

INFORMATIQUE ET ÉDUCATION

LOGO ET LA NOTION DE DIDACTICIEL

par **BENOIT CÔTÉ**

professeur

Département de psychopédagogie
Université Laval

Le terme LOGO réfère à une conception de l'utilisation de l'ordinateur en éducation de même qu'à un langage de programmation qui en constitue le principal outil d'application. Cette approche a été développée depuis une quinzaine d'années par Seymour Papert et son équipe du Massachusetts Institute of Technology. Elle est centrée sur l'utilisation du potentiel éducatif de l'activité de programmer un ordinateur.

Ce texte tente de décrire ce que représente l'expérience d'entrer en interaction avec le système LOGO. Je l'écris à partir de mon expérience personnelle de travail avec des personnes de différents âges.

1. Rencontre avec une tortue

S'introduire à LOGO, c'est entrer en communication avec une entité qui parle un certain langage. Par exemple, on tape sur le clavier :

BONJOUR

Cela répond :

NE SAIS QUE FAIRE POUR BONJOUR

On tape :

COMMENT ÇA VA

Cela répond :

NE SAIS QUE FAIRE POUR COMMENT

Cette entité s'exprime en français mais elle ne semble pas comprendre beaucoup ce qu'on lui dit ! En fait elle possède un vocabulaire bien précis et ses propres règles de syntaxe.

La meilleure façon de s'introduire à LOGO est de s'adresser à la tortue. Celle-ci correspond à un petit triangle qui peut se déplacer sur l'écran graphique. LOGO comprend les mots AVANCE, RECULE, DROITE, GAUCHE, qui ordonnent à la tortue de se déplacer ou de pivoter sur elle-même. Celle-ci pos-

LOGO ET LA NOTION

sède également un crayon dont on peut spécifier la couleur, qui peut être à l'état « levé » ou « baissé », et qu'on peut même remplacer par une gomme à effacer. Nous avons donc tout ce qu'il faut pour produire divers dessins simplement en donnant à LOGO une série d'instructions utilisant les mots qu'il connaît avec les nombres appropriés.

Je n'ai encore jamais rencontré quelqu'un qui ne soit pas fasciné par cette activité, du moins au départ. J'ai vu un vice-recteur travailler ardemment à produire un hexagone, qu'il croyait être un octogone, avec la langue entre les lèvres, comme un enfant de huit ans. Malgré l'éventail des réactions possibles, il y a toujours un sentiment de puissance à constater qu'on est vraiment capable de faire fonctionner cette machine, mêlé à de nombreuses surprises et frustrations devant les résistances qu'elle nous apporte. On obtient rarement du premier coup le résultat que l'on souhaite. Il faut taper correctement les instructions, en laissant des espaces entre les mots, comme en français. Il faut réaliser que lorsque la tortue pivote elle ne se déplace pas, que ses angles sont souvent extérieurs aux figures, qu'elle tourne à sa droite et pas nécessairement à la nôtre... j'ai vu des mathématiciens s'y prendre à plusieurs reprises pour produire un carré.

Dans cet environnement, la notion d'erreur perd son sens péjoratif. Le temps de l'ordinateur n'est pas perçu comme une ressource rare et dispendieuse, à ménager à tout prix. Au contraire, on peut se permettre de faire toutes sortes d'essais et de recommencer autant de fois que l'on veut. Souvent nos erreurs produisent des effets intéressants qui nous conduisent vers de nouvelles pistes. En général, les enfants entrent facilement dans cet esprit d'exploitation. Les adultes sont plus réservés. Ils ont parfois besoin qu'on attire leur attention sur les diverses possibilités de la machine.

2. Apprendre en enseignant

LOGO peut faire énormément de choses mais il faut savoir les lui expliquer. De ce point de vue, programmer un ordinateur consiste à définir de nouveaux mots qui vont s'ajouter au vocabulaire déjà existant. Prenons un exemple.

