



**HAL**  
open science

## Programmation à l'école : chassées par la porte, elles rentrent par la... fenêtre !

Paul Delannoy

► **To cite this version:**

Paul Delannoy. Programmation à l'école : chassées par la porte, elles rentrent par la... fenêtre!. Revue de l'EPI (Enseignement Public et Informatique), 1996, 82, pp.79-89. edutice-00001115

**HAL Id: edutice-00001115**

**<https://edutice.hal.science/edutice-00001115>**

Submitted on 14 Nov 2005

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**NOTIONS DE PROGRAMMATION  
À L'ÉCOLE ÉLÉMENTAIRE :  
CHASSÉES PAR LA PORTE,  
ELLES RENTRENT PAR LA... FENÊTRE !**

**Paul DELANNOY**

**RÉSUMÉ**

Les enfants plongés dans une activité de construction de micro-robots pilotables par ordinateur auxquels on propose de passer pour cela par un langage informatique, même très simple, sont - presque tous - bloqués par l'apprentissage syntaxique, plus que par l'aspect conceptuel d'une telle activité ; ils disposent en effet, pour intégrer celui-ci, du support sensible de leur activité de montage et de la motivation forte que l'on peut résumer par « je vais réussir à faire bouger quelque chose en le commandant par l'ordinateur. »

Toutefois, des principes d'ouverture, comme celui que propose Logo en permettant à l'enfant de nommer *ad libitum* les procédures dont il a besoin, sont ressentis par les mêmes enfants comme simplificateurs.

La conclusion pédagogique qui a été tirée de ces remarques préliminaires est qu'il semble souhaitable de proposer à l'enfant une activité de construction du logiciel de pilotage comparable à celle qu'il vient de vivre en construisant le micro-robot de son choix ; tout en respectant, d'une part, les notions fondamentales de programmation qui sont implicites dans cette activité (affectation/nomination, boucle, test/attente d'événement, sous-programme, séquentialité de l'exécution, variables...) et, d'autre part, l'origine de la difficulté rencontrée, qui est la nécessité, pour atteindre l'objectif assigné, de passer de la réalité physique (via une description symbolique de celle-ci), à une construction formelle à base de symboles (le programme) dont l'efficacité se juge ultérieurement, lors de son exécution.

L'« idée simple »<sup>1</sup> qui est alors née est qu'il suffit de proposer à l'enfant de choisir, dans une fenêtre d'ordinateur qui les lui présente, les symboles correspondant aux effecteurs et aux capteurs qu'il a effectivement montés sur son robot ; à ce moment même, on lui demande de donner un nom à chaque élément qu'il sélectionne ; nom qu'il tape au clavier : l'étape de formalisation via la description symbolique est, très souplement, commencée.

En approfondissant l'idée pour en faire la base d'une réalisation pratique, la possibilité d'aller ainsi jusqu'à l'élaboration d'un véritable langage, spécifique à l'échange dans le triplet (enfants, ordinateur, robot) s'est révélée. Comme ce langage est construit sur une base écrite qui permet la communication extérieure, ce n'est plus « un langage que personne ne parle »<sup>2</sup> Ce n'est pas non plus un langage réduit à un alphabet graphique<sup>3</sup>, dont la transmissibilité reste souvent aléatoire, et qui peut représenter une difficulté supplémentaire dans la représentation des notions nouvelles.

## CADRE DU TRAVAIL

L'élaboration de cette idée est due aux conclusions faites par un instituteur maître formateur de l'IUFM des Pays de la Loire à l'occasion d'une recherche (RESCOUSSE) sur l'usage de réseaux d'échanges entre classes utilisant des équipements de robotique pédagogique. Cette recherche est supportée par l'IUFM des Pays de la Loire et le LIUM<sup>4</sup>; elle a débuté en février 1991 et doit se terminer avec cette année scolaire 1995-1996.

Sa première phase a consisté en l'introduction dans l'atelier informatique de l'école d'un ensemble de pièces Fishertechnik et d'une possibilité de pilotage sous Logo. Les maîtres et les élèves, rodés à ce type de situation, ont découvert ensemble les possibilités et les limites de cette solution. L'organisation de l'atelier a été définie, des projets de construction sont nés, mais, dans le contexte de l'équipement de l'école en ordinateurs puissants, Logo a été ressenti comme un frein.

