_A : E \rightarrow [0,1]$$

$$v \rightarrow U(v)$$

où $U_A(v)$ est le degré d'appartenance de v à A .

C'est ainsi qu'au lieu de représenter la règle :

si taux de glucides $1,2 \text{ g/l}$ de sang alors trop de glucides par une courbe en escalier (fig. 8), on la représentera par une fonction continue. La limite à partir de laquelle le taux de glucides est trop important devient floue : on ne passe plus de l'état normal à l'état anormal lorsque le taux varie d'un centième.

Prenons un autre exemple :

Si âge ≤ 35 alors jeune se représente dans la logique classique par la fonction de la figure 9, et en logique floue, par la fonction de la figure 10 beaucoup plus vraisemblable.

Cette théorie permet également la représentation des concepts « nets », 1 étant l'appartenance entière, et 0, l'exclusion totale.

Les fonctions d'appartenance peuvent être définies par un tableau ou par d'autres fonctions. La théorie des ensembles flous permet donc de modéliser des concepts flous tels que l'hyperglycémie, ou la jeunesse. L'évaluation est numérique, ou faite par référence à d'autres concepts, qui peuvent également être flous.

■ Les ensembles ultra-flous

Les degrés d'appartenance d'un élément à un ensemble flou n'ont pas toujours à être exacts, le degré d'appartenance peut donc être lui-même un nombre flou. Les fonctions

d'appartenance sont des fonctions continues, le degré d'appartenance ne requiert donc pas la précision qui est nécessaire lorsque l'appartenance est représentée par la valeur unique 1. Les ensembles flous dont les fonctions utilisent des valeurs floues sont appelés des ensembles ultra-flous (fig. 11).

■ Les opérations

Les opérations de la logique classique sont toutes présentes en logique floue.

La complémentarité :

$$\forall v \in E, U_{A^c}(v) = 1 - U_A(v)$$

La conséquence :

$$\forall v \in E, U_{A \Rightarrow B}(v) = \min(1, 1 - U_A(v) + U_B(v))$$

L'inclusion :

$$A \subset B \Leftrightarrow \forall v \in E, U_A(v) < U_B(v)$$

L'égalité :

$$A = B \Leftrightarrow \forall v \in E, U_A(v) = U_B(v)$$

L'intersection ou conjonction :

$$\forall v \in E, U_{A \cap B}(v) = \min(U_A(v), U_B(v))$$

notée $U_A(v) \wedge U_B(v)$

L'union ou disjonction :

$$\forall v \in E, U_{A \cup B}(v) = \max(U_A(v), U_B(v))$$

notée $U_A(v) \vee U_B(v)$

La logique floue permet donc de réaliser toutes les opérations possibles en logique classique, mais aussi de rendre compte de l'imprécision de concepts et de manipuler cette imprécision avec des opérateurs logiques.

■ Les variables linguistiques

Les ensembles flous peuvent eux-mêmes être modifiés par des variables linguistiques. Zadeh en avait limité le nombre pour ne pas complexifier la re-

INTELLIGENCE ARTIFICIELLE

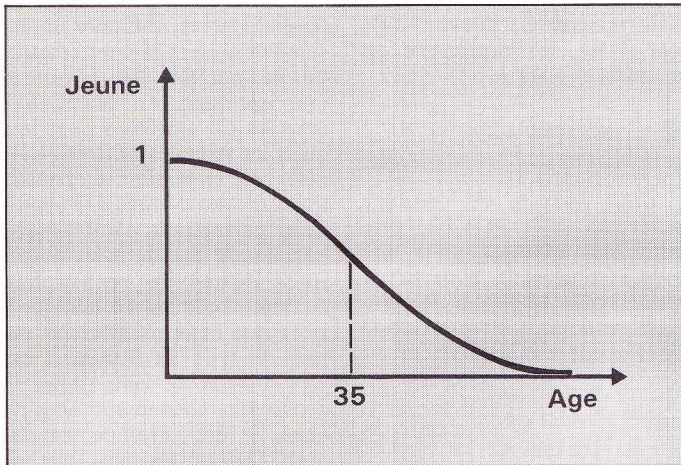


Fig. 10. - Représentation d'une règle en logique floue.

