

L'ordinateur et ses périphériques

Une course d'obstacles épistémologiques

Emmanuel Saint-James

Sorbonne Université

Par trois fois, l'histoire de l'informatique a été présentée comme une succession de générations : de processeurs, de machines, de langages de programmation. Ces considérations technologiques, non dénuées d'intérêts économiques plus ou moins avoués, sont irrecevables et la première partie de cet article montrera pourquoi. La deuxième partie exposera une vision fondée sur la notion bachelardienne d'obstacle épistémologique, en percevant dix obstacles qui furent franchis en adjoignant à l'*unité arithmétique et logique* de nouveaux *périphériques*, débouchant non seulement sur des alliages, au sens deleuzien, inédits, mais aussi sur des branches théoriques nouvelles de la discipline. Ainsi, par un étonnant paradoxe, ce qui est central dans l'évolution des ordinateurs est ce qui en est à la périphérie. Toutefois, une partie de la discipline n'apparaît pas dans cette perspective, ce qui s'explique par un statut épistémologique en fait différent, cette branche entretenant avec la science informatique un rapport semblable à celui de la médecine avec la biologie¹.

1. Les idées du présent article ont d'abord été exposées à une conférence de 2022 dans le cadre des 75 ans d'informatique en France. Filant la métaphore entre cette course de 900 mois semée de périphériques et le 400 mètres-haies, comportant lui aussi dix obstacles, cette conférence et son support iconographique animé, ironiquement sportif, est disponible à cette adresse : <https://www.lip6.fr/75ans/?guest=Saint-James>.

Je remercie Jean-Luc Mounier pour ses encouragements à élaborer cette conférence, et je remercie Éric Guichard pour son invitation à la réexposer à l'ENS de Lyon.

Les générations perdues

Les générations de langages

En première approche, l'histoire des langages de programmation est essentiellement une suite de *dispenses de codage* après le premier d'entre eux :

1. les langages binaires des processeurs imposent le codage des instructions, des adresses en mémoire, des structures de données et de l'algorithme ;
2. les langages d'assemblage, version alphabétique des précédents, dispensent du codage des instructions ;
3. les langages de programmation dits impératifs, type FORTRAN, dispensent du codage des expressions arithmétiques ;
4. les langages de programmation dits fonctionnels (ou applicatifs selon la terminologie de l'époque), type LISP, dispensent du codage des données ;
5. les langages de programmation dits logiques (ou déclaratifs), type PROLOG, dispensent du codage de l'algorithme.

Cette suite appelle quelques remarques :

- le codage manuel en binaire ne subsiste pratiquement plus que chez les constructeurs de processeurs ;
- l'écriture directe en langage d'assemblage est devenue marginale ;
- chaque génération permet d'exprimer plus brièvement que la précédente une même intention.

Fort de ces évolutions, le cinquième terme de la suite a été vu comme devant fatalement marginaliser voire éliminer ses prédécesseurs. Pourtant, c'est lui qui ne leur a pas survécu. Ce pronostic s'est révélé erroné parce qu'il ne tenait pas compte d'autres phénomènes :

- les influences mutuelles :
 - la *récurtivité* et les *fonctions anonymes*, naguère considérées comme des bizarreries déraisonnables des langages fonctionnels, sont disponibles dans nombre de langages impératifs d'aujourd'hui, adoptant même le *passage à la continuation* pourtant né dans un langage fonctionnel dynamiquement typé,
 - la reprise sur échec, principe central de la programmation logique, est disponible dans beaucoup de langages, via une interface avec PCRE², l'interprète d'expressions rationnelles décrivant des *motifs*,
 - les mécanismes d'*instanciation* et d'*héritage de classe*, ont été adoptés par toutes les familles de langages (montrant au passage qu'il s'agit d'une méthodologie universelle et non d'une nouvelle famille de langages comme cela est souvent présenté) ;
- le besoin de langages à l'expressivité volontairement restreinte :

2. Perl Compatible Regular Expressions.

- renoncement à la complétude algorithmique, notamment dans les langages d'interrogation de bases de données comme SQL,
- rejet d'expressions non typables statiquement, en ADA ou ML et ses successeurs,
- restriction des entrées-sorties dans les langages embarqués comme Javascript,
- la réflexivité, facilitée en Lisp par l'identité entre programmes et données, est découragée, voire interdite.