---

1 suivant le vocabulaire de S. Papert, dont les questions fondamentales sont toujours d'actualité dans ce domaine.

2 Cf. Paul Delannoy, colloque AFDI - AQOPS, Montréal, avril 1992

3 Comme le préconise, par exemple, l'institut TECHNION, en Israël, avec le logiciel Technologica.

4 Laboratoire d'informatique de l'Université du Maine, Le Mans.

## HYPOTHÈSES

Devant cette constatation, la question d'expliquer la situation pour tenter d'avancer dans la recherche a été posée. Voici l'essentiel de la réponse : pour programmer un mouvement de robot dans le contexte d'un langage de programmation très général comme celui-là, les commandes de base proposées ont deux défauts majeurs :

- d'une part elles ont la rigidité syntaxique *a priori* de toute instruction de programme,
- d'autre part elles s'adressent directement à des symboles (M1 pour un moteur, E4 pour un palpeur...) et non pas au « moteur de la grue » ou au « pare-chocs avant du camion » ;

Ces deux caractéristiques entraînent l'enfant à une activité de symbolisation et de représentation mentales qui, si elle n'est pas forcément hors de sa portée, reste pénible et qu'il ressent comme l'éloignant de son enthousiasme initial.

Or, qu'est-ce qui provoque cet enthousiasme ? Si l'on pose la question, plusieurs constantes sont présentes dans les réponses des enfants : construire le robot, faire bouger le robot en le commandant, programmer l'ordinateur pour que le robot bouge, sont les plus évidentes ; une constante moins perceptible au départ, et qui se révèle au cours de l'activité, entraînant souvent une acquisition majeure dans l'apprentissage de l'enfant, est la perception du robot « comme une marionnette » : les enfants n'ont pas souvent l'envie ni l'idée de monter des capteurs !

## MÉTHODOLOGIE

Avec la fin de cette première étape, l'équipe de RESCOUSSE a vu arriver deux éléments moteurs puissants : d'une part de nouveaux ordinateurs pour l'école, suffisamment puissants pour envisager de nouveaux programmes de pilotage et tournant sous Windows, d'autre part l'intérêt porté à ce travail par un collègue membre d'une petite équipe de développeurs désirant produire des logiciels éducatifs. La seconde phase, qui doit se terminer cette année, est donc axée sur la production d'un logiciel offrant à l'enfant d'aller de la phase de montage du robot à sa programmation dans un langage qu'il élabore lui-même (mais qui n'est plus Logo).

Pour tenter d'élaborer un logiciel qui les aide vraiment à passer de la construction matérielle à la construction du logiciel, il faut retenir les quelques éléments suivants :

- l'enfant aborde l'ordinateur naturellement, et se repère aisément dans un monde de symboles graphiques, surtout lorsque ceux-ci représentent clairement une réalité physique qu'il connaît ;
- c'est une difficulté pour lui que de distinguer action directe et action différée : s'il désigne par un symbole graphique un moteur qui est effectivement connecté à l'ordinateur, il ne veut pas attendre pour le faire fonctionner : un clic sur l'icône DOIT faire marcher le moteur dès ce moment là ;
- les notions de programmation qu'il va découvrir sont : la nomination des actions des moteurs et des états des capteurs, la séquence d'actions, le test d'événement, la boucle et le branchement (qui, pour l'enfant, reste une notion très naturelle), le sous-programme ;
- la notion de variable et d'argument reste plus éloignée, et ne se présente directement que lors de l'usage de l'instruction de temporisation, mais reste utile ;
- toutes ces notions se basent sur la définition et la manipulation d'objets non réels, dont l'existence n'est due qu'à la nécessité, pour programmer l'ordinateur, de transformer la réalité extérieure à celui-ci en symboles par lui manipulables ;
- a contrario, la représentation de ces objets dans les fenêtres de l'écran se doit de rester claire pour l'enfant.