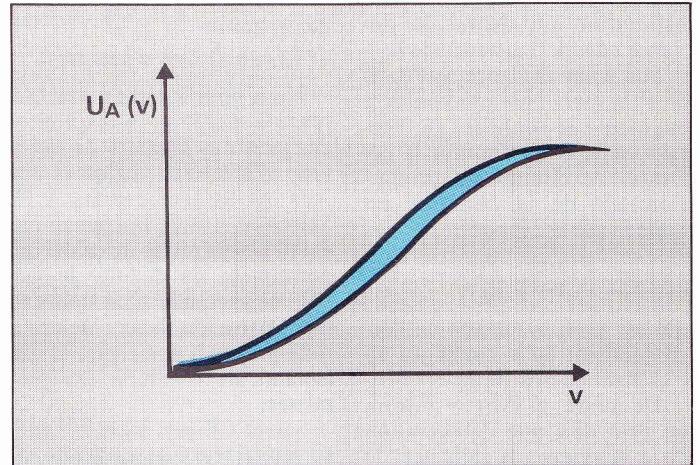


Fig. 11. - Matérialisation du degré d'appartenance à un ensemble ultra flou.

présentation de manière exagérée. L'ensemble des valeurs prises par les variables linguistiques est donc fini et dénombrable. « Vrai », « faux », « pas vrai », « très vrai », « plutôt vrai » sont quelques exemples de valeurs. Appliquées à un ensemble flou, ces valeurs permettent de déterminer de nouvelles fonctions d'appartenance, et donc de créer de nouveaux ensembles flous, dérivés de l'ensemble initial. Ainsi, les ensembles « très jeune », « assez jeune », « pas très jeune », ou « pas jeune », seront des ensembles dérivés de l'ensemble « jeune ». Leurs fonctions d'appartenance se calculent de la manière suivante :

$$U_{\text{très jeune}}(v) = (U_{\text{jeune}}(v))^2$$

$$U_{\text{assez jeune}}(v) = (U_{\text{jeune}}(v))^{1/2}$$

$$U_{\text{pas très jeune}}(v) = 1 - U_{\text{jeune}}(v)$$

$$U_{\text{pas jeune}}(v) = 1 - (U_{\text{jeune}}(v))$$

Dès lors que le sens de jeune est fixé, et que les règles de calcul sont établies, le sens de tous les dérivés sont déterminés.

■ Les quantificateurs flous

Les adverbes tels que : parfois, généralement, rarement, beaucoup, utilisés par les experts dans la formulation d'assertions dans leurs domaines d'expertises, sont d'une grande importance pour la représentation des connaissances ; la logique floue permet de les quantifier, et de représenter donc des nuances essentielles du langage naturel. Ces adverbes sont, en terme de logique floue, des

quantificateurs flous. D'où l'intérêt de l'utilisation de la logique floue dans les systèmes experts.

L'intérêt d'utiliser des valeurs floues et des prédicats flous dans le domaine des systèmes experts médicaux est grand : la normalité ne peut être représentée par des intervalles aux bornes précises ; la capacité d'un système à prendre en compte les cas limites dépend du choix de ces bornes. Si elles sont précises, le système rejette ou conserve systématiquement les valeurs proches des limites, selon qu'elles en sont en deçà ou au-delà. C'est le problème que posent en particulier toutes les mesures biologiques. Les limites de la normalité données par nos experts sont numériques, donc trop précises, puisque dans la réalité, le médecin ou le biologiste, contrairement à la machine, ne prennent pas en compte des valeurs limites pour fonder un diagnostic, mais analysent d'autres valeurs avant de parvenir à un résultat. La logique floue peut donc trouver, ici, un domaine d'application.

Comme le discours des experts ne comportait que très peu de ce que nous avons appelé les quantificateurs flous, un autre mode de représentation de l'incertitude a été choisi, plus adapté au domaine particulier qu'est la contraception féminine ; il s'agit des facteurs de certitudes.

