

Langue et langages : quelques réflexions

www.adjectif.net/spip/spip.php



Pour citer cet article :

Nivat, Maurice (2015). Langue et langages : quelques réflexions. *Adjectif.net* Mis en ligne le 11 novembre 2015. [En ligne] <http://www.adjectif.net/spip/spip.php?article367>.

Résumé :

Maurice Nivat est un pionnier de l'informatique, membre correspondant de l'Académie des sciences. Il s'intéresse depuis de nombreuses années aux questions d'apprentissage et d'enseignement de l'informatique, dès l'école primaire.

Mots clés :

Langage, Réflexion, Langues maternelles



Langue et langages : quelques réflexions

par Maurice Nivat

Il est bien connu que l'informatique a donné naissance à une étonnante quantité de langages de programmation, il y en a des milliers et je pense qu'il en apparaît toujours de nouveaux.

Après bien des hésitations, des décisions ministérielles récentes vont faire que ces langages, du moins certains d'entre eux, fassent l'objet d'un enseignement pour les élèves du second degré et, avec eux bien sûr, la programmation ou l'art d'écrire, dans ces langages, des textes baptisés "programmes" qui permettent de faire réaliser des actions par des machines électroniques. Les programmes décrivent des façons de faire, que l'on appelle "algorithmes", d'un mot un peu barbare mais ancien puisqu'il apparaît dès le treizième siècle.

On peut difficilement programmer si l'on n'a pas une idée précise de l'algorithme qu'il s'agit de décrire, comme il est difficile de parler si l'on ne sait pas quoi dire. Et pour avoir une idée de l'algorithme que l'on va programmer, il est nécessaire d'avoir un but clairement défini et de savoir de quels moyens on dispose. La prolifération des langages de programmation vient de ce qu'il y a beaucoup de façons de décrire un algorithme comme il y a beaucoup de façons d'exprimer quelque chose que l'on veut dire, beaucoup de langages, c'est-à-dire de systèmes de notation possibles, dans lesquels on peut écrire, et aussi de nombreuses écritures possibles dans un même langage.

Je bondis toujours quand je lis, dans des manuels d'informatique, que le corrigé de l'exercice n° 17 page 234 est « le » programme suivant, l'exercice consistant à écrire un programme pour décrire un petit algorithme, décrit tant bien que mal en français augmenté de quelques formules plus ou moins mathématiques, en Python ou ce que vous voudrez.

C'est aussi absurde que le serait de dire qu'il y a un seul texte en français décrivant une situation donnée ou rendant compte de faits observés : il ne faut pas enseigner longtemps la programmation pour se rendre compte qu'à un tel exercice, les élèves peuvent apporter les réponses les plus diverses et, parfois, toutes exactes. La

correction des copies rendues s'apparente plus à celle des compositions françaises par le professeur de français qu'à celle d'un devoir de maths et peut se révéler cauchemardesque pour le prof, tout en apparaissant assez arbitraire à l'élève.

Dans la plupart des manuels, c'est avec les premières lignes écrites dans un langage de programmation que commence l'apprentissage de l'informatique et de la programmation et je pense que c'est grand dommage car les élèves, quels qu'ils soient, ont commencé à utiliser des programmes bien avant, quand ils ont appris à compter, à faire des additions et multiplications ou les règles de phonétique et de grammaire permettant de lire à haute voix un texte écrit en français, d'écrire sous la dictée. Ces programmes que les enfants apprennent (non sans mal souvent) dès le CP à l'âge de six ans sont écrits en français, seule langue ou langage qu'ils connaissent, et décrivent soit des algorithmes simples comme celui de l'addition, soit des algorithmes très complexes comme celui de l'orthographe.

En fait, les enfants ont dans la tête quantité d'algorithmes, plus ou moins précis, plus ou moins bien décrits, appris, connus, en français ou dans quelque chose de très près du français, un français élémentaire avec des ajouts de vocabulaire tenant au domaine dans lequel se situe l'algorithme, qui peut être celui du comportement en général, de la vie sociale, du jeu, du sport, de matières scolaires.

En commençant l'enseignement de l'informatique par la présentation d'un langage de programmation artificiel, on donne l'impression que l'informatique est un domaine nouveau, quelque peu ésotérique, concernant les ordinateurs et eux seulement, ce qui est méconnaître que la recherche d'algorithmes pour faire ceci ou cela, avec les moyens qu'on a, est une activité que tout le monde pratique et ce, depuis sa plus tendre enfance.

Dire qu'apprendre la programmation, c'est apprendre à « coder » est une absurdité, le premier codage que l'on apprend, c'est la langue maternelle qui n'est qu'un code, au sens strict, faisant correspondre, de façon très arbitraire (il y en a quand même plusieurs milliers dans le monde), des mots à des objets, des actions, des gestes, des sensations, des sentiments, des idées.

L'enfant est déjà habitué à toutes espèces de restrictions, distorsions, extensions de sa langue maternelle pour répondre aux divers besoins de communication rapide ou secrète ou de dénominations d'objets et d'actions liés à des activités ou des situations particulières qui ne sont pas dans le dictionnaire. Il a déjà senti qu'il y a des façons de dire l'histoire, ou la géographie, ou la science, ou la littérature qui ne sont pas les mêmes, même s'il est incapable de préciser en quoi ces façons diffèrent.