Le choix d'un logiciel n'utilisant que des icônes (représentant plus ou moins bien les éléments réels et les éléments formels nécessaires à l'écriture du programme), même dans la construction du programme, n'est sans doute pas efficace au vu de ces éléments. De plus, pour que des groupes différents puissent échanger leurs solutions sur des problèmes de robotique voisins, il serait nécessaire, avec un logiciel ne proposant que des symboles graphiques, que ceux-ci soient figés ; on perdrait alors un degré de liberté important dans l'activité.

Les auteurs du logiciel développé dans le cadre de la recherche RESCOUSSE ont donc volontairement renoncé à faire un logiciel uniquement piloté par icônes, préférant mettre l'accent sur la description par nomination écrite des éléments du robot ; il y a donc dès le début

usage de l'écrit dans la représentation, c'est un point important aux yeux des concepteurs <sup>5</sup>.

Ce logiciel est développé par une petite équipe de Nantes qui s'appuie sur une jeune société de diffusion de logiciels éducatifs (il est donc couvert par un copyright et la description ci-dessous ne doit pas être pillée, merci). Le nom WINO ROBOT est en cours de dépôt à l'INPI.

## WINO ROBOT : LE LOGICIEL RÉALISÉ

Lorsqu'il a réalisé son montage électromécanique, et qu'il l'a connecté sur l'ordinateur de pilotage, l'enfant est invité à utiliser le logiciel pour réaliser cette phase de technologie de contrôle. Le logiciel a auparavant été préparé par le maître, qui a indiqué quel type de matériel et quelle interface sont utilisés pour cette session de travail. L'enfant se trouve alors devant une représentation de la boîte de pièces sur laquelle ne sont figurés que les éléments actifs (moteurs, capteurs, ...) disponibles.

### Phases 1 et 2 : Choix et description des éléments, des connexions

L'enfant dispose donc d'une représentation de l'ensemble des éléments d'un robot qu'il peut avoir relié à l'ordinateur. Il doit alors désigner ceux qu'il a effectivement utilisés, en leur donnant EN MÊME TEMPS un nom qu'il choisit (exemple : *moteur\_roues*, *moteur\_grue*, *pare-chocs avant*,...) et qu'il tape au clavier. Pour cela, il clique sur un des objets en bas, celui-ci se place en haut et le dialogue de saisie du nom est lancé, avec un nom « générique » proposé par défaut. Dans la figure ci-dessous, l'enfant a déjà placé un moteur et un capteur, et il est en train de placer une lampe.

---

<sup>5</sup> Un autre point mérite d'être noté ici : le paramétrage initial peut être effectué aussi bien pour le matériel Lego Dacta que pour le matériel Fischertechnik ; tout autre système de pilotage pourra venir s'y intégrer à la demande.

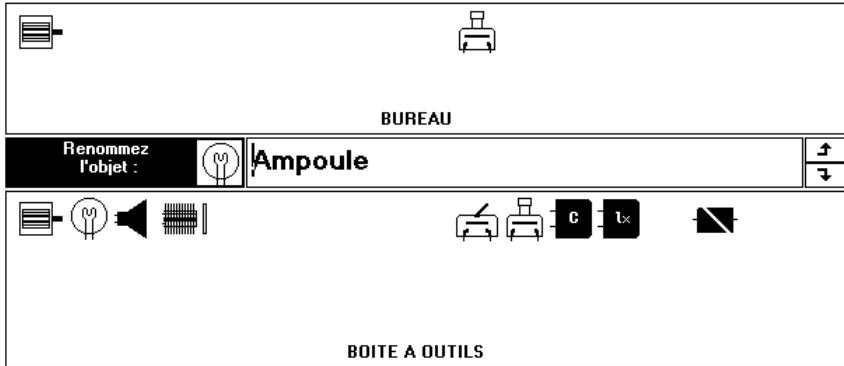


Fig. 1 : la boîte à outils de WINO ROBOT

Après cette phase qui sollicite l'enfant pour une première analyse de sa construction, il lui est demandé de placer chacun des éléments ainsi désignés, et nommés, sur une représentation des lignes de sortie et d'entrée disponibles avec l'interface qu'il utilise. La figure 2 illustre cette phase, avec un moteur connecté sur la ligne de sortie n° 4, et un capteur sur la ligne d'entrée n° 4, ainsi qu'un capteur analogique.