Les facteurs de certitudes

Nous avons pu observer qu'une grande majorité des rè-

gles obtenues au cours des entretiens étaient données, par les experts, pour vraies à 100 %. Elles amenaient à des contre-indications formelles ou à des examens complémentaires à effectuer de manière impérative ; seules quelques règles ne présentaient pas ce caractère catégorique, règles sur la validité desquelles les experts ne parvinrent pas à un accord unanime. Les thèmes sur lesquels des divergences apparaissaient étaient relatifs au tabagisme ou au risque lié à la prise d'un contraceptif durant une longue période. Comme il y a, en la matière, autant de positions que de médecins - aucune donnée statistique sérieuse ne venant corroborer ces affirmations -, il a été décidé d'affecter à ce type de règles des facteurs de certitude, tels qu'ils ont été utilisés par le système Mycin.

Le système expert Mycin a été créé dans le cadre d'un programme de recherche médical au Stanford Research Institute. Les inférences logiques sont représentées sous la forme de règles de production, et pondérées par des facteurs de certitude. Devant les difficultés posées par l'obtention et l'utilisation de données statistiques ou probabilistes dans le domaine médical, E.H. Shortliffe et B.G. Buchanan ont mis au point une méthode de traitement de l'incertitude basée sur le degré de confiance que l'expert accorde aux règles qu'il propose. Les problèmes liés au recueil des probabilités conditionnelles étant ainsi éludés, il leur a fallu établir des formules

de calcul différentes de celle de Bayes pour permettre la combinaison de facteurs de certitudes.

Le système Mycin est utilisé de façon opérationnelle pour le traitement par antibiotiques d'infections microbiennes du sang, dont le germe responsable n'est pas complètement identifié. Devant l'efficacité de Mycin dans le traitement de l'incertitude, nous avons fait le choix d'utiliser la même méthode pour quelques connaissances incertaines, cela bien que les notions de facteurs de certitudes et les formules de calcul d'évaluation des conclusions des règles ne soient pas très satisfaisantes d'un point de vue rigoureusement scientifique.

■ Les degrés de confiance et de défiance

Les facteurs de certitude, pour le système Mycin, ont été établis à partir des degrés de confiance et de défiance tels qu'ils sont décrits ci-dessous.

$MB[h, e] = x$ indique que la mesure de l'accroissement de la confiance dans l'hypothèse h , basée sur l'évidence e , est x .

$MD[h, e] = x$ indique que la mesure de l'accroissement de la défiance dans l'hypothèse h , basée sur l'évidence e , est x .

$$\text{avec : } 0 \leq MB[h, e] \leq 1$$

$$\text{et } 0 \leq MD[h, e] \leq 1$$

Ces notations ne sont pas sans rapport avec les notations probabilistes.

Posons $P(h)$, le degré de confiance en h à un moment

quelconque, $1 - P(h)$ est une estimation du degré de défiance envers l'existence de h .

Soit e une évidence, si $P(h/e) > P(h)$ alors l'observation de e augmente la confiance de l'expert en h , et en diminue la défiance. la diminution proportionnelle de la défiance est donnée par le rapport :

$$\frac{P(h/e) - P(h)}{1 - P(h)}$$

qui est aussi la mesure de l'accroissement de la confiance en h sachant e : $MB[h, e]$.

De même, si $P(h) > P(h/e)$, on peut dire que l'observation de e augmente la défiance de l'expert envers h , et en diminue la confiance.

Ainsi le rapport :

$$\frac{P(h) - P(h/e)}{P(h)}$$

est la mesure de l'accroissement de la défiance en h sachant e , ou $MD[h, e]$.

Remarquons que si $P(h/e) = P(h)$, $MB[h, e] = MD[h, e] = 0$.

De manière générale, on a donc :

| | | |
|--------------|---------------|------------------------------|
| $MB[h, e] =$ | si $P(h) = 1$ | 1 |
| | si $P(h) = 0$ | $\max [P(h/e), P(h)] - P(h)$ |
| $MD[h, e] =$ | si $P(h) = 0$ | 1 |
| | si $P(h) = 1$ | $\min [P(h/e), P(h)] - P(h)$ |

Ces deux mesures peuvent donc être obtenues à partir de probabilités, qui, dans l'esprit des auteurs, ne reflètent que l'opinion des médecins, et non des mesures statistiques.