En définitive, les langages de programmation tendent vers une certaine homogénéité, mais en restreignant l'expressivité. Celle-ci n'est plus la priorité, la sécurité l'a remplacée.

Les générations de machines

Les ordinateurs reposent sur des architectures différentes, pourtant il a été proposé un découpage générationnel trompeur, qui a d'ailleurs fini par tourner court :

1. première génération, utilisation de tubes à vide ;
2. deuxième génération, utilisation de transistors ;
3. troisième génération, utilisation de circuits intégrés (câblage de plusieurs transistors) ;
4. quatrième génération, utilisation de circuits très intégrés (changement qui n'est plus que quantitatif) ;
5. un projet dit de *cinquième génération* lancé en 1982 promettait en dix ans de résoudre par une *machine Prolog* le problème de la lenteur de ce langage ; mais ce projet :
 - n'a abouti qu'à un nouveau compilateur Prolog, aucune machine n'ayant jamais été décrite et encore moins réalisée ;
 - n'avait pas médité l'échec des *machines Lisp*, qui avaient déjà reporté sur le matériel le problème, également présent en Prolog, de la récupération de mémoire, d'où un gain indéniable mais insuffisant pour rendre compétitives ces machines spécialisées³.

Cette cinquième génération annoncée par le Japon a tellement inquiété les pouvoirs publics européens qu'il se précipitèrent pour financer un projet apparemment concurrent. Le fiasco qui en résulta, typique de ce qui se passe

3. LICOT, responsable de ce projet, éditait un journal dont seuls les sommaires ont été mis en ligne ; le numéro 24 de 1989 (<https://www.airc.aist.go.jp/aitec-icot/ICOT/Museum/JOURNAL/icot-jrnl24.html>), possédé par l'auteur, suggère déjà qu'il s'agissait d'un mauvais choix, au regard du *neuro boom* (sic) apparu depuis 1982.

lorsque le pouvoir politique s'immisce dans la dynamique de la recherche scientifique s'appelait *machines à réduction*⁴ :

- l'idée était d'abandonner l'architecture dite «de von Neumann», au profit des modèles mathématiques du calcul antérieurs à l'ordinateur ;
- ces modèles, le *lambda-calcul* et son dérivé la *logique combinatoire*, reposent sur un mécanisme de *réduction*, consistant à remplacer l'application d'une fonction sur un argument par une version spécialisée de cette fonction pour cet argument ;
- cette réécriture induit des copies incessantes à un coût faramineux, raison pour laquelle les pionniers de l'informatique se sont empressés d'inventer :
 - l'*adresse en mémoire*, réalisation concrète de la notion abstraite de variable en mathématique,
 - le *registre d'adresse*, permettant de substituer une variable à sa valeur par une indirection via un registre d'adresse, non plus par réécriture de tout le programme ;

Ce projet de machines à réduction n'aura abouti qu'à une réduction de ses crédits. En l'occurrence, la montagne n'aura même pas accouché d'une souris, puisque, comme on le verra, c'est justement l'invention de la souris d'ordinateur qui sera la vraie révolution de cette époque.