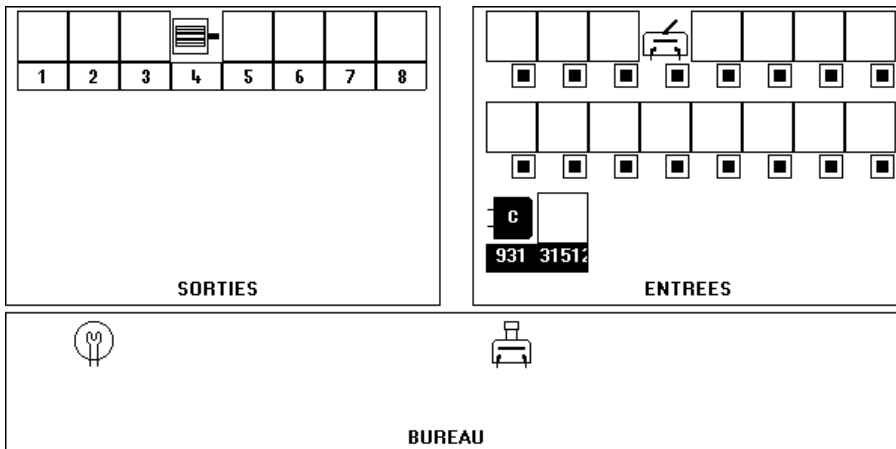


Fig. 2 : La phase de description des connexions

L'interface est évidemment active, si bien que cette description bénéficie totalement à la fois de la mémoire de l'enfant qui agit, qui peut avoir noté sur l'interface les lignes qu'il a utilisées, et de l'interactivité du procédé, qui permet de vérifier d'un clic (sur le n° de la ligne) le

fonctionnement d'un effecteur ou de voir l'information transmise par un capteur varier « en temps réel » (le carré sous le capteur binaire est vide ou plein, la valeur de l'entrée analogique est affichée).

Une fois ces deux étapes passées, la description ainsi obtenue du robot « aux yeux du logiciel » peut être enregistrée.

L'étape suivante va permettre de nommer également, pour chaque élément actif figurant dans la description : chaque état d'un capteur binaire (ex : *pare-chocs avant bute*, *pare-chocs avant libre*), chaque état d'un moteur (ex : *moteur\_roues\_AV*, *moteur\_roues\_AR*, *moteur\_roues\_ST*) La notion de fonction apparaît ici si l'on demande également de donner un nom à la valeur d'un capteur analogique (ex : TEMP ou LUMIERE).

Dans cette phase, qui peut se décrire comme la phase de définition du lexique du langage de programmation qui va se constituer, la partie haute de l'écran est représentée dans la figure 3 (par exemple, pour un moteur, une lampe, et deux capteurs binaires).

Cette zone haute récapitule les objets choisis puis branchés et testés lors des deux phases précédentes.

Un clic sur l'un de ces symboles d'objet déplace sur lui la marque (flèche verticale) du symbole en cours et lance le dialogue de nomination des états de l'objet physique qu'il représente (si celui-ci est sur une ligne d'entrée) ou de ses actions (s'il est sur une ligne de sortie).

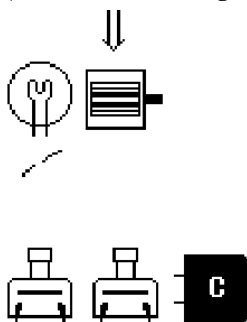


Fig. 3 : Les symboles des objets branchés

La partie basse de l'écran correspondante sert alors à saisir les descriptions écrites que l'enfant veut donner aux états des capteurs, et aux actions des moteurs. Elle est représentée ci-dessous pour le cas du moteur désigné dans la figure précédente. L'enfant peut passer aux autres objets grâce aux flèches *Avant* et *Après* ou en cliquant sur un autre symbole d'objet dans la partie haute.