Le facteur de certitude est établi à partir de ces données.

$$CF[h, e] = MB[h, e] - MD[h, e]$$

où CF est le facteur de certitude de l'hypothèse h , devant l'évidence e , avec :

Si $CF[h, e] = 0$ alors e et h sont indépendants.

Dans la pratique, les experts doivent ajuster les facteurs de certitude de telle sorte que :

$$CF[hi, e] = 1$$

dans le cas où les hypothèses hi sont deux à deux indépendantes. Si par contre les hypothèses ne sont pas exclusives, la somme des $CF[hi, e]$ peut être supérieure à 1.

■ Calcul du facteur de certitude de la conclusion d'une règle

Nous nous sommes attachés, jusqu'à présent, à expliciter l'origine des facteurs de certitude affectés aux règles, et aux faits qui en sont déduits. E.H. Shortliffe et B.G. Buchanan ont établi des formules de calcul qui permettent de déduire le facteur de certitude d'un nouveau fait à partir du facteur de certitude propre à une règle, et des facteurs de certitude des prémisses.

Soient $CF[e1, E]$ et $CF[e2, E]$ les facteurs de certitude affectés aux deux prémisses, soit $CFr[h, e1 \text{ et } e2]$ le facteur de certitude affecté à la règle, on a :

$CF[h, e1 \text{ et } e2 \text{ et } E] = CF[h, e1 \text{ et } e2] \min(CF[e1, E], CF[e2, E])$

Prenons un exemple dans la base de règles du système Mycin :

Prémisses

La coloration de Gram de l'organisme est GRAM - ($CF1$

$$\max [P(h/e), P(h)] - P(h)$$

$$\min [P(h/e), P(h)] - P(h)$$

= 1). La morphologie de l'organisme est de type bâtonnet ($CF2 = 0.8$). L'aérobicité de l'organisme est aérobique ($CF3 = 0.6$).

Conclusion

Il existe une évidence fortement suggestive ($CFr = 0.8$), que la classe de l'organisme soit enterobacteriace.

$CF1, CF2$ et $CF3$ sont les facteurs de certitude affectés aux trois faits, ils ont été obtenus par calcul : par inférence d'autres règles, ou directement attribués par l'utilisateur.

CFr est le facteur de certitude affecté à la règle ; il est, lui, permanent. Calculons le facteur de certitude pour l'appartenance à la classe enterobacteriace, CFc :

$$CFc = CFr \times 0,6 = 0,8 \times 0,6 = 0,48$$

où 0,48 est le facteur de certitude attaché au nouveau fait, « la classe de l'organisme est enterobacteriace ».

■ Combinaison des conclusions de plusieurs règles

Deux facteurs de certitude différents, $CF1$ et $CF2$, peuvent être obtenus pour une même conclusion, à partir de deux règles ; le facteur de certitude résultant, CF , sera calculé de la manière suivante :

$$FC = FC1 + FC2 - FC1 \cdot FC2$$

pour $FC1 \geq 0$ et $FC2 \geq 0$

$$FC = FC1 + FC2 + FC1 \cdot FC2$$

pour $FC1 \leq 0$ et $FC2 \leq 0$

$$FC = (FC1 + FC2) / (1 + \min(|FC1|, |FC2|))$$

pour $FC1 \times FC2 \leq 0, |FC1| \times |FC2| \neq 1$

$$FC = 1 \text{ si } |FC1| \times |FC2| = 1$$

avec $FC1 \times FC2 \leq 0$

Comme on peut le constater, le facteur de certitude résultant est renforcé par l'application de plusieurs règles ayant la même conclusion. Ceci ne correspond pas toujours à la réalité : on peut trouver des situations dans lesquelles deux observations isolées confirment chacune la même hypothèse, mais où leur conjonction l'infirme totalement.

D'autre part, si l'utilisateur ne fournit pas, lui-même, un facteur de certitude pour le fait qu'il introduit, le système lui attribue par défaut un facteur de certitude égal à 1.

■ Comparaison des différents modes de représentation de l'incertitude

A la différence des systèmes purement probabilistes, le système Mycin peut expliquer quelles règles il utilise, et surtout par quels calculs il obtient les facteurs de certitude affectés aux conclusions.