Les générations de processeurs

Connue sous le nom de «loi de Moore», pionnier de la fabrication de micro-processeurs, une autre perspective concerne l'évolution de ceux-ci, faussée cette fois par des problèmes de vocabulaire :

- en science, le mot «loi» s'applique à des phénomènes naturels, l'utiliser pour un objet manufacturé est usurper la confiance mise dans ce mot dans les sciences de la nature (un choix peu surprenant dans une langue désignant l'informatique par «*computer science*» qui institue cette usurpation) ;
- elle a connu plusieurs énoncés⁵ :
 - en 1965, que le nombre de transistors sur un circuit intégré doublait tous les ans,
 - en 1975, que ce ne serait finalement que tous les deux ans,
 - et finalement que ce serait la «performance des ordinateurs» qui doublerait tous les 18 mois, glissements de sens révélateur :

4. Une description peut être trouvée dans Simon Peyton Jones : *GRIP—a high-performance architecture for parallel graph reduction* in Conference on Functional Programming Languages and Computer Architecture LNCS274 1987 Springer, également disponible à l'adresse <https://www.microsoft.com/en-us/research/wp-content/uploads/1987/09/GRIP.pdf>, où on peut constater que la récupération de la mémoire repose sur la solution classique du marquage bloquant.

5. Voir Christophe Lécuyer et Hyungsub Choi *Les secrets de la Silicon Valley ou les entreprises américaines de microélectronique face à l'incertitude technique* Revue d'histoire moderne et contemporaine 2012/3 n° 59-3 Belin.

- il ne s'agit plus du seul processeur mais de l'ordinateur,
- il ne s'agit plus de taux d'intégration mais de « performance », mot suffisamment vague pour permettre de postuler ce qu'on veut.

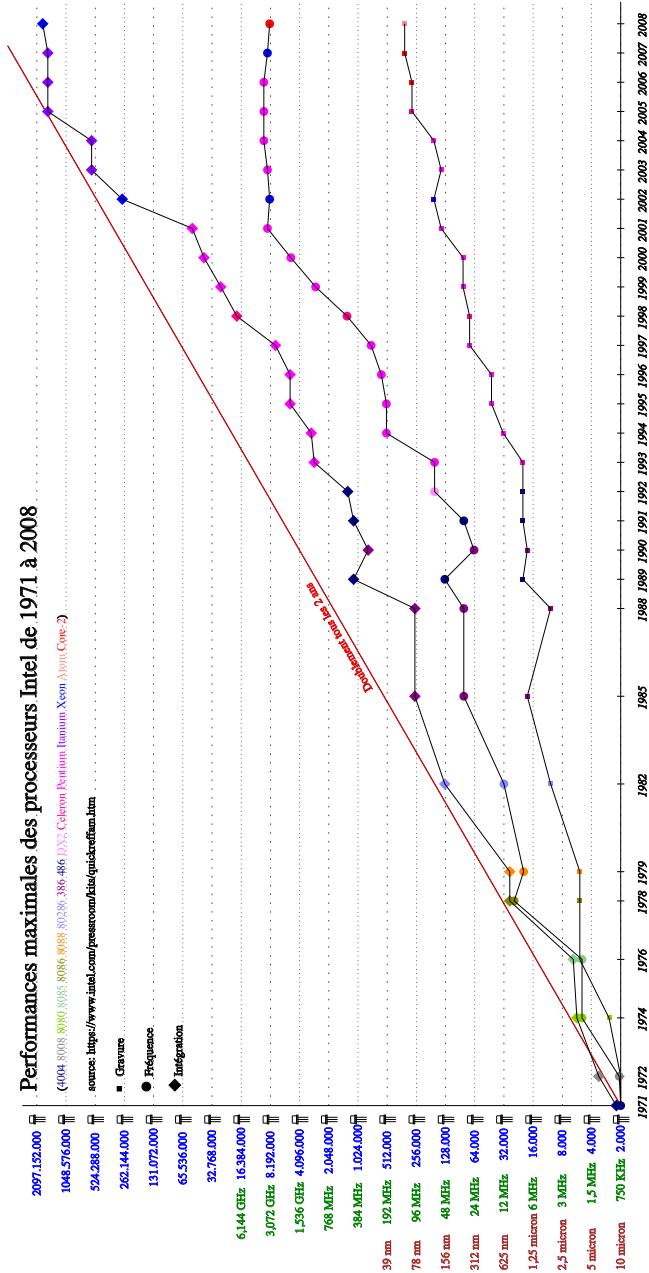
Le site Web officiel de l'entreprise concernée fournit des tableaux de chiffres sur ses processeurs de 1971 à 2008⁶. Le graphique de la page suivante les présente sur une échelle logarithmique afin que la diagonale du repère représente le doublement tous les 2 ans. Il montre que le nombre de transistors intégrés dans un processeur suit, quoique légèrement en dessous, cette diagonale, mais que depuis 2001 l'odyssée de la fréquence et de la gravure stagne.