Je veux lui faire tracer un escalier. Pourquoi pas ! Je tape :

ESCALIER

LOGO répond :

NE SAIS QUE FAIRE POUR ESCALIER

Donc il va falloir lui expliquer. Comment c'est fait un escalier ? Eh bien, c'est une succession de marches. Si je peux lui en faire tracer une, je n'aurai

pas de difficulté à en produire plusieurs, car LOGO connaît le mot **RÉPÈTE**. Je commence donc par lui expliquer comment tracer une marche. Je tape :

POUR MARCHÉ

Cet énoncé indique à LOGO que je veux lui définir un nouveau mot, soit **MARCHÉ**. Maintenant il s'agit de taper les instructions qui vont faire partie de cette définition. Disons qu'une marche consiste à faire avancer la tortue de 30 unités, à la faire pivoter de 90 degrés, puis à la faire avancer de nouveau, soit :

**AVANCE 30
DROITE 90
AVANCE 30
FIN**

LOGO n'exécute pas ces instructions mais les place en mémoire comme correspondant à la définition de **MARCHÉ**. Lorsqu'il rencontre le mot **FIN**, il répond :

MARCHÉ EST DÉFINIE

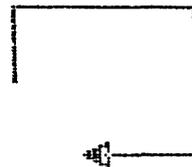
Ce nouveau mot a maintenant le même statut que les autres. On n'a qu'à le taper pour obtenir le dessin qu'il définit. Et on peut l'utiliser pour en construire d'autres. Ainsi je me définis un escalier de, disons, trois marches en tapant :

**POUR ESCALIER
RÉPÈTE 3 [MARCHÉ]
FIN**

Maintenant je suis prêt à essayer mon « escalier ». je tape :

ESCALIER

j'obtiens sur l'écran graphique le dessin suivant :



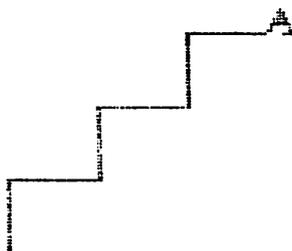
Surprise ! La première idée qui me vient à l'esprit est que la machine est défectueuse. Mais après quelques essais, je suis forcé d'admettre que ma définition est incomplète. Je reprends sur papier l'exécution des instructions de **MARCHÉ**, en indiquant pour chacune le changement de position ou d'orientation de la tortue qu'elle produit. Au moment de commencer la seconde exécution de **MARCHÉ**, je réalise que la tortue devrait pivoter à gauche avant de recommencer. Je transforme donc ma définition en :

POUR MARCHÉ
AVANCE 30
DROITE 90
AVANCE 30
GAUCHE 90
FIN

J'essaie de nouveau :

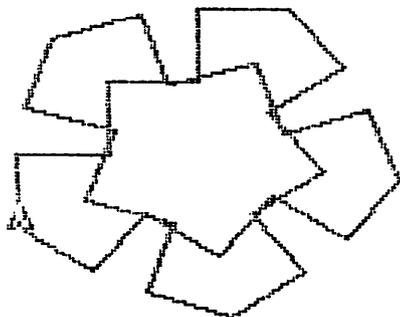
ESCALIER

Et ça « marche » !



LOGO est un langage indéfiniment extensible. Chaque nouveau mot peut être réutilisé de toutes sortes de façons et, dans le cas du graphisme, produire différents patterns plus ou moins complexes. Par exemple :

?RÉPÈTE 5 [ESCALIER DROITE 144] IMDESSIN



On appelle *procédure* une série d'instructions à laquelle on donne un nom, tel MARCHÉ ou ESCALIER. Programmer en LOGO consiste donc à construire, corriger, modifier et mettre en relation des procédures. C'est un processus qui peut devenir relativement complexe et qui se déroule habituellement en plusieurs étapes. C'est un peu comme expliquer quelque chose à quelqu'un. On commence à parler. Il réagit. On constate qu'il n'a pas tout à fait compris, qu'on s'est mal exprimé. On se reprend... On se retrouve ainsi dans une position d'enseignant. Il faut expliquer à LOGO ce qu'est un CARRÉ, un CERCLE, une MAISON, une SPIRALE...

Mais notre élève est aussi bête que logique. On oublie facilement que l'on a affaire à une machine à traiter des symboles, qui exécute littéralement ce que l'on tape, sans chercher à deviner nos intentions. Elle se charge cependant de nous rappeler que si on ne sait pas précisément de quoi l'on parle, ce n'est pas elle qui va le découvrir à notre place.