Dans le domaine médical, les conclusions auxquelles parviennent les experts ne sont pas toujours obtenues de manière scientifique ou rigoureuse, ils ne se basent pas uniquement sur des connaissances livresques, mais aussi sur les connaissances empiriques, acquises par l'expérience. Il est donc plus intéressant, *a priori*, de représenter l'incertitude au moyen de degrés de confiance, dont les valeurs sont acquises par l'expert au cours de sa carrière,

qu'au moyen de probabilités ou de statistiques, difficiles à rassembler et dont la combinaison donne parfois des résultats inacceptables, en particulier dans les cas extrêmes pour lesquels les données chiffrées sont rares et entachées d'erreurs. Pourtant, les systèmes probabilistes permettent d'utiliser parallèlement des données objectives issues de recherches épidémiologiques et cliniques, alors que dans un système expert du type de Mycin, il est plus difficile de faire coexister des règles totalement empiriques, et des résultats statistiques. Utiliser les deux méthodes dans un même système serait sans doute la solution ; un pas en ce sens avait été fait par E.H. Shortliffe, lorsqu'il a écrit des formules permettant de reconvertir des règles de Mycin en énoncés de probabilités, mais les facteurs de certitude, même s'ils ne sont pas sans rappeler les probabilités, ne se combinent pas selon les mêmes règles de calcul.

Néanmoins, nous avons fait le choix de ce mode de représentation de l'incertitude : il donne des résultats probants dans le domaine médical et il correspond aux besoins de traitement de l'incertitude qui sont ceux de notre domaine spécifique. Le recueil des connaissances, la mise en forme et la constitution de la base, telles sont les longues étapes parcourues par le cognicien.

Claire Nédellec

Bibliographie

- « A model of inexact reasoning in Medecin ». E.H. Shortliffe, B.G. Buchanan, in *Mathematical Bio-sciences* 1975.
- « L'intelligence dévoilée ». C. Rémy in *Micro-Systèmes* n° 64, mai 1986.
- « Le virus I.A. ». *Terminal* n°s 31, 32, 33, février-avril 1987.
- « Les systèmes experts, principes et exemples ». H. Farreny, chez Cepades-Editions, 1985.
- « L'intelligence artificielle, promesse et réalités ». A. Bonnet, chez Inter-Edition, 1984.
- « L'intelligence artificielle ». *La Recherche* n° 170, oct. 1985.
- « Logique pour l'intelligence artificielle ». Turner, chez Masson, 1980.
- « Intelligence artificielle et systèmes experts ». A. Lescort, chez Cedic/Nathan, 1985.
- « La conception des systèmes intelligents ». I. Aleksander, chez Hermes Publishing, 1984.