Cette prétendue loi est donc moins trompeuse que les deux perspectives précédentes, mais elle obscurcit voire ignore les évolutions qui ont crédibilisé ces discours sur le doublement des performances :

- la plus grande finesse de gravure a permis en fait :
 - l'accélération des calculs parce que le chemin parcouru par les électrons est plus court,
 - l'adjonction au processeur de mémoires dites *cache*, en plus des registres,
 - la multiplication des unités arithmétiques et logiques dans le processeur (les *multicœurs*);
- l'adoption d'une longueur identique pour toutes les instructions d'où :
 - le chargement de toute instruction, opérandes compris, en un seul cycle d'horloge,
 - la parallélisation des étapes (de 2 à 31 selon les processeurs) de traitement des instructions par un pipeline ayant autant d'étages que d'étapes, ce qui permet d'approcher l'égalité entre fréquence d'horloge et nombre d'instructions exécutées par seconde (quand un pipeline de n étages est plein, n étapes sont exécutées pour n instructions en 1 cycle d'horloge, soit l'équivalent d'une seule instruction exécutée en 1 cycle);
 - le branchement prédictif, qui augmente les moments où le pipeline est plein.

Les informations sur les années postérieures à 2008 ne sont plus fournies sous forme tabulaire, trop de choses ayant changé depuis. La fréquence d'horloge, en particulier, n'est plus une constante : elle varie automatiquement en fonction de l'activité des cœurs, afin d'éviter des surchauffes ainsi qu'une consommation excessive lors d'un fonctionnement sur batterie. Le nombre de transistors a cessé de croître exponentiellement : seulement une dizaine de fois plus par décennie. La gravure a continué à s'affiner, s'approchant du nanomètre, mais ce progrès est moins dû à l'industrie de l'informatique, à strictement parler, qu'à celle de la photolithographie, notamment l'entreprise hollandaise ASML dont les machines produisent les *wafers* constitutifs de la plupart des processeurs modernes.

6. <https://www.intel.com/pressroom/kits/quickreffam.htm>.



Sans nier l'importance des investissements financiers sur le plan matériel, il faut noter que l'amélioration générale est aussi due à l'investissement intellectuel ayant abouti à des idées nouvelles (pipeline et multicœurs). L'opposition maladroite des dénominations RISC (reduced instruction set computer) et CISC (complex instruction set computer) a masqué que la question centrale est la longueur unique des instructions, car elle permet le pipeline, et non la longueur des jeux d'instructions (ceux des premiers processeurs étaient d'ailleurs encore plus réduits que ceux des victorieux processeurs RISC).

Obstacle épistémologique et alliage innovant

Les trois perspectives citées avaient pour but de justifier des investissements dans une direction bien précise. Ce point de vue inévitablement biaisé n'est pas utilisé en histoire des sciences, qui s'est forgé des concepts plus profonds :

- la notion d'*obstacle épistémologique*, élaborée par Gaston Bachelard⁷, qui empêche d'étudier scientifiquement un phénomène, notamment les préjugés inconscients qui se sont glissés dans la connaissance générale ;
- surmonter ces obstacles permet d'effectuer une *rupture épistémologique* dans la compréhension d'un phénomène ;
- des auteurs plus récents ont parlé de *changement de paradigme*, notion équivalente ; à l'inverse, on peut voir chez Léon Brunschvicg⁸ une description avant la lettre de ruptures épistémologiques dans les élargissements successifs de la notion de nombre en mathématique (entiers, fractions, irrationnels, complexes) ;
- enfin, la notion d'*alliage*, bien connue en métallurgie mais que le philosophe Gilles Deleuze a étendue⁹.