3. Construire et explorer

L'éventail des projets possibles avec la tortue est extrêmement vaste. L'ordinateur devient dans ce contexte un médium à travers lequel peuvent s'exprimer différentes attitudes et différents degrés d'ingéniosité. Cependant on peut identifier deux grands styles de fonctionnement permettant de caractériser la façon qu'a un individu de concevoir et de réaliser son projet, selon qu'il décide ou non au départ du résultat qu'il veut obtenir.

Dans le premier cas, on part avec une idée bien précise. On veut faire tracer un paysage avec une maison, un arbre, un chemin et un arc-en-ciel. Ou bien ce sera un bateau avec des vagues et un soleil. Ou bien une bicyclette très détaillée. Ou bien ses nom et prénom... Comme on ne peut pas tout faire en même temps, il faut décider par où commencer et dans quel ordre procéder.

À ce niveau, la nature modulaire et hiérarchique de LOGO devient un outil très puissant. On va suggérer à la personne de décomposer son problème en éléments relativement indépendants les uns des autres et de construire une procédure pour chacun de ces éléments. Elle peut ainsi s'assurer que chaque procédure produit l'effet désiré avant de l'insérer dans une superprocédure qui va rassembler plusieurs éléments. Par exemple, la procédure PAYSAGE va contenir les sous-procédures BATEAU, VAGUES et SOLEIL. La procédure BATEAU va contenir BASE, MAT, VOILE1, VOILE2 et DRAPEAU. La procédure VAGUES va contenir la sous-procédure VAGUE. Réaliser un projet correspond alors à construire un système hiérarchique de procédures et sous-procédures.

Cette activité nous fait prendre contact avec l'une des dimensions fondamentales de l'art de résoudre des problèmes. Il n'y a rarement qu'une seule façon de séparer un problème en sous-problèmes. Certaines sont plus efficaces et plus élégantes que d'autres. Par exemple, il est intéressant de savoir exploiter les symétries et autres formes de régularité des figures. Il est très utile de distinguer entre les instructions qui définissent une figure donnée et celles qui servent à aller placer la tortue à l'endroit où cet élément doit être situé. Le choix du point de départ et de l'ordre de construction des éléments entraîne également des conséquences habituellement non tri-

LOGO ET LA NOTION

viales sur la simplicité du système et le fait qu'il sera plus ou moins facile à tester et à corriger.

J'ai constaté que les gens qui réalisent ce genre de projets ont tendance à planifier leur travail avec de plus en plus de précision. Au départ ils fonctionnent parfois avec un plan qu'ils ne prennent même pas la peine de mettre sur papier. Ou bien ils tracent sur une feuille un dessin très approximatif de ce qu'ils veulent obtenir. Graduellement, ils en viennent à spécifier à l'avance les dimensions de leur figure, ce qui leur fait prendre conscience des difficultés de leur projet. Ils sont alors en mesure de concevoir de façon plus délibérée leur système de procédures et de choisir une représentation qui rend les problèmes plus abordables. Je ne dis pas que cette évolution se produit automatiquement. Mais on peut entrevoir en quoi de telles activités de programmation constituent un terrain propice au développement de ce type d'habiletés.

Le second style de projet est caractérisé par une démarche d'exploration et d'expérimentation. On ne part pas d'une idée bien précise. On se laisse guider par les différentes possibilités qui apparaissent en cours de route. On aime essayer toutes sortes de variations. On est fasciné par la recherche des possibilités de la machine et parfois très fier d'en démontrer les limites.

Je fais un carré. J'en place un second juste à côté. Je continue pour faire une rangée. Arrivé à la limite de l'écran, je découvre que lorsque la tortue déborde à droite elle réapparaît à gauche. Est-ce la même chose pour le haut et le bas ? Je vérifie et constate que c'est vrai. Je continue ma rangée pour rejoindre le premier carré tracé au centre. Ça n'arrive pas juste. Comment faire une rangée de carrés qui couvrent exactement l'écran ? Ça va me prendre les dimensions exactes de l'écran... On peut continuer comme cela indéfiniment. On ne va pas en rester à une seule rangée de carrés. On va couvrir tout l'écran. Et pourquoi des carrés ? Pourquoi pas des cercles, des rectangles, des pentagones et, si on veut vraiment être sophistiqué, des figures emboîtées à la Escher...