BORDEAS LEVEURIE & LEO BURNETT



Marlboro

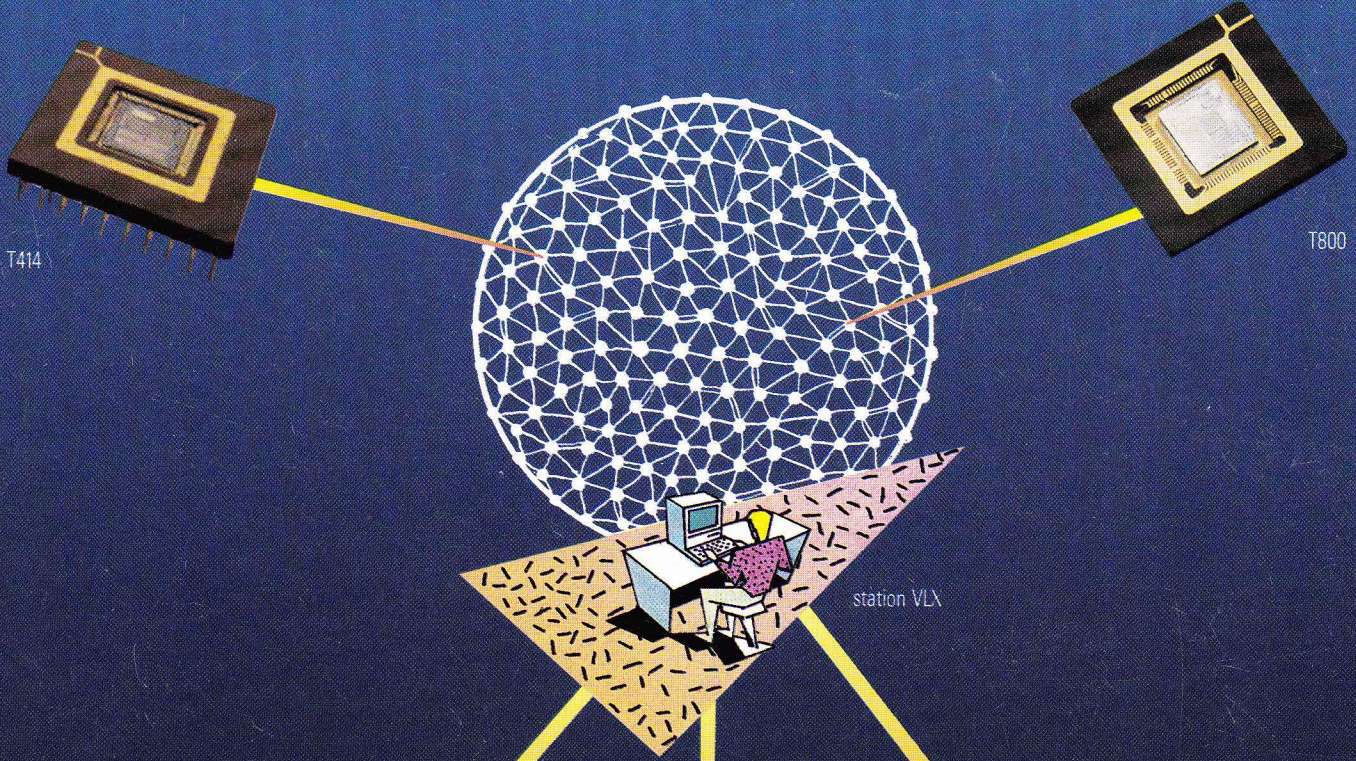
BRIQUETS FEUDOR EN VENTE DANS LES BUREAUX DE TABAC.



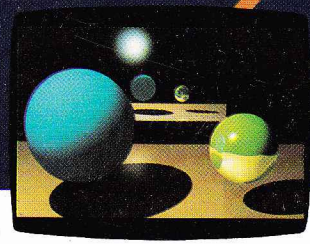
VOLVOX D'ARCHIPEL:

LES MICROS-SUPERORDINATEURS

130 Mips, 19.5 Mflops*



Synthèses d'images, systèmes dédiés et temps réel.



Traitement de l'image et de la parole, systèmes sécuritaires.



C.A.O. calcul scientifique, systèmes de gestion de bases de données.



Les micros-superordinateurs VOLVOX sont accessibles sous MSDOS™ ou UNIX™ et sont compatibles avec **TDS** d'**inmos**. Ils sont constitués d'une machine hôte compatible PC™ et d'une machine multi-processeurs de taille quelconque et configurable selon des topologies variées. Un VOLVOX peut ainsi offrir des performances très élevées (centaines de Mflops, milliers de Mips) à faible coût. Chaque élément du multi-processeurs est composé d'un **Transputer** (T414/T800) d'**inmos**, d'une mémoire locale (256 Ko à 8 Mo de RAM) et d'un système d'intercommunication haut débit.

* ou 4x13 Mega Whetstone : performances limitées par le nombre de slots d'un boîtier PC classique. Des boîtiers d'extension sont disponibles.

ARCHIPEL
architectures parallèles
9, av. du Rhône, 74000 Annecy tél. 50 52 87 32

ARCHIPEL, c'est une équipe d'ingénieurs, possédant un savoir-faire unique dans les architectures parallèles. ARCHIPEL propose :

- outils de développement **occam** / assembleur, outils de co-traitement, debugger et gestion d'écran. — des cartes additionnelles basées **Transputer** (bus PC, VME™).
- des cours de formation en programmation parallèle : **occam** / assembleur, réseau de **Transputers**, développement d'applications.
- un support technique pour la conception d'applications et de cartes spécifiques. — des stations de travail VLX, compatibles 80286 et 80386 parmi les plus performants.

SERVICE-LECTEURS N° 201