L'histoire de l'informatique se révèle ponctuée d'alliages nouveaux entre le processeur et un *périphérique*. La thèse ici défendue est que l'histoire de l'informatique est essentiellement une suite de franchissements d'obstacles épistémologiques permettant des alliages entre le processeur et un périphérique nouveau ou issu d'un autre univers technologique. Chacun de ces franchissements suscite l'élaboration ou le rajeunissement de méthodologies voire de théories qui finissent par déborder leur inspiration matérielle originelle.

7. Gaston Bachelard. *La formation de l'esprit scientifique*, Paris, Vrin, 1938.

8. Léon Brunschvicg. *Les étapes de la philosophie mathématique*, Paris, Félix Alcan, 1912.

9. Gilles Deleuze et Félix Guattari. *Mille Plateaux*, Paris, Les Éditions de minuit, 1980.

La mémoire

Il semble aujourd'hui évident que la mémoire soit indispensable à un ordinateur. Cependant :

- un des pionniers du calcul automatisé considérait que l'utilisation d'une mémoire conduirait nécessairement à un ralentissement des calculs et a cherché une autre voie même après la diffusion des premiers ordinateurs¹⁰ ;
- ici l'obstacle était de s'aligner sur l'épistémologie de la physique et non des mathématiques :
 - pour la physique, *connaître c'est mesurer* (avec des instruments de mesure),
 - pour les mathématiques, *connaître c'est abstraire* (avec des variables).

L'irruption de la mémoire artificielle a sorti de l'enfance la balbutiante *algorithmique*. Réduire le nombre d'opérations dans une méthode de calcul est une vieille idée, mais pas l'optimisation de la gestion des résultats intermédiaires à garder visibles et rapidement accessibles sur le support d'écriture. L'algorithmique de l'*analyse numérique*, en particulier, s'est complètement renouvelée avec la question des erreurs d'arrondis en base 2, et celle du recours aux représentations en virgule fixe ou en virgule flottante selon les situations.

Le clavier

Avant que l'on songe à emprunter son clavier à la machine écrire (pourtant apparue dès le xviii^e siècle), les périphériques d'entrée étaient :

- soit des potentiomètres, pour les calculateurs analogiques ;
- soit des câbles à enfoncer dans des prises pour relier deux circuits électriques ;
- soit des boutons-poussoirs reliant ou déconnectant deux circuits.

Dans tous les cas, l'idée était que calculer était une opération manuelle de *configuration*. *Programmer* (verbe inventé à cette époque), c'est obtenir un acte à partir d'un texte, ce qui récuse la dichotomie traditionnelle entre la pensée et l'action. Cette irruption du texte en informatique est largement à l'origine de la *mathématisation de la linguistique*, avec les *grammaires génératives* et la renaissance des *théories des types de données*. Les premières relèvent de la syntaxe, mais une syntaxe strictement linéaire interdisant indices, exposants et plus généralement tout effet de mise en pages. Les deuxièmes relèvent non pas, comme l'affirme sans retenue le vocabulaire informatique, de la sémantique, mais de la *signification*, les ordinateurs ne faisant que manipuler des systèmes de signes dont le *sens* leur échappe. Les

10. Girolamo Ramunni. *Louis Couffignal (1902-1966) : un pionnier de l'informatique en France*, Colloque sur l'histoire de l'informatique en France, Grenoble, 1988, AFCET/CNRS.

théories des types garantissent seulement une *cohérence* dans les opérations sur les données, au prix d'une perte d'expressivité comme cela a été signalé.