Dans cette phase, comme dans toutes les autres, l'interface est active, et un clic sur les icônes du moteur à gauche de la figure 4 activera ou stoppera celui-ci, de même qu'un capteur verra son état actuel traduit par un point rouge sous celle des deux icônes correspondantes. C'est grâce à elle que le langage de programmation pourra comporter des instructions personnalisées, telle que *grue\_descend* ou *moteur\_roues\_en\_avant*, et des événements testables, tels *pare\_chocs\_avant* ou *le\_crochet\_de\_la\_grue\_est\_en\_haut*.

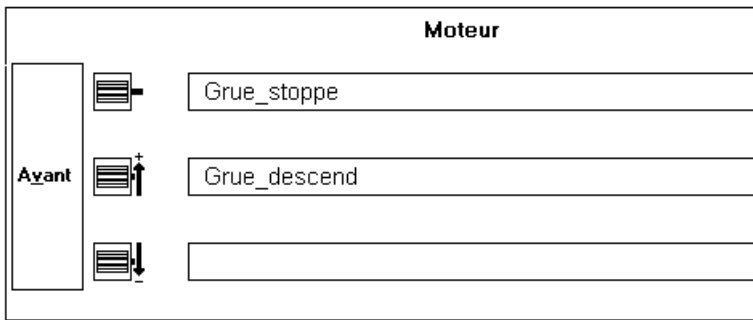


Fig. 4 : La phase de définition du lexique

Cette étape contient l'essentiel de l'innovation introduite<sup>6</sup> dans l'élaboration de WINO ROBOT ; en effet, le résultat de la description constitue la liste des **briques de base** logicielles disponibles pour l'assemblage du programme ; il y a donc bien eu séparation de la difficulté d'origine en deux composantes : celle de la description formelle dans le langage des éléments du robot, et celle de la description par anticipation de la séquence d'actions à exécuter.

Ce résultat, couplé au choix d'un petit nombre d'instructions de programmation et à leur mise à disposition sous forme d'icônes (voir ci-dessous), permet à WINO ROBOT d'offrir des succès très rapides aux enfants qui l'utilisent. Cette décomposition en deux parties clairement séparées de la difficulté identifiée au départ évite la nécessité précédemment imposée aux enfants de formaliser leur travail de programmation au niveau de l'interface (lignes d'entrée, lignes de sortie) ; elle prépare l'avènement de cette formalisation en conférant à celle-ci son vrai statut de métalangage par rapport à la description (en « langage courant ») du robot et de ses connexions... Cet avènement

<sup>6</sup> que l'on retrouvera sans difficulté, sous une autre forme, dans l'architecture du logiciel Roboteach © de Pascal Leroux, LIUM.

constitue une globalisation possible ultérieurement qui sera également facilitée par le fait que WINO ROBOT ne se limite pas à l'usage d'un seul type de matériel robotique.

Le résultat de ces premières étapes de description et de nomination, qui constitue pour le logiciel la description du montage de l'enfant, peut être enregistrée indépendamment de la suite du travail. Elle sert de base à la phase de programmation.

### **Phase 3 : l'abord de la programmation**

Le modèle utilisé pour la programmation est celui du programme séquentiel "linéaire sauf contrordre" (test, aller à, répétition, sous-programme), représenté verticalement, l'exécution s'effectuant de haut en bas, avec la possibilité de nommer les sous-programmes.

Cette phase se déroule dans une fenêtre d'éditeur, représentée ci-dessous (figure 5), qui présente en outre : à droite des fonctions générales (sortie du logiciel, enregistrement/relecture, exécution (simulée, réelle, pas-à-pas), ajout/effacement de lignes,...) et à gauche des éléments issus des premières phases.

Dans cette partie en haut à gauche, les icônes sont devenues actives : un clic sur l'une d'entre elles déroule un « pop-up » menu déroulant les noms des actions correspondantes donnés à la phase 3. Lorsqu'une icône de capteur est visible (c'est-à-dire lorsque l'on a commencé une instruction d'attente d'événement), un clic sur elle déroule les noms des états de ce capteur tels qu'ils ont été désignés dans la phase lexicale précédente.