Les jeunes enfants (de tout âge) aiment souvent simplement explorer les nombres que l'on peut utiliser avec AVANCE. Ils reçoivent éventuellement le message :

TORTUE HORS LIMITES

Ils peuvent alors essayer de découvrir exactement où se trouve cette limite. De plus, ils peuvent réaliser que lorsque la tortue est orientée de façon horizontale ou verticale, le fait d'avancer d'un grand nombre de pas la fait revenir sur le même trajet et tracer une seule ligne. Ce n'est pas le cas dans les autres orientations.

Ils découvrent alors la possibilité de remplir complètement l'écran tout en passant à travers de fascinants effets de moiré qui comme tels peuvent être exploités. Et il y a aussi les nombres décimaux et les négatifs...

Les sujets intéressés par ce type de démarche d'exploration n'ont habituellement aucune difficulté à saisir le concept de variable. Ils se sentent rapidement limités par le fait de définir des figures avec des valeurs constantes pour les angles et les longueurs. Ils sont heureux d'apprendre qu'on peut remplacer un nombre par un nom de variable dans la définition d'une procédure, et ne spécifier cette valeur qu'au moment de l'exécution de la procédure. De cette façon, on va obtenir, par exemple, des carrés de différentes grandeurs en tapant :

CARRÉ 20
CARRÉ 50
CARRÉ 100

Ces sujets s'intéressent rapidement aussi à l'utilisation de RÉPÈTE pour générer toutes sortes de patterns. On les introduit alors à la récursion, qui est la possibilité pour une procédure de s'utiliser elle-même comme sous-procédure. Ce mécanisme, couplé avec l'utilisation de variables, devient un outil très puissant pour générer et explorer des figures arbitrairement complexes, et de façon générale répéter des processus tout en y introduisant des variations.

Par exemple, après avoir constaté qu'on génère un carré avec :

RÉPÈTE 4 [AVANCE 50 DROITE 90]

et un triangle avec :

RÉPÈTE 3 [AVANCE 50 DROITE 120]

on voit apparaître une forme générale que l'on peut inscrire dans une procédure à trois variables avec l'instruction :

RÉPÈTE : FOIS AVANCE : L DROITE : A

On peut alors essayer différentes valeurs pour ces variables. On constate que pour obtenir un polygone fermé, le produit de la valeur de : FOIS, soit le nombre de fois que l'on répète les instructions AVANCE et DROITE, et de la valeur de l'angle : A doit être de 360. Mais on peut se demander ce qui va se produire si on choisit pour : A un nombre qui ne divise pas 360, par exemple 144. On ne peut pas répéter des instructions 2.5 fois ($360 \div 144 = 2.5$). Alors qu'est-ce qui va se passer ? Est-il possible d'obtenir une figure fermée dans ce cas et si oui, quelle va être cette figure ? (Voir le dernier exemple avec ESCALIER dans la section précédente.)

Pour continuer ce projet, on peut définir une procédure récursive qui va répéter les instructions AVANCE et DROITE mais en introduisant à chaque coup un accroissement portant soit sur la longueur, l'angle ou les deux. On génère de cette façon toute

DE DIDACTICIEL

une famille de nouvelles figures dont on peut explorer les propriétés.

À ce niveau, l'exploration devient de l'expérimentation. On voit apparaître toutes sortes de phénomènes qu'on peut chercher à expliquer. Mais on réalise rapidement qu'il faut procéder de façon systématique. On arrive difficilement à comprendre quoi que ce soit si on laisse toutes les variables varier en même temps. Il faut conserver des traces de nos observations, les représenter sous forme de tableaux, chercher des régularités, formuler des hypothèses et les vérifier à l'aide de nouvelles observations. On se retrouve donc au cœur d'une démarche de recherche expérimentale.

4. LOGO sans la tortue

La tortue ne constitue qu'un sous-ensemble de LOGO. Elle définit ce que Papert appelle un micro-monde, soit un petit univers de connaissances avec lequel on peut entrer en relation de façon active et personnelle, et qui dans ce cas nous fait prendre contact avec un certain nombre d'idées puissantes en mathématique et en résolution de problèmes, tout en incluant une composante d'expression artistique. Mais en même temps, les activités avec la tortue constituent un excellent moyen d'apprendre la programmation.