On lui a même appris à se servir d'un dictionnaire plus ou moins encyclopédique, et de divers livres de classe pour y rechercher et éventuellement trouver des informations en réponse à des questions qu'on lui pose ou qu'il se pose.

Je pense qu'il serait de bonne pédagogie de construire les connaissances de l'informatique qu'on cherche à donner aux enfants sur ce savoir. Il s'agit aussi bien de la connaissance explicite de quelques algorithmes, que de celle, diffuse, de beaucoup d'autres, en s'appuyant sur leur expérience déjà vécue des niveaux de langue, des changements de sens des mots et de phrases quand on change de contexte ou de type de discours.

Et c'est ce pourquoi je pense qu'il conviendrait de réfléchir à cette question, entre informaticiens, enseignants de diverses disciplines et spécialistes de pédagogie.

C'est en effet une question qui intéresse toute l'informatique, dans toutes ses innombrables applications, car un programme donné comme un texte à lire ne peut l'être, s'il peut l'être, que par des spécialistes qui y passent beaucoup de temps. Pour que les autres, non spécialistes, pour qui de nombreux programmes sont écrits, sachent ce qu'il fait, quelles données il faut lui fournir et quand, quel résultat on peut en espérer, dans quelles conditions il peut ou doit être utilisé, il faut fournir avec le programme des explications détaillées et compréhensibles en bon français, ce que nous appelons souvent la documentation.

Même entre eux, d'ailleurs, les informaticiens parlent des programmes qu'ils écrivent, échangent des informations ou des directives en français (émaillé de formules et de termes techniques), ils ne parlent pas un langage de programmation, ceux-ci ne sont pas faits pour être parlés.

Et hélas, souvent, ce discours en français sur les programmes est très généralement insuffisant, pas assez précis, voire ambigu, ou bien reste incompréhensible au profane (qui peut quand même être le donneur d'ordre) : des fiascos célèbres de systèmes onéreux qui n'ont jamais réussi à bien marcher ne s'expliquent pas autrement que par la mauvaise communication entre les commanditaires non-informaticiens et les exécutants informaticiens.

Le problème de savoir dire en français, dans la langue de tout le monde ce que fait l'informatique, ce qu'est un algorithme, ou comment marche un programme déborde largement le cadre de l'enseignement dans les écoles, lycées et collèges.

Dans le monde réel, celui de tous les jours, qui comme chacun sait est de plus en plus « numérique », l'informatique consiste toujours essentiellement à écrire des programmes qui décrivent des algorithmes, mais ceux-ci sont de plus en plus souvent destinés à être exécutés par des systèmes alliant des opérateurs humains et des machines (de tous poils, mécaniques ou électroniques) ou des programmes régulant et faisant fonctionner de tels systèmes dont la complexité va croissant (et dépasse de beaucoup la capacité de création ou de compréhension par un seul homme ou une seule femme).

La diffusion souhaitable d'une culture informatique à tous les jeunes scolarisés mais aussi tous les citoyens a pour but de permettre à tous de se diriger et se comporter dans un monde où le travail humain est inextricablement mêlé à celui de très nombreuses machines au sein de systèmes complexes dont le bon fonctionnement repose sur des algorithmes de plus en plus nombreux et sophistiqués.

La frontière mouvante où se situe l'innovation est celle qui passe entre l'homme et la machine. Une grande partie de l'innovation technologique ces vingt dernières années a consisté à élargir le champ d'action de la machine pour réduire la part de l'homme en augmentant la fiabilité, la sûreté et l'efficacité.

Le but de l'enseignement de l'informatique au plus grand nombre, qui n'écrira jamais de programme autre que jouet, est de faire comprendre ce mouvement d'informatisation galopante, de démythifier la machine et de rendre familiers les algorithmes, ces curieux objets qu'on ne voit jamais (comme les nombres), que nous utilisons tout le temps sans souvent nous en rendre compte et qui cependant, de plus en plus, nous gouvernent.

Il s'agit surtout de pouvoir parler, dans notre langue, en français, de l'informatique comme on peut parler de cuisine, de jardinage, de baignoles, de football sans être cuisinier, jardinier, constructeur de voiture ou joueur professionnel.

Le risque existe que les programmes et le dispositif prévu pour cet enseignement n'oublie le fait que la principale difficulté, et le principal intérêt, de la programmation, c'est de décider de ce que l'on veut faire faire à la machine et de rassembler les données et les moyens qu'il faut pour y parvenir, ceci incluant tous les morceaux de programmes déjà écrits par d'autres dont on dispose, qu'ils s'appellent logiciels, sous-programmes, routines, procédures ou applications.

Le plaisir que l'on peut prendre à participer à la construction d'un système complexe incluant une large part d'informatique (ce qui est le cas désormais de tous les systèmes un peu compliqués) s'apparente beaucoup à celui des gamins faisant du mécano : certes, il faut bien que tous les écrous localement soient serrés mais il ne faut jamais perdre de vue l'ensemble qui doit tenir debout et satisfaire aussi bien l'œil que la raison.