Sous cet ensemble lexical ainsi regroupé, une série d'icônes constitue le menu des instructions de base (SI...ALORS...SINON, RÉPÉTER, ALLER Á, ATTENTE (temps, événement) création de sous-programmes,...) et dont la gestion ne nécessite donc qu'un minimum de - voire aucune - frappes au clavier.

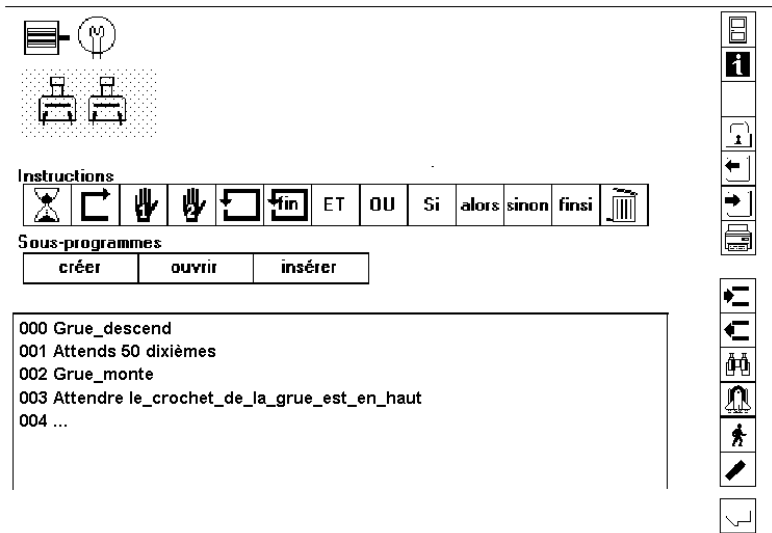


Fig. 5 : l'écran de programmation du robot

De gauche à droite :

- le sablier se lira ATTENDS et réclamera la saisie d'un nombre de dixièmes de secondes,
- la flèche “passant par la marge gauche” est le ALLER À,
- les mains correspondent à l'attente d'événement (état particulier d'un capteur)
- les deux flèches “passant par la droite” sont RÉPÉTER et FIN RÉPÉTER
- le reste des instructions est relatif aux conditionnelles.

Les sous-programmes se créent de la même manière, l'écran change simplement de couleur pour identifier la situation particulière.

Notons également la présence d'outils de gestion des capteurs de mesure analogique, qui permettent de faire un peu d'expérimentation assistée par ordinateur (ExAO) avec les enfants (par exemple : mesure continue de températures dans la classe, enregistrement au format tableur).

## CONCLUSION (PROVISOIRE)

Le logiciel est, au moment de la rédaction, en cours de finition dans sa version française. L'aide en ligne est rédigée en parallèle, et le produit final est prévu pour le début de décembre 1995.

La structure prévue doit permettre d'en établir une version en anglais, qui sera proposée au congrès ED-Media de juin 1996 à Boston.

Ce logiciel sera mis à la disposition de plusieurs classes dans l'académie, pour test et évaluation. Cette utilisation dans plusieurs classes du logiciel permettra de plus, par l'usage de réseaux d'échanges entre elles (dont l'existence couplée à celle des ensembles de robotique est un élément central de la recherche RESCOUSSE dans laquelle ce développement vient s'inscrire) :

- d'une part de comparer effectivement les réponses d'enfants différents à des problèmes technologiques voisins ou identiques,
- d'autre part de leur ouvrir la voie, sur la base de normes d'échanges qu'ils auront eux-mêmes négociées, à l'interprétation en commun de données variées sur un thème.

Cet échange sera présenté et confronté à d'autres expériences, en liaison avec l'organisation du Festival International des Sciences et de la Technologie de La Ferté Bernard, Sarthe, en mai 1995.

Paul DELANNOY

IUFM des Pays de la Loire,  
I.N.R.P. département TECNE,  
Laboratoire Informatique de l'Université du Maine,  
Ave O. Messiaen, B.P. 535 72017 LE MANS  
CEDEX

Tél : (33) 43 83 30 95 et (33) 40 16 30 01

Fax : (33) 43 83 35 65 et (33) 40 16 30 30

E-mail : [delannoy@corvette.univ-lemans.fr](mailto:delannoy@corvette.univ-lemans.fr)