LOGO est un langage de programmation complet et en réalité très sophistiqué. C'est un produit du langage LISP, le langage le plus utilisé dans les travaux en intelligence artificielle. En LOGO, on peut effectuer des opérations non seulement sur des nombres et des chaînes de caractères, mais également sur des ensembles formés de ces objets, que l'on appelle des *listes*. Celles-ci constituent un moyen puissant et efficace de créer et de manipuler des structures d'information. De plus, les procédures LOGO peuvent elles-mêmes être représentées et manipulées sous forme de listes, ce qui permet de créer des systèmes qui passent facilement de la construction à l'interprétation d'expressions symboliques.

La génération de phrases au hasard constitue un excellent projet pour s'initier aux listes tout en abordant des contenus de français. Il s'agit essentiellement de placer dans la mémoire de l'ordinateur un ensemble de mots et de lui définir des règles lui permettant d'en choisir et de les combiner pour faire des phrases. Voyons un exemple.

Je fais mémoriser à LOGO une liste de mots en tapant :

```
RELIE « MOTS [BATEAU CHÈVRE TOMBER  
AVEC ADMIRABLE  
LA PETIT]
```

RELIE est un mot LOGO comme AVANCE ou RÉPÊTE. L'action qu'il déclenche consiste à associer un nom à un objet et à placer ces informations en mémoire. Si je veux les utiliser par la suite, je peux taper quelque chose comme :

ÉCRIS EXTRAIS : MOTS

Ici LOGO va chercher l'objet associé à la variable : MOTS, soit notre liste de mots. Il en extrait un élément au hasard et affiche cet élément sur l'écran. Le mot EXTRAIS ne fait pas partie comme tel du vocabulaire primitif de LOGO. Cependant à ce niveau j'ai trouvé utile de le donner directement à mes étudiants, sans leur faire construire.

Nous avons donc le moyen de générer des mots au hasard. Évidemment on est intéressé à générer des phrases qui contiennent plus qu'un mot. Pour ce faire, nous utilisons un primitif LOGO qui justement s'appelle PHRASE. Ainsi les instructions suivantes vont générer deux phrases de respectivement deux et quatre mots :

```
ÉCRIS PHRASE EXTRAIS : MOTS EXTRAIS :  
MOTS
```

```
ÉCRIS (PHRASE EXTRAIS : MOTS  
EXTRAIS : MOTS EXTRAIS :  
MOTS EXTRAIS : MOTS)
```

On constate assez rapidement que les phrases obtenues ne sont pas très intéressantes. Le choix et l'assemblage au hasard de mots tirés d'un grand sac constituent un modèle pour le moins simpliste de la production de phrases. C'est ici que le projet commence. Comment améliorer cette situation ?

Une première approche consiste à distinguer certaines catégories de mots en construisant des listes distinctes pour les noms, les adjectifs, les articles, les verbes... Mais il faut aussi tenir compte des relations entre ces catégories. Comme avec la tortue, il est utile de séparer le problème en sous-problèmes et de se construire un petit système de procédures et sous-procédures.

On peut par exemple définir une procédure qui va piger au hasard dans les listes appropriées pour générer une séquence comprenant un article, un adjectif et un nom accordés au masculin singulier. Une fois qu'on a réussi cette première procédure, on en construit d'autres sur le même modèle de façon à varier le genre et le nombre. On peut alors définir une superprocédure qui va avoir accès à la liste de ces procédures, va en choisir une au hasard et l'exécuter.

Si on exécute cette superprocédure une première fois, puis on choisit au hasard un verbe, puis on l'exécute une seconde fois, on obtient une phrase de la forme :

LOGO ET LA NOTION

LE NOIR CHEVAL MONTE LA PETITE CÔTE

Évidemment les phrases générées dépendent des listes de mots que l'on fournit à l'ordinateur. Mais on obtient facilement des choses amusantes et parfois poétiques.

Ce projet peut devenir fort complexe. En réalité, il constitue une excellente introduction au domaine de l'intelligence artificielle. Cependant il est toujours possible de le limiter à une forme abordable pour à peu près n'importe qui. On peut même fonctionner de façon très intéressante avec des procédures qui n'utilisent que le primitif ÉCRIS. Quel que soit le niveau où on se situe, on se trouve plongé dans un monde de langage où on est forcé de préciser nos notions grammaticales, de la même façon que la tortue nous aide à préciser nos concepts de géométrie.