Le stockage

Initialement, le clavier servait à perforer des cartes, support utilisé notamment pour collecter des données lors des recensements et permettre leur exploitation statistique. Ce précédent longtemps vivace dans l'industrie informatique a freiné le développement des mémoires de stockage électrofiées : le recensement est une opération d'*archivage*, qui est une forme restreinte de *mémorisation* (mot récent, employé d'abord en didactique) en ce qu'elle ne permet pas de modifier ce qui a été mémorisé, usage sans intérêt et même nuisible à l'archivage.

La disponibilité de supports de stockage réinscriptibles et fiables a donné naissance aux versions informatisées des *fichiers*, des *répertoires* et des *bases de données*. Ces dernières ont suscité l'invention de langages spécifiques d'interrogation et d'actualisation, reposant sur des algorithmes sophistiqués.

Le contrôleur d'interruptions

Le contrôleur d'interruptions a permis de remplacer les demandes incessantes et souvent vaines du processeur aux périphériques, par des envois toujours utiles de ceux-ci vers celui-là. Cette nouvelle architecture n'a été possible qu'en renonçant à l'idée qu'une pensée bien construite est nécessairement linéaire. La lecture séquentielle d'un programme utilisant des interruptions n'est alors plus le reflet de la séquence d'instructions que provoquera son exécution : à tout moment une interruption peut lancer l'exécution d'un *preneur d'événement* à la place du processus en cours d'exécution, provisoirement abandonné puis réexécuté ultérieurement à partir de là où il avait été suspendu.

Cette possibilité d'interruption, en particulier par l'horloge pour forcer le processeur à cesser d'exécuter un processus pour se consacrer à un autre lui-même précédemment suspendu, est à l'origine du développement théorique et méthodologique des *systèmes d'exploitation des ordinateurs*, notamment leur *ordonnanceur* permettant le *temps partagé*.

L'écran alphanumérique

Le remplacement des bruyantes machines à écrire par de silencieux écrans était une évidence dont la réalisation a été ralentie par des raisons budgétaires, mais pas seulement. Le gain ergonomique potentiel n'a pas été immédiatement perçu, rédiger un texte étant une activité apparemment pas susceptible des mêmes interrogations sur la fatigue au travail qu'un métier manuel.

C'est à partir de ce moment que s'est développée la branche de l'informatique nommée « interface homme machine », incluant l'ergonomie du logiciel et plus tard le *Web-design*, une branche très liée à l'évolution des périphériques.

Le réseau

L'interconnexion des ordinateurs a été freinée d'abord par la volonté des constructeurs d'ordinateurs mais aussi d'appareils de télécommunications, de garder captives leurs clientèles respectives. Toutefois le plan épistémologique présentait un double obstacle :

- la tendance à assimiler sécurité et inaccessibilité physique, qui a été dépassée :
 - par un renouveau de la *cryptographie*, bénéficiant des avancées de l'algorithmique,
 - par une mathématisation poussée des protocoles de (télé)communication, notamment avec les *codes correcteurs d'erreurs* permettant la fiabilité d'*Internet* ;
- la difficulté à constater qu'un système d'exploitation passe moins de temps à gérer des *fichiers* que des *flux de données* ; cette vision dynamique et non plus statique :
 - a débouché sur l'invention des canaux de communication (le *pipe* et la *socket*) par le système d'exploitation Unix (conçu par une entreprise de téléphonie, non pas d'informatique, ce n'est pas un hasard), permettant la réalisation des algorithmes perpétuels que sont les *serveurs*,
 - a renouvelé la recherche dans les systèmes d'exploitation, les concurrents d'Unix étant obligés de le copier, avec POSIX, ou de disparaître.

L'imprimante

L'imprimante a été un périphérique déterminant dans l'histoire de l'informatique. Avant son apparition, un dirigeant d'un grand constructeur informatique aujourd'hui disparu déclarait : « *je ne vois aucune bonne raison pour un particulier d'avoir un ordinateur chez lui* ». Réflexion surprenante maintenant que chacun a dans sa poche un ordinateur servant aussi de téléphone, mais pas dénuée de fondement à l'époque. En effet, le premier micro-ordinateur, comme nombre d'autres inventions françaises, a été un échec faute de comprendre où était le marché. Celui-ci n'est devenu évident qu'avec son *alliage* avec une imprimante individuelle, qui a permis de capter le marché de la machine à écrire. La confusion ici était entre la *découverte scientifique* ou l'*invention technologique*, qui procèdent d'une nouveauté intrinsèque au domaine concerné, et l'*innovation*, laquelle est définie par les économistes comme *ce qui provoque un déplacement de la demande sur un marché*, ici de la machine à écrire au départ, ensuite de tout le secteur de l'imprimerie avec les imprimantes laser.

De là sont nées les recherches en *typographie numérique*, qui seront reprises par les cartes graphiques ultérieures. Ce nouveau modèle graphique est appelé *vectorel* car il remplaçait la grille de points par des vecteurs, aujourd'hui eux-mêmes remplacés par courbes de Bézier ; mais le terme est resté.

Les périphériques gestuels

Une multitude de nouveaux périphériques apparaît ensuite, dont la caractéristique commune est d'aller par paire ou plus : l'un capte, l'autre restitue, des mouvements d'un objet ou d'un être vivant, et même sa voix et son souffle. Naissent simultanément la souris et l'écran à pixels séparés (dit *écran graphique* par opposition à *écran alphanumérique*), les haut-parleurs intégrés à l'ordinateur couplés à un micro, une caméra ou un instrument de musique MIDI. Les documents gérés par ces périphériques auraient pu être regroupés sous la dénomination de *documents gestuels*, ce qui n'est pas le cas dans l'incons(is)tante, non versionnée et indéboulonnable nomenclature MIME (multipurpose Internet mail extensions) malgré sa définition bien postérieure à eux. Cette incohérence témoigne de l'obstacle épistémologique rencontré :

- penser est plus général que verbaliser ; ce qui se conçoit bien ne s'énonce pas nécessairement par des mots ; les compositeurs, les architectes, les dessinateurs et les sculpteurs le savent bien ;
- les branches de l'informatique induites par ces périphériques sont l'analyse et la synthèse des sons, des images et des mouvements (informatique musicale, reconnaissance des formes, traitement du signal, robotique mais aussi algorithmes de compression) et cette liste s'allonge avec les nouveaux capteurs qui ne cessent d'apparaître, en informatique médicale notamment.

Les coprocesseurs

La présence de plusieurs processeurs dans un ordinateur offre la possibilité de calculs parallèles, terme qui a été employé dans des situations en fait différentes :

- les systèmes d'exploitation à temps partagé sont confrontés à un apparent parallélisme, vécu d'abord comme une gêne, résolue par des mécanismes de verrouillage d'accès, notamment le *sémaphore* pour éviter les *étreintes fatales* ;
- de l'adjonction de coprocesseurs numériques ou graphiques a germé l'idée qu'un processeur pouvait en commander un autre ;
- admettre que des processeurs pouvaient être vus comme périphériques d'un autre aussi puissant (voire moins puissant) était l'obstacle à franchir pour déboucher sur l'*algorithmique répartie*, sans laquelle les *moteurs de recherche du Web* ne pourraient faire face à l'évolution constante des sites Web ;

Cette difficulté à accepter les calculs parallèles pouvait être surmontée en prenant des leçons de musique, les différentes architectures d'ordinateurs n'étant pas sans évoquer certaines formes musicales :

- *Single Instruction Single Data* : monophonie ;
- *Multiple Instruction Single Data* : polyphonie classique ;
- *Single Instruction Multiple Data* : polyphonie romantique ;
- *Multiple Instruction Multiple Data* : polytonalité, polymodalité, atonalité.

Le périphérique distant

La rupture se situe ici dans l'idée qu'un périphérique n'est pas nécessairement relié physiquement et exclusivement à un processeur : les satellites d'un système de positionnement émettent un signal disponible pour qui veut, et doivent être vus comme des périphériques partagés par tous les ordinateurs capables d'interpréter ce signal.

La mobilité, déjà esquissée avec le protocole HTTP qui fonctionne en mode déconnecté contrairement à ses prédécesseurs (FTP, SMTP, IMAP), induit un flot de données perpétuellement changeant et souvent très volumineux qui remet en question l'algorithmique classique consistant à lire préalablement la donnée sur laquelle travailler. Ce qu'on pourrait appeler l'*algorithmique incrémentale* est présentement une branche de la discipline en plein développement.

Une science, l'autre pas

Cette perspective centrée sur les périphériques n'inclut pas la dénommée «intelligence artificielle», ce qui appelle plusieurs remarques :

- ce domaine repose essentiellement sur deux piliers, le modèle de réseau neuronal, emprunté à la biologie, et la notion de base de connaissances, empruntée à une branche de l'informatique évoquée précédemment, les *bases de données* ; ces concepts anciens donnent de meilleurs résultats aujourd'hui grâce à l'accélération des processeurs et à l'agrandissement des mémoires, sans avoir nécessité de franchissement d'un nouvel obstacle épistémologique ;
- comme déjà signalé, la science étudie des phénomènes, non des objets manufacturés ; à ce titre, l'IA n'est pas une science mais plutôt un ensemble de technologies, issues hier de la mécanique, aujourd'hui de l'informatique, demain peut-être de la biologie ;
- l'informatique a d'abord été vue comme une branche des mathématiques ou de la physique, et a fini par s'en dissocier, étant incompatible avec les épistémologies respectives de ces deux disciplines ; l'IA prendra sans doute son indépendance de l'informatique, pour entretenir avec elle des rapports assez semblables à ceux de la médecine avec la biologie.

Bachelard suggérerait que les obstacles épistémologiques reposaient souvent sur un vocabulaire inadapté; sur ce plan, l'IA en fournit un florilège intéressant :

- la locution «intelligence artificielle», traduite mot à mot de l'anglais alors que le mot français «intelligence» se traduit par «*cleverness*» (contre-performance pour une activité incluant la traduction automatique) est doublement trompeur; parler de «savoir numérique» serait plus juste;
- la locution «apprentissage profond», moins employée que son original anglais «*deep learning*» témoigne d'un anthropocentrisme discutable, parler de «mimétisme massif» décrirait mieux la technologie sous-jacente, et il est significatif que ses échecs soient nommés «*over-fitting*», le mot «plagiat», qui devrait s'imposer, étant évidemment moins discret.

Avec de telles approximations, il n'est pas étonnant que les adversaires de la science l'accusent de «ne pas penser». Il faut renoncer à l'inflation du vocabulaire scientifique dans l'espoir d'obtenir plus de crédits, et revenir aux étymologies grecques, latines ou arabes pour forger un vocabulaire précis.

Jean Rostand dans sa passionnante histoire de la biologie, expose que la question centrale de la biologie est «qu'est-ce que la vie?», qui reste toujours sans réponse mais permet de faire avancer la discipline. On pourrait reprendre la formule pour physique & matière, psychologie & individu, et bien d'autres.

Et l'informatique?

- Étudierait-elle l'intelligence? Non, on peut prendre comme contre-exemple une création musicale par ou avec un ordinateur, dont on attend une séduction sonore plus qu'une démonstration d'intelligence. Laissons cette étude à la psychologie génétique, ou peut-être aux mathématiques.
- Au vu de la variété des architectures matérielles et logicielles, c'est semblait-il une question plus générale et plus humaniste : «qu'est-ce que la pensée?»

