

Quelques principes fondamentaux pour toute initiation à l'informatique

Charles Duchâteau

► **To cite this version:**

Charles Duchâteau. Quelques principes fondamentaux pour toute initiation à l'informatique. Troisième rencontre francophone de didactique de l'informatique, Jul 1992, Sion, Suisse. pp.159-163. edutice-00359232

HAL Id: edutice-00359232

<https://edutice.archives-ouvertes.fr/edutice-00359232>

Submitted on 6 Feb 2009

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.

QUELQUES PRINCIPES FONDAMENTAUX POUR TOUTE INITIATION À L'INFORMATIQUE

Charles DUCHÂTEAU

Résumé : l'atelier proposé avait comme but essentiel de faire prendre conscience aux participants de quelques traits dominants de la discipline informatique qui en forment en quelque sorte des principes fondateurs et doivent dès lors constituer la trame de toute initiation à l'informatique et transparaître même lors de la simple utilisation d'outils logiciels.

Il est évidemment impossible d'aborder l'ensemble de ces grands principes pendant un atelier aussi court et plus difficile encore d'en rendre compte en si peu de pages. Je me contenterai donc ici de donner quelques grandes lignes de mon propos en renvoyant pour plus de détails à un article à paraître dans la revue de l'EPI.

MÉTHODOLOGIE

Plutôt que d'annoncer un contenu précis et de déflorer ainsi la recherche entreprise, il était préférable de proposer aux participants de préparer leurs réponses à quelques questions centrales et qui permettent d'aborder quelques principes importants. Voici une liste de ces questions, étant entendu que les réponses doivent adopter le point de vue des apprentissages souhaités :

1. Décrivez en une demi-page ce qu'est pour vous l'ordinateur, ou plutôt l'essentiel de ce que vous en avez découvert et aimeriez partager avec d'autres.
 2. Décrivez deux tâches qui vous paraissent actuellement hors de portée d'un traitement informatique.
 3. Un de vos amis qui ignore tout de l'informatique est prêt à passer avec vous une heure à travailler avec un ordinateur « pour en savoir plus ». Que prévoyez-vous pour lui ?
 4. A quelle condition une « information » peut-elle être « digérée » par un ordinateur ?
 5. Qu'appelleriez vous « manipulations de base d'un ordinateur » ?
- etc.

UN ORDINATEUR : UN INTERLOCUTEUR FORMALISTE ?

Les facettes retenues (qui n'en constituent pas une «définition») tiennent en peu de mots. C'est

- une machine
- à traiter des informations, DE MANIÈRE FORMELLE (ou FORMALISTE)
- pour autant qu'on lui ait préalablement indiqué comment mener à bien ce traitement.

L'atelier s'est surtout attaché à préciser, illustrer et clarifier le caractère « formaliste » des traitements effectués par l'ordinateur. C'est en classant des exemples, tirés de notre expérience «humaine» de traitement d'informations qu'on peut arriver à démarquer les traitements formels de ceux qui, pour l'instant du moins, semblent nécessiter un accès au «sens» des informations traitées. Ce classement débouche sur un tableau du type suivant :

	Traitement formel	Traitement non formel (Sens, signification, ...)
Traitements portant sur des nombres	- additionner par écrit - calculer une moyenne - ...	
Traitements portant sur du texte (langage)	- compter les mots d'un texte - fournir les fréquences de tous les mots d'un texte - conjuguer	- résumer un texte - chercher les mots-clés d'un texte - traduire un texte - ...
Autres	- dériver une fonction	

D'autres traitements d'informations sont actuellement difficilement classables, comme « corriger un texte », « calculer la primitive de n'importe quelle fonction », « vérifier des connaissances »,...

On constate, comme il se doit, que les traitements numériques sont, par nature, toujours formels et que les traitements formels (ou formalistes) portant sur des informations «textuelles » nous sont absolument inhabituels. Ce tableau peut bien entendu être étoffé.

L'informatique peut être vue comme une entreprise qui a pour objet de « faire reculer » la double ligne du milieu du tableau vers la droite pour englober toujours plus largement des traitements, qui paraissent jusque là faire appel au « sens », en les formalisant,...

Il est possible de se doter d'un critère qui aide à décider ce qui est formalisable et ce qui ne l'est pas (encore) : le critère du « copain portugais ». Il suffit d'imaginer qu'on souhaite s'adresser à un ami, borné et obéissant, parlant exclusivement une langue qui n'est pas la nôtre (par exemple le portugais). Toutes les explications que nous pourrions lui donner seront traduites en portugais. On constate aisément que toutes les manipulations portant sur des nombres pourront lui être fournies, et que, même s'il ne comprend pas ce qu'il fait, on pourra lui faire calculer une moyenne, lui faire trier des nombres, ... On pourra même lui faire conjuguer des verbes français : il ne saura même pas qu'il est en train de conjuguer, encore moins que c'est du français, mais s'il suit nos indications il écrira sans problème toutes les formes conjuguées, par exemple des verbes réguliers de la première conjugaison. Le problème change bien entendu si nous souhaitons lui faire résumer un texte français ou lui faire traduire un texte français en anglais ! On perçoit qu'il s'agit là d'une tâche insurmontable, puisque comme notre ami ne comprend pas le texte français, toutes nos indications ne peuvent s'appuyer que sur la forme des chaînes de caractères composant le texte (ce ne sont même pas des mots pour lui). Et pourtant, ce qui sera fait demain pour le copain portugais pourra l'être après-demain pour l'ordinateur...

On peut d'emblée faire percevoir les difficultés liées à l'exhaustivité des indications formalistes à fournir en reprenant le problème de l'explication à fournir au « copain portugais » à propos de la conjugaison. Les consignes pourraient être (pour l'indicatif présent des verbes réguliers) de la première conjugaison, du style

- retirer les deux derniers caractères du mot fourni (« *porter* » devient « *port* »)
- écrire « je », puis un espace puis le morceau de mot obtenu en y collant « e » (on obtient « *je porte* »)
- passer à la ligne, écrire « tu », puis un espace puis le morceau de mot obtenu en y collant « es » (on obtient « *tu portes* »)
- etc.

On constate que ces consignes seront parfaites jusqu'à ce qu'on fournisse au copain portugais le verbe « aimer » qui sera évidemment conjugué, sur nos indications, en « *je aime* ». On perçoit aisément la difficulté du « dépliage formaliste » auquel oblige « la programmation ». Nous ne conjugurons pas de manière « formaliste » et il nous est donc difficile de fournir les consignes formelles (complètes et correctes) nécessaires. Nous sommes là au coeur de ce qui fait l'une des difficultés essentielles de l'algorithmique : l'interlocuteur (= l'exécutant) auquel nous nous adressons est « formaliste » et nos indications doivent s'inscrire dans ce cadre. Au delà de la technique, des langages et des environnements de programmation et, quel que soit le « paradigme » de programmation adopté, c'est l'une des difficultés majeures de cette « discipline ».

La dernière phrase de la définition enferme évidemment tout l'univers de l'analyse et de la programmation. Elle explique aussi pourquoi l'ordinateur (réduit à la seule dimension du matériel) n'existe pas : ce que tout utilisateur a toujours

devant les yeux, c'est un tandem «ordinateur» + «logiciel». Les conséquences de ce constat sont nombreuses : ainsi, il est impossible d'apprendre à un utilisateur les manipulations «de base» de l'ordinateur, puisque ce dernier n'existe pas. On peut seulement faire maîtriser les manipulations de base d'un couple particulier «ordinateur» - «logiciel». Les changements d'interlocuteurs sont d'ailleurs, pour les débutants, sources de bien des mécompréhensions et problèmes.

Enfin, cet éclairage particulier de l'ordinateur (qui est plutôt celui que perçoit l'analyste-programmeur) doit être complété par un autre point de vue, celui de l'utilisateur. Mais alors, comme signalé ci-dessus, le concept d'ordinateur devient inadéquat et doit faire place à un ensemble d'*outils*. Et les règles communes à ces instruments divers sont bien difficiles à mettre au jour.

D'AUTRES TRAITS ESSENTIELS DE L'UNIVERS INFORMATIQUE

Le caractère « formel » des traitements d'informations mis en oeuvre sur les ordinateurs constitue donc bien l'un des traits fondamentaux que doit faire percevoir toute initiation à l'informatique. On pourrait en ajouter bien d'autres, de niveaux et de types fort divers, comme le fait de « *penser en arbre* » (présent aussi bien dans la gestion des répertoires d'un disque dur que dans le parcours des menus d'un logiciel), *l'arbitraire de nommer*, (on « peut » appeler « SOUSTRAIRE » une procédure qui n'effectue pas une soustraction ou « RAPPORT » un fichier texte contenant une simple lettre), le glissement d'une civilisation de l'analogique à un *univers du digital* (ou du digitalisé) (dont l'ordinateur est l'un des aspects essentiels), le « *faire faire en différé* » qui est au coeur de l'algorithmique et de l'automatisation des tâches (fut-ce sous la forme des macros-instructions permises par les logiciels en attendant que ceux-ci soient assortis de véritables outils de programmation), etc. Ce sont tous ces aspects (et beaucoup d'autres qu'il nous reste à tirer au clair) qui doivent constituer la trame d'un « cours d'informatique » qui doit continuer à privilégier les aspects formatifs et non la « mise sur le marché » de « cliqueurs » inadaptables et bornés.

En bref, comme l'écrit Jean-Claude Boussard (Boussard 92), « *Enseigner l'informatique, c'est en fait parler - pour la première fois dans un même discours - de systèmes de faits et de valeurs qui soient à la fois formels, finis mais très complexes, automatisés et intimement intégrés à la vie sociale.* »

Charles DUCHÂTEAU

CeFIS, Facultés N-D. de la Paix
rue de Bruxelles, 61

B-5000 NAMUR

Tél +32 81 725060

Fax +32 81 725064

E-mail : cduchateau@cc.fundp.ac.be

BIBLIOGRAPHIE

- ACM Turing Award Lectures ; The First Twenty Years. 1966-1985.* ACM Press, New York, 1987.
- J. ARSAC (1987) *Les machines à penser. Des ordinateurs et des hommes.* Editions du Seuil, Paris, 1987.
- J-C. BOUSSARD (1992) *On n'arrête pas l'obscurantisme.* Document de travail, Université de Nice
- P. BRETON, G. DUFOURD, E. HEILMANN (1990) *L'option informatique au lycée. Secondes, Premières, Terminales.* Hachette, Paris, 1990.
- C. DUCHATEAU (1990) *Images pour programmer.* Be Boeck Wesmael, Bruxelles, 1990
- D. HOFSTADTER, D. DENNETT. (ED). (1987) *Vues de l'esprit. Fantaisies et réflexions sur l'être et l'âme.* InterEditions, Paris, 1987.