Les possibilités de projets de programmation sont pratiquement illimitées. On peut en concevoir qui touchent à des notions de mathématique du niveau élémentaire jusqu'à universitaire. On peut construire des dialogues sur à peu près n'importe quel sujet. On peut programmer toutes sortes de jeux. Nos seules limites sont reliées à notre ingéniosité, notre patience et la quantité de mémoire de notre ordinateur.

5. Et la notion de didacticiel

Souvent, lorsqu'on parle d'applications pédagogiques de l'ordinateur, on établit une distinction entre l'apprentissage de la programmation, que l'on voit comme une sorte de sensibilisation au monde de l'informatique, et l'utilisation de l'ordinateur comme moyen d'enseignement. Dans ce dernier cas, on réfère à ce qu'on appelle des *didacticiels*, soit des logiciels qui habituellement consistent à enseigner directement certaines notions spécifiques.

Du point de vue de LOGO, cette distinction n'est pas nécessaire et en fait conduit à des conséquences plutôt négatives. D'une part on obtient des didacticiels très rigides, qui laissent peu de place à l'activité de l'utilisateur et véhiculent une conception restrictive de ce qu'est l'apprentissage et la connaissance. D'autre part, on a tendance à considérer la programmation comme une nouvelle discipline, qui va exiger ses propres spécialistes et ajouter une autre cloison dans un univers scolaire déjà trop compartimenté.

L'existence de LOGO démontre qu'il est possible et souhaitable d'intégrer ces deux pôles. La tortue constitue une sorte de didacticiel ouvert en mathématique, soit un ensemble d'activités pouvant faire intervenir des notions mathématiques des niveaux élémentaire jusqu'à universitaire. Évidemment ces activités impliquent certaines interventions didac-

tiques et pédagogiques. Ce n'est pas la machine qui présente ce qu'est un rayon, un angle obtus ou le théorème de Pythagore. L'ordinateur devient plutôt un outil au service d'une pédagogie axée sur la réalisation de projets et la résolution de problèmes.

LOGO étant un langage extensible, on peut le façonner en fonction de divers objectifs de type didactique. Il s'agit de construire des procédures que l'on va fournir directement aux étudiants et qui vont servir de points de départ pour certaines activités portant sur des thèmes précis. Certains systèmes contiennent actuellement la possibilité de faire fonctionner simultanément une trentaine de tortues, de spécifier leur forme et leur couleur, et de les mettre en mouvement. Ces systèmes ont été utilisés pour des projets reliés à la physique, la biologie, les arts visuels et l'apprentissage de la lecture et de l'écriture. J'ai esquissé dans la section précédente ce que pourrait être un projet faisant appel à des notions grammaticales. Il y a davantage à faire pour créer un micro-monde de français qui soit aussi riche et accessible que celui de la tortue. Ce travail sera grandement facilité lorsqu'on aura introduit dans LOGO un système de traitement de texte. Il est possible aussi d'ajouter des procédures permettant la représentation et la manipulation de réseaux d'information, et qui pourraient être utilisées dans des projets en sciences humaines.

Le fait d'inclure un didacticiel à l'intérieur d'un langage de programmation du type LOGO nous garantit à la fois une certaine qualité de *plasticité* et diverses possibilités de *transfert*. En effet, non seulement peut-on exécuter le système mais aussi et surtout le développer et le modifier, justement parce qu'il est fait de procédures que l'on peut utiliser pour en définir d'autres ou que l'on peut modifier à son gré. De plus, les mêmes outils de programmation et souvent les mêmes stratégies de résolution de problèmes sont applicables pour des projets portant sur des thèmes très variés. Il y a là une base très intéressante pour comprendre que par exemple apprendre des mathématiques, de la grammaire et de la musique ne constituent pas nécessairement des tâches très différentes.

Bibliographie

- ABELSON, H., *Apple Logo*, Byte/Mc Graw-Hill, 1982.
- ABELSON, H. et diSESSA, A., *Turtle Geometry*, MIT Press, 1981.
- BOSSUET, G., *L'ordinateur à l'école*, P.U.F., 1982.
- LAWLER, R.W., *Designing computer-based microworlds*, Byte, août 1982, 138-160.
- PAPERT, S., *Jaillissement de l'esprit : ordinateurs et apprentissage*, Flammarion, 1981.
- SOLOMON, C., *Introducing LOGO to children*, Byte, août 1982, 